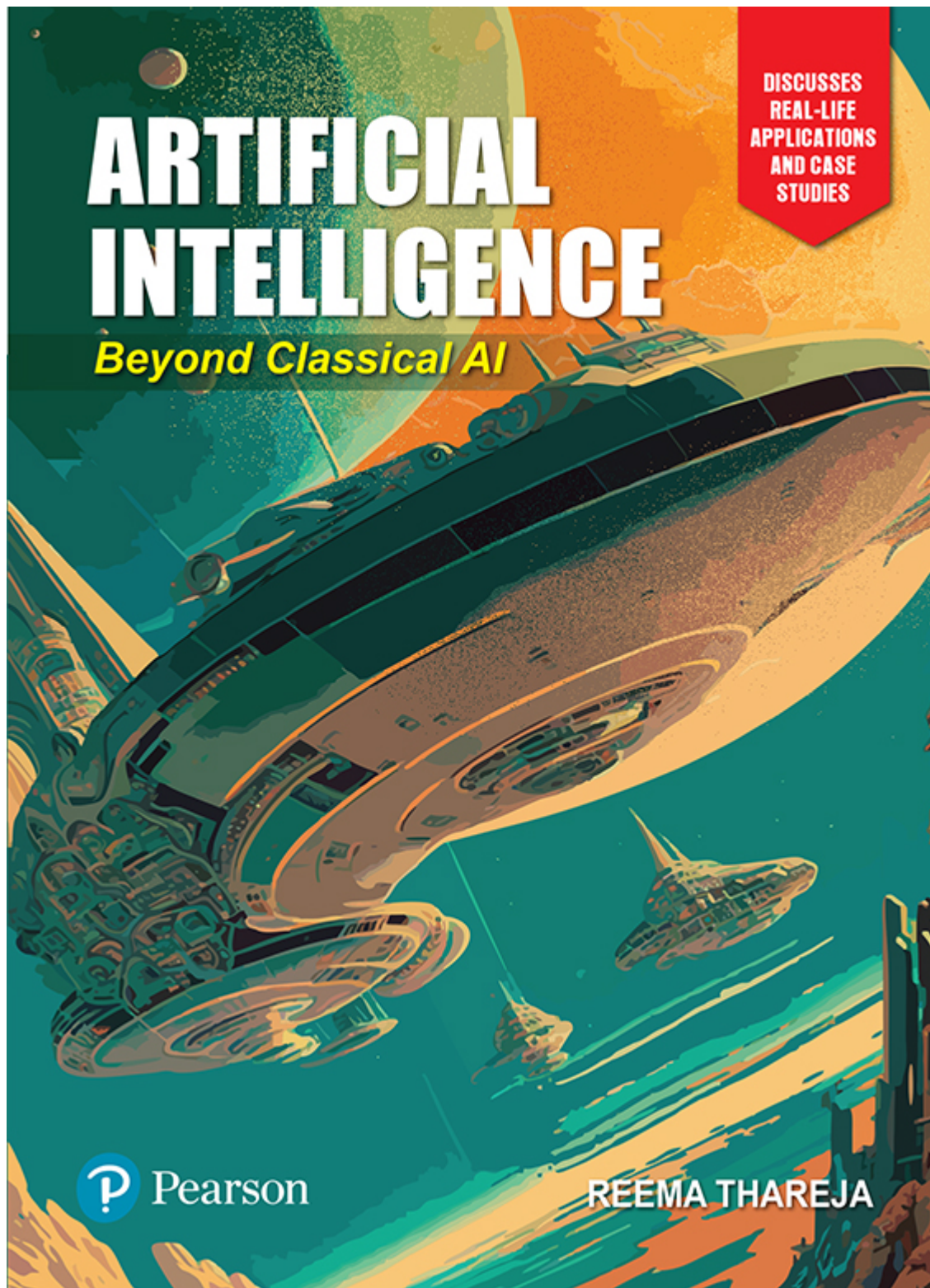




Previous chapter

◀ [Artificial Intelligence: Beyond Classical AI](#)

Next chapter

[About Pearson](#) >

Table of contents expanded

About the Author



Dr Reema Thareja is teaching in the Department of Computer Science, Shyama Prasad Mukherji College, University of Delhi. In her 18 years of teaching experience, she has taught several courses including BA, BSc, MSc, BBA, MBA, BCA, MCA.

She has authored several books in Computer Science and her books have been well accepted across the globe. She has published more than 22 research papers in international journals of repute and filed four patents. She also launched a free mobile app, Jruma, for Apple as well Android devices to promote learning through quizzing and excite users by giving them an opportunity to earn while they learn.

Besides being an author, she is an eminent speaker. Dr Thareja has conducted several Faculty Development Programs, Seminars, Webinars and Student's Workshop in India, UAE and the US. She was also invited as a speaker for the Global Virtual Summit in New York.

She is a member of Computer Society of India, Editorial Board and Elite Speakers group of IMRF, Member of Technical Advisory Board of Refactor Academy, Member of Board of Subject Experts of P. B. Siddhartha College of Arts & Science and Sri Durga Malleswara Siddhartha Mahila Kalasala. For her immense contributions in the field of education, she has received multiple awards and recognitions.

Previous chapter

< [Acknowledgments](#)

Next chapter

[1. Introduction to Artificial Intelligence](#) >

Table of contents expanded

Contents

Preface

About the Book

Acknowledgments

About the Author

Chapter 1 Introduction to Artificial Intelligence

1.1 What is Artificial Intelligence?

1.1.1 How Does AI Work?

1.1.2 Advantages and Disadvantages of Artificial Intelligence

1.2 History of Artificial Intelligence

1.3 Types of Artificial Intelligence

1.3.1 Weak AI

1.3.2 Strong AI

1.3.3 Reactive Machines

1.3.4 Limited Memory

1.3.5 Theory of Mind

1.3.6 Self-Awareness

1.4 Is Artificial Intelligence Same as Augmented Intelligence and Cognitive Computing?

1.5 Machine Learning and Deep Learning

1.6 Applications of AI

1.7 Robotics—an Application of AI

1.7.1 Types of Robots

1.7.2 Uses of Robotics

1.8 Drones Using AI

1.9 The Future of AI

1.10 No Code AI

1.10.1 Why No-Code AI?

1.10.2 Future of No Code AI

1.10.3 Why No-Code AI Must be Used?

1.11 Low Code AI

1.11.1 Low-Code vs No-Code AI Development

1.11.2 Who Uses Low-Code Development?

1.11.3 Low-Code AI Platform for Computer Vision

1.11.4 Components of Low-Code AI Platforms

1.11.5 Disadvantages of Low-Code/No-Code Platforms

1.11.6 Is Low-Code the Future of Software Development?

Programmer's Zone

Key Terms

Recapitulation Points

Answer the Questions

Fill Up

State True or False

Multiple Choice Questions

Find Out Three AI Tools for

Application Based Questions

Links to Explore

Answers

Chapter 2 Artificial Intelligence Technologies

2.1 Techniques in AI

2.2 Machine Learning Model

2.2.1 Types of Machine Learning Algorithms

2.3 Regression Analysis in Machine Learning

2.3.1 How Regression Analysis Works?

2.3.2 Model Evaluation Metrics

2.3.3 Types of Regression

2.4 Classification Techniques

2.4.1 K-Nearest Algorithm

2.4.2 Decision Trees

2.4.3 Random Forests

2.5 Clustering Techniques

2.5.1 Overview of Clustering Techniques

2.5.2 K-Means Algorithm

2.5.3 Applications of Clustering in Real-World Scenarios

2.5.4 Evaluation Metrics for Clustering

2.5.5 How K-Means Algorithm Works?

2.5.6 Pros and Cons of K-Means Algorithm

2.5.7 Density-Based Spatial Clustering of Applications with Noise (Dbscan)

2.6 Naïve Bayes Classification

2.6.1 Understanding Conditional Probability

2.6.2 The Bayes Rule

2.6.3 Types of Events

2.6.4 Naive Bayes Algorithm

2.6.5 Laplace Correction

2.6.6 Pros and Cons of Naive Bayes Algorithm

2.6.7 Applications

2.7 Neural Network

2.7.1 Working of Neural Networks

2.7.2 Pros and Cons

2.7.3 Applications of Neural Networks

2.7.4 How Neural Networks Work?

2.7.5 What is an Activation Function?

2.7.6 Gradient Descent

2.8 Support Vector Machine (SVM)

2.8.1 How Does SVM Work?

2.8.2 Advantages of SVM

2.8.3 Disadvantages of SVM

2.8.4 Applications of SVM in Real World

Key Terms

Recapitulation Points

Answer the Questions

Fill Up

State True or False

Multiple Choice Questions

Case Study-Based Assignment

Enhancing Audio, Visual and Cognitive Skills

Answers

Chapter 3 Artificially Intelligent Machine

3.1 Defining Intelligence

3.2 Components of Intelligence

3.3 Differences Between Human and Machine Intelligence

3.4 Agent and Environment

3.4.1 Key Terminology

3.4.2 Rationality

3.4.3 Structure of Intelligent Agents

3.4.4 Types of Agents

3.4.5 The Nature of Environments

3.4.6 Types of Environments

3.5 Search

3.5.1 Types of Search Algorithms

3.5.2 Properties of Search Algorithms

3.6 Uninformed Search Algorithms

3.6.1 Depth First Search (DFS)

3.6.2 Depth-Limited Search Algorithm (DLS)

3.6.3 Breadth First Search (BFS)

3.6.4 Uniform Cost Search (UCS)

3.6.5 Iterative Deepening Depth-First Search (IDDFS)

3.6.6 Bidirectional Search

3.7 Informed Search Algorithms

3.7.1 Pure Heuristic Search

3.7.2 Best-First Search Algorithm (Greedy Search)

3.7.3 A★ Search Algorithm

3.7.4 AO★ Algorithm

3.8 Hill Climbing Algorithm in Artificial Intelligence

3.8.1 Features of Hill Climbing

3.8.2 Types of Hill Climbing

3.8.3 State Space Diagram for Hill Climbing

3.8.4 Simulated Annealing

3.8.5 Applications

3.8.6 Pros and Cons of Hill Climbing Algorithms

3.9 Adversarial Search

3.9.1 Game Scenarios

3.9.2 Key Terminology

3.9.3 Game Tree

3.9.4 Uses of Adversarial Search

3.9.5 Mini-Max Algorithm

3.10 Alpha-Beta Pruning

3.10.1 Pseudo-Code for Alpha-Beta Pruning

3.10.2 Worst-Case and Best-Case in Alpha-Beta Pruning

3.11 Local Search Algorithms

3.11.1 Hill-Climbing Search

3.11.2 Local Beam Search

3.11.3 Simulated Annealing

3.11.4 Travelling Salesman Problem

3.12 Single Agent Path-Finding Problems

3.12.1 Search Terminology

3.12.2 Brute-Force Search Strategies

Programmer's Zone

Key Terms

Recapitulation Points

Answer the Questions

Fill Up

State True or False

Multiple Choice Questions

Answers

Chapter 4 Knowledge Representation

4.1 Introduction

4.2 Knowledge Representation

4.2.1 What Knowledge Needs to be Represented?

4.2.2 What is Knowledge?

4.2.3 What is Logic?

4.2.4 Cycle of Knowledge Representation in AI

4.2.5 Knowledge Representation Requirements

4.3 Knowledge-Based Agent

4.3.1 The Architecture of Knowledge-Based Agent

4.3.2 Operations Performed By KBA

4.3.3 A Generic Knowledge-Based Agent

4.3.4 Various Levels of Knowledge-Based Agent

4.3.5 Approaches to Designing a Knowledge-Based Agent

4.4 Types of Knowledge

4.5 Techniques of Knowledge Representation in AI

4.5.1 Logical Representation

4.5.2 Semantic Network Representation

4.5.3 Frame Representation

4.5.4 Production Rules

4.5.5 Propositional Logic (PL)

4.6 Syntax of Propositional Logic

4.7 Logical Connectives in Propositional Logic

4.7.1 Logical Equivalence

4.7.2 Truth Table of Propositional Connectives

4.7.3 Precedence of Connectives

4.7.4 Properties of Operators

4.7.5 Limitations of Propositional Logic

4.8 Inference Rules

4.8.1 Types of Inference Rules

4.8.2 First-Order Logic

4.8.3 Generalized Modus Ponens Rule

4.8.4 Unification in FOL

4.8.5 Unification Algorithm

4.8.6 Resolution

4.8.7 The Resolution Inference Rule

4.8.8 The Resolution Process

4.8.9 Explanation of Resolution Graph

4.9 Forward Chaining and Backward Chaining in AI

4.9.1 Horn Clause and Definite Clause

4.9.2 Forward Chaining

4.9.3 Forward Chaining Algorithm

4.9.4 Why Do We Use Forward Chaining?

4.9.5 Backward Chaining

4.9.6 Pros and Cons of Backward Chaining

Programmer's Zone

Key Terms

Recapitulation Points

Answer the Questions

Fill Up

State True or False

Multiple Choice Questions

Answers

Chapter 5 Reasoning and Learning in Artificial Intelligence

5.1 Reasoning

5.1.1 Features of a Reasoning System

5.1.2 Why Focus on Reasoning System?

5.1.3 Methods of Reasoning

5.2 Probabilistic Reasoning in Artificial Intelligence

5.2.1 Uncertainty

5.2.2 Probability

5.3 Bayes' Theorem

5.3.1 Bayesian Belief Network in Artificial Intelligence

5.3.2 Joint Probability Distribution

5.3.3 Understanding the Semantics of Bayesian Network

5.4 Learning

5.4.1 Components of a Learning System

5.4.2 Forms of Learnings

5.5 Clustering

5.5.1 Clusters

5.5.2 Overview of Clustering Techniques

5.6 Explanation-Based Learning (EBL)

5.7 Reinforcement Learning

5.7.1 Components of Reinforcement Learning

5.7.2 Example: Reinforcement Learning in Real World

5.7.3 Characteristics of Reinforcement Learning

5.7.4 Differences Between Reinforcement Learning and Supervised Learning

5.7.5 Types of Reinforcement

5.7.6 Applications of Reinforcement Learning

5.7.7 When to Use Reinforcement Learning in a Large Environment?

5.7.8 Benefits of Reinforcement Learning

5.7.9 Challenges with Reinforcement Learning

5.7.10 Future of Reinforcement Learning

5.7.11 When Not to Use Reinforcement Learning?

Programmer's Zone

Key Terms

Recapitulation Points

Answer Them

Fill Up

State True or False

Multiple Choice Questions

Answers

Chapter 6 Computer Vision (CV)

6.1 Human Vision vs Computer Vision

6.2 How Does Computer Vision Work?

6.3 The Evolution of Computer Vision

6.4 Tasks in Computer Vision

6.5 How Computer Vision Works with Deep Learning?

6.6 Applications of Computer Vision

6.7 Which Vision Succeeded?

6.8 Challenges in Computer Vision

6.9 Understanding Image Pixels

6.9.1 What is Resolution?

6.9.2 DPI and PPI

6.10 Convolutional Neural Networks (CNN)

6.10.1 Understanding the Working of a Simple CNN

6.10.2 THE Pooling Layer

6.10.3 Limitations

6.11 Working with Images Using OpenCV

6.12 Immersive Experience

6.12.1 Elements of Immersion

6.12.2 Types of Immersive Experiences

6.12.3 Applications of Immersive Experiences

6.12.4 Virtual Reality Systems

6.12.5 Types of Augmented Reality

6.12.6 Mixed Reality

6.12.7 Challenges of Immersive Experiences

Programmer's Zone

Key Terms***Recapitulation Points******Review Questions******Fill Up******State True or False******Multiple Choice Questions******Identify the Process******Answers*****Chapter 7 Natural Language Processing****7.1 What is Natural Language Processing?****7.2 Everyday NLP Examples****7.3 Chatbot****7.3.1 Open and Closed Chatbots****7.3.2 Uses of Chatbots****7.3.3 How a Chatbot Works?****7.3.4 Types of Chatbots****7.4 How Does Natural Language Processing Work?****7.5 Components of NLP****7.6 Steps in NLP**

NLP, AI, Machine Learning: What's the Difference?

7.7 Phases of NLP

Data Pre-Processing

Data Processing

7.8 Syntax vs Semantic Analysis

7.9 Applications of Natural Language Processing

7.10 Pros and Cons of Natural Language Processing

Limitations of NLP Systems

Challenges

7.11 Evolution of Natural Language Processing

7.12 Handling Ambiguities

7.12.1 Techniques of Ambiguity Resolution

7.13 The NLP Model of Perception

7.13.1 NLP Communication Model

7.13.2 The Five Perceptual Positions

7.14 Constituency Grammar

7.15 Context-Free Grammar

7.15.1 Derivations

7.15.2 Using Nominals in CFG

7.15.3 Uses of CFG

7.15.4 Types of Derivation

7.15.5 Parse Tree

7.15.6 Ambiguity in CFG

7.15.7 Parser

7.16 Speech Recognition

7.16.1 How Does Speech Recognition Work?

7.16.2 Applications of Speech Recognition

7.16.3 Algorithms for Speech Recognition

7.16.4 Advantages of Speech Recognition

7.16.5 Disadvantages of Speech Recognition

Programmer's Zone

Key Terms

Recapitulation Points

Review Questions

Fill Up

State True or False

Multiple Choice Questions

Answers

Chapter 8 Current Trends in Artificial Intelligence

8.1 AI and Ethical Concerns

8.1.1 Ethical Use of Artificial Intelligence

8.1.2 Is AI Dangerous? Will Robots Take Over the World?

8.1.3 Ethics in AI

8.1.4 AI and Bias

8.1.5 Towards Ethical and Trustworthy AI

8.1.6 Why is Ethical AI Important?

8.1.7 Impact of AI on Jobs

8.2 AI as a Service (AIaaS)

8.2.1 Factors Triggering Growth of AIaaS

8.2.2 The Growth of AIaaS

8.2.3 Challenges of AIaaS

8.2.4 Vendors of AIaaS

8.3 Robotics

8.3.1 Artificially Intelligent Robot

8.3.2 Characteristics of Robots

8.3.3 Types of Robots

8.3.4 Types of Robots Based on Degree of Human Control

8.3.5 Components of a Robot

8.3.6 AI Technology Used in Robotics

8.3.7 Planning and Navigation

8.4 Recent Trends in AI

8.4.1 Collaborative Systems

8.4.2 Machines Assisting Humans

8.4.3 Algorithmic Game Theory and Computational Social Choice

8.4.4 Multi-Agents Reinforcement Learning (MARL)

8.4.5 Neuromorphic Computing

Programmer's Zone

Key Terms

Recapitulation Points

Review Questions

Fill Up

State True or False

Multiple Choice Questions

Answers

Chapter 9 Where AI Is Heading Today?

9.1 Expert System

9.1.1 Popular Examples of the Expert System

9.1.2 Characteristics of an Expert System

9.1.3 Components of an Expert System

9.1.4 Participants in the Development of Expert System

9.1.5 Capabilities of the Expert System

9.1.6 Advantages of Expert Systems

9.1.7 Limitations of Expert Systems

9.1.8 Applications of Expert Systems

9.1.9 Expert System Technology

9.1.10 Development of Expert Systems

9.2 Internet of Things

9.2.1 Examples of Applications of IoT

9.2.2 IoT Products

9.2.3 Challenges

9.2.4 Sensors

9.3 Artificial Intelligence of Things (AIoT)

9.3.1 How Does AIoT Work?

9.3.2 Where Does AI Unlock IoT?

9.3.3 Applications and Examples of AIoT

9.3.4 Benefits and Challenges of AIoT

9.3.5 Future of AIoT

9.4 Edge Computing

9.4.1 Why is Edge Computing Important?

9.4.2 Edge Computing Use Cases and Examples

9.4.3 Benefits of Edge Computing

9.4.4 Challenges of Edge Computing

9.4.5 Edge Computing Implementation

9.4.6 Edge Computing, IoT and 5G Possibilities

9.5 Metaverse

9.5.1 Use Cases of Metaverse

9.5.2 Where is Metaverse Today?

Key Terms

Recapitulation Points

Review Questions

Fill Up

State True or False

Multiple Choice Questions

Answers

Chapter 10 AI Evolving with New Age Techniques

10.1 Fuzzy Logic

10.1.1 Characteristics of Fuzzy Logic

10.1.2 Architecture of a Fuzzy Logic System

10.1.3 Membership Function

10.1.4 Example of a Fuzzy Logic System

10.1.5 Classical and Fuzzy Set Theory

10.1.6 Fuzzy Set

10.1.7 Applications of Fuzzy Logic

10.1.8 Advantages of Fuzzy Logic

10.1.9 Disadvantages of Fuzzy Logic

10.1.10 Trends

10.2 Genetic Algorithms

10.2.1 Advantages of Genetic Algorithms

10.2.2 Limitations of Genetic Algorithms

10.2.3 Basic Terminology

10.2.4 Basic Structure

10.2.5 Genotype Representation

10.2.6 Population

10.2.7 Population Initialization

10.2.8 Population Models

10.2.9 Fitness Function

10.2.10 Parent Selection

10.2.11 Crossover

10.2.12 Mutation

10.2.13 Termination Condition

10.2.14 Application Areas

10.3 Soft Computing

10.3.1 Characteristics of Soft Computing

10.3.2 Need for Soft Computing

10.3.3 Applications of Soft Computing

10.3.4 Elements of Soft Computing

10.4 Transfer Learning

10.4.1 How Transfer Learning Works?

10.4.2 When to Use Transfer Learning

10.4.3 Approaches to Transfer Learning

10.4.4 Traditional Machine Learning vs Transfer Learning

10.4.5 Classical Transfer Learning Strategies

10.4.6 Transfer Learning for Deep Learning

10.4.7 Steps in Transfer Learning

10.4.8 Types of Deep Transfer Learning

10.4.9 Applications of Transfer Learning

10.4.10 When Does Transfer Learning *Not* Work?

Key Terms

Recapitulation Points

Review Questions

Fill Up

State True or False

Multiple Choice Questions

Answers

Programming Language Prolog

Case Study

1 N Queen Problem with Backtracking

2 Warnsdorff's Algorithm For Knight's Tour Problem

3 Rat in a Maze

4 Unique Paths in A Grid

5 The Wumpus World

6 Smart Cities

Index

[Previous chapter](#)

[Next chapter](#)

[< Title Page](#)

[Preface >](#)

Table of contents expanded

1

Introduction to Artificial Intelligence

1.1 What is Artificial Intelligence?

In 2004, John McCarthy defined artificial intelligence (AI) as the science and engineering of making intelligent machines, especially intelligent computer programs. However, much before this definition was coined, the birth of AI was marked by Alan Turing's seminal work, '*Computing Machinery and Intelligence*,' published in 1950. Alan Turing, also known as the 'father of computer science,' raised questions like 'Can machines think?' in this paper. Continuing with his work, he later proposed the 'Turing Test', where a human interrogator would try to distinguish between a computer and human text response.

Few languages that are popularly used to code AI applications are R, Python and Java.

From a layman's view, artificial intelligence (AI), simply means the intelligence demonstrated by machines that help them to mimic the actions of humans. AI simulates natural intelligence in machines that are programmed to learn from experiences, adjust to new inputs and perform human-like tasks.

From a researcher's view, AI is a set of algorithms that generates results without having to be explicitly instructed to do so, thereby making machines capable of thinking and acting rationally and humanely.

Most AI examples, from chess-playing computers to self-driving cars, heavily depend on deep learning and natural language processing techniques.

AI applications perform specialized tasks by processing large amounts of data and recognizing patterns in them. Besides learning from experience, AI applications can recognize objects, understand and respond to language and make decisions to solve real-world problems. AI perform tasks that normally require human intelligence. The main focus of this technology is to build machines and algorithms which are capable of performing computational tasks that would otherwise require human-like brain functions.

While computers execute certain tasks like sorting, computing, memorizing, indexing, finding patterns, etc. far better than humans, they still cannot beat human skills for identifying emotions, recognizing faces, communication and conversation. This is where AI will play a crucial role to enable machines achieve human capabilities.

NITI Aayog: The National Strategy for Artificial Intelligence has defined AI as follows:

AI refers to the ability of machines to perform cognitive tasks like thinking, perceiving, learning, problem solving and decision-making. Initially conceived as a technology that could mimic human intelligence, AI has evolved in ways that far exceed its original conception. With incredible advances made in data collection, processing and computation power, intelligent systems can now be deployed to take over a variety of tasks, enable connectivity and enhance productivity.

1.1.1 How Does AI Work?

AI systems work effectively when fed with a large amount of labelled training data. This data is thoroughly analysed to discover correlations and patterns. These patterns are then used to make predictions about fu-

ture states. For example, a chatbot fed with examples of text chats can learn to converse with humans in real-world applications.

AI programming focuses on three cognitive skills: learning, reasoning and self-correction.

Learning Processes

AI programs focus on acquiring data and creating rules for turning that data into actionable information. Rules, also known as algorithms, provide step-by-step instructions to complete a specific task.

Reasoning Processes

The success of an AI program depends on choosing the right algorithm to reach the desired outcome.

Self-Correction Processes

AI programs are designed to continually enhance their algorithms to provide the most accurate results.

Case Study: Why is artificial intelligence important?

AI allows business organizations to gain insights into their operations that they may not have been previously aware of. There are numerous tasks that AI-enabled machines can perform better than humans. This has acted as the driving force for exploring new business opportunities. It was because of AI that computer software could be used to connect riders to taxis. Uber, one of the largest companies in the world, is utilizing sophisticated machine learning algorithms to predict when people would need a greater number of rides in a particular area so that the company can get drivers on the road before they are needed.

1.1.2 Advantages and Disadvantages of Artificial Intelligence

Artificial intelligence, when used along with machine learning and deep learning techniques, is quickly evolving to process massive amounts of data much faster and makes predictions more accurately than humanly

possible. While huge volumes of data are being generated every day, only AI applications can utilize this data to quickly turn it into actionable information.

Google has become one of the largest players in AI. It provides a range of online services by using machine learning to understand how people use their services and then improving them.

Advantages

1. Performs well on tasks that uses detailed data.
2. Takes less time to perform tasks that needs to process huge volumes of data.
3. Generates consistent and accurate results.
4. Can be used 24 X 7.
5. Optimizes tasks by better utilizing resources.
6. Automates complex processes.
7. Minimizes downtime by predicting maintenance needs.
8. Enables companies to produce new products having better quality and speed.

Disadvantages

1. Involves more cost.
2. Technical expertise required to develop and use AI applications.
3. Lack of trained professionals.
4. Incomplete or inaccurate data may result in disastrous results.
5. Lacks the capability to generalize tasks.

In 2017, Google's CEO, Sundar Pichai, pronounced that Google would operate as an 'AI first' company.

Conclusion: Today, every big enterprise has used AI to improve its operations and gain advantage on their competitors.

1.2 History of Artificial Intelligence

Let us analyse the support for the modern field of AI, right from 1943 to the present day (**Fig. 1.1**).

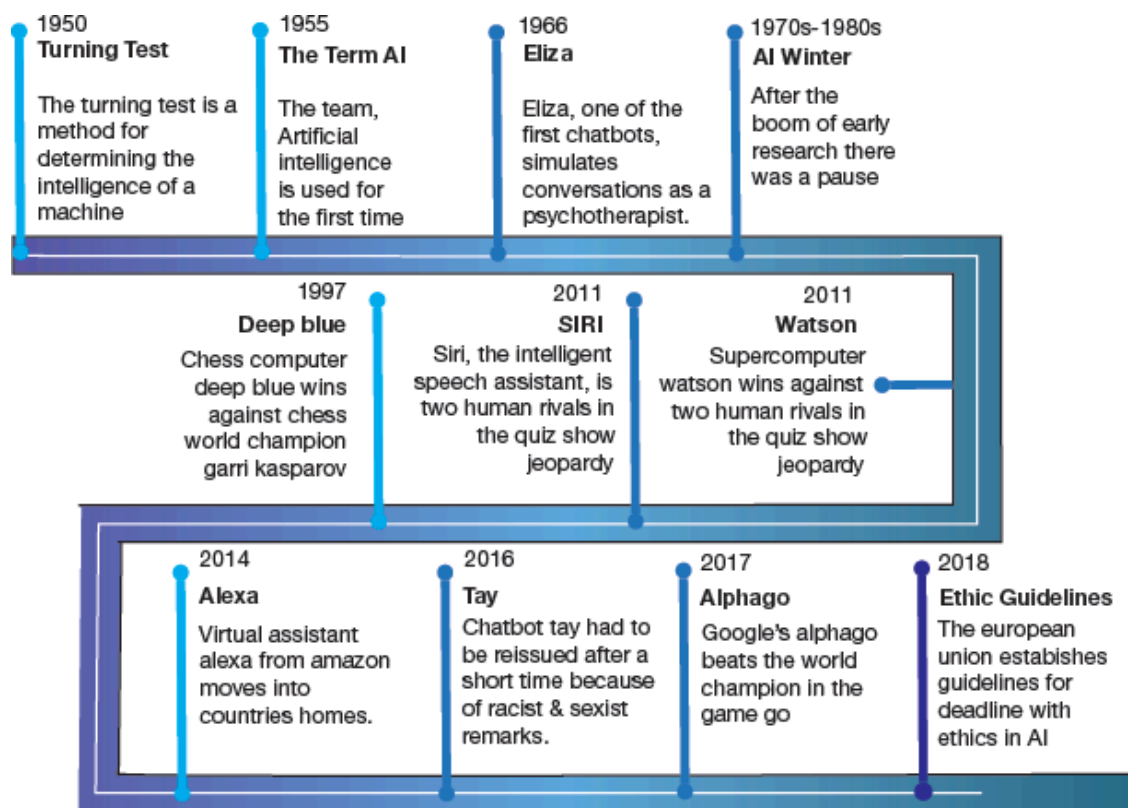


FIGURE 1.1 History of Artificial Intelligence (AI)

In **1943**, Warren McCullough and Walter Pitts wrote a paper that proposed the first mathematical model for building a neural network.

1950 was a significant year in the field of AI. A series of events took place this year including the following:

- Alan Turing demonstrated Turing Test, that could determine whether a machine is intelligent or not.
- Harvard undergraduates Marvin Minsky and Dean Edmonds built the first neural network computer.
- Claude Shannon publishes a paper on programming a computer for playing chess.
- Isaac Asimov published his work on the 'Three Laws of Robotics.'

In **1952**, Arthur Samuel developed a self-learning program to play checkers.

In **1954**, the IBM computer could translate 60 carefully selected Russian sentences into English.

In **1956**, John McCarthy coined the term ‘artificial intelligence’. The scope and goals of AI were discussed in a conference later. It is from this point that we consider the birth of artificial intelligence as we know it today.

In the same year, Allen Newell and Herbert Simon demonstrated the first reasoning program.

In **1958**, John McCarthy developed the AI programming language, Lisp. He also published a paper discussing a complete AI system that could learn from experience as effectively as humans do.

In **1959**, Allen Newell, Herbert Simon and J.C. Shaw developed the General Problem Solver (GPS), a program designed to imitate human problem-solving.

In the same year, Arthur Samuel coined the term ‘machine learning’.

In **1963**, John McCarthy started the AI Lab at Stanford.

MIT Professor Joseph Weizenbaum developed ELIZA, an early natural language processing program that laid the foundation for today’s chatbots.

In **1969**, the first successful expert systems to diagnose blood infections was developed at Stanford.

In **1972**, logic programming language PROLOG was created.

The 1974–1980 period is known as the ‘First AI Winter,’ because it was during this period that DARPA cutbacks in academic grants and funding for AI dried up, stalling further research.

In **1980**, DEC developed the first successful commercial expert system, R1.

In **1982** Japan launched the ambitious Fifth Generation Computer Systems (FGCS) project to develop supercomputers that could support performance and a platform for AI development. Later, in **1983**, even the US government funded research in advanced computing and artificial intelligence.

1987–1993 marked the ‘Second AI Winter.’ In 1992, Japan terminated the FGCS project owing to failure in meeting its ambitious goals. The US also ended the project in 1993 after spending nearly \$1 billion as the results were far from what were expected.

In **1997**, IBM’s Deep Blue machine could beat world chess champion, Gary Kasparov.

In **2005**, STANLEY, a self-driving car, won the DARPA Grand Challenge. Even the US military started investing in autonomous robots like Boston Dynamics’ ‘Big Dog’ and iRobot’s ‘PackBot.’

In **2008**, Google did a remarkable work in speech recognition and introduced the feature in its iPhone app.

In **2011**, Apple released Siri, an AI-powered virtual assistant that could be used through its iOS operating system.

In **2012**, Andrew Ng, founder of the Google Brain Deep Learning project, fed 10 million YouTube videos as a training set to a neural network and used deep learning algorithms to enable the neural network to recognize a cat without being told what a cat is.

In **2014**, Google makes the first self-driving car that could pass a state driving test. In the same year, Amazon released Alexa.

In **2015**, Baidu’s Minwa supercomputer used a convolutional neural network (special deep neural network) to identify and categorize images with a higher rate of accuracy than the average human.

In **2016** Google DeepMind’s AlphaGo defeated world champion Go player Lee Sedol. Go is an ancient Chinese game and winning this game was a

major breakthrough in the field of AI.

In the same year, the first ‘robot citizen’, a humanoid robot, Sophia was created that could perform facial recognition, verbal communication and facial expression.

In **2018**, Google released NLP engine BERT that reduced barriers in translation and understanding by machine learning applications. In the same year, Waymo One service was launched that allowed users to request a pick-up from one of the company’s self-driving vehicles.

In **2020**, Baidu released its LinearFold AI algorithm to develop a vaccine during the early stages of the SARS-CoV-2 pandemic. The algorithm could predict the RNA sequence of the virus in just 27 seconds, 120 times faster than other methods.

1.3 Types of Artificial Intelligence

We can classify an artificial intelligent system into one of the following categories (**Fig. 1.2**).

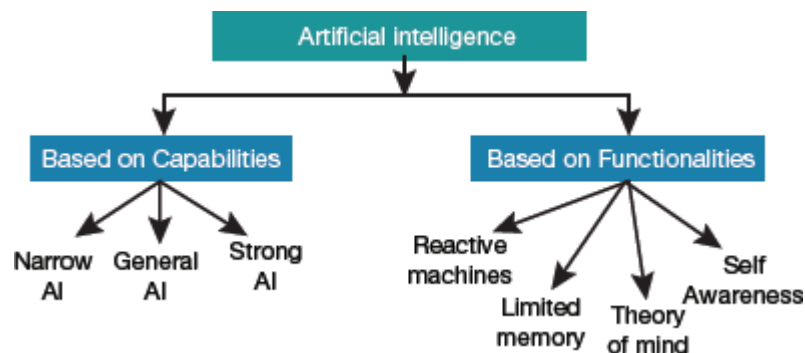


FIGURE 1.2 Types of Artificial Intelligence

1.3.1 Weak AI

Weak AI, also known as narrow AI,

is specifically designed to perform a specific type of task. For example, Siri and Alexa are weak AI systems. These systems are already trained with appropriate responses to classify things accordingly. When you instruct Alexa to play a song, it responds by playing that song.

In fact, majority of AI applications that we use today (predicting weather, stock prices, optimizing business, etc.) come under this category. Weak AI systems operate within a limited context and are the most successful realization of artificial intelligence to date. Application of narrow AI has resulted in significant societal benefits. Google search, Image recognition software, self-driving cars and IBM's Watson are some examples of such systems.

1.3.2 Strong AI

Strong AI, also known as artificial general intelligence or artificial super intelligence (ASI) or superintelligence, makes full attempt to resemble the human brain. It utilizes cognitive skills and fuzzy logic to perform tasks for which it had not been trained earlier. Such a system needs capability for visual perception, speech recognition, decision-making and translations between languages.

As of now, we see general AI only in sci-fi movies, where machines emulate human intelligence and think strategically to perform complex tasks. However, it is believed that soon, ASI will surpass the intelligence and ability of the human brain as demonstrated by HAL, the superhuman, rogue computer assistant in *2001: A Space Odyssey*.

Many people fear that intelligent robots would overrun humanity, but experts says that we need not worry about it as it is not likely to happen in the near future.

Moreover, artificial intelligence systems can also be categorized **into** four groups based on their functionality. This categorization can be given as discussed in **Table 1.1**.

TABLE 1.1 Weak AI vs Strong AI

Weak AI	Strong AI
It supports a narrow range of applications with a limited scope.	It supports a wider range of application with a wide scope.
This application is good at specific tasks.	This application has an incredible human-level intelligence.
It uses supervised and unsupervised learning to process data.	It uses clustering and association techniques to process data.
Example: Siri, Alexa.	Example: Advanced robotics

1.3.3 Reactive Machines

Reactive machines are very basic machines with no memory to store and thus use past experiences to determine future actions. They just perceive the world and react to it. IBM’s Deep Blue, which defeated chess grand-master Kasporov, is such an example. The AI system sees the pieces on a chess board and responds without referring to any of its prior experiences. Since it lacks capabilities to store past experiences, a reactive machine cannot improve with practice.

A strong AI program must pass a Turing Test as well as the Chinese room test.

Reactive machines can perform only a limited number of specialized tasks. Such machines are more trustworthy and reliable as they react the same way to the same stimuli every time.

Deep Blue could only identify the pieces on a chess board and compute its next move based on the rules of chess and the present positions. The AI application could not pursue future potential moves by its opponent or plan its own pieces in a better position. Every move was treated as its own reality, without any connection with any other previous movement.

Google’s AlphaGo is another example of a game-playing reactive machine. The application evaluates future moves using neural network. This gives it an edge over Deep Blue game.

1.3.4 Limited Memory

Limited memory machines retain data but only for a short period of time and can use it only for a limited time. They cannot keep adding data permanently to a library of their experiences. Such systems are usually used in autonomous vehicles since they need not store data (like *recent speed of nearby cars, distance between cars, the speed limit, etc.*) that can help them navigate roads.

AlphaGo also bested world-class competitors of the game, defeating champion Go player Lee Sedol in 2016.

No doubt, limited memory AI is more complex and is used for better possibilities than reactive machines. These machines continuously train a model to analyse and utilize new data. The model also improves using feedback received from humans or the environment and stored as data.

There are three major machine learning models that applies limited memory artificial intelligence:

1. **Reinforcement learning:** It enables continuous learning through repeated trial-and-error to make better predictions.
2. **Long Short-term Memory (LSTM):** Such models use past data to help predict the next item in a sequence. More recent information is given a higher priority for prediction than past data. However, this does not mean that past data is not considered while making predictions.
3. **Evolutionary Generative Adversarial Networks (E-GAN):** The ML model evolves over time to explore new ways of utilizing previous experiences to jump formulate new decisions. The model is continuously striving to discover a better path. It makes use of simulations and statistics, to predict outcomes throughout its evolutionary mutation cycle.

1.3.5 Theory of Mind

The theory of mind focuses on imitating the human brain. For this, it forms representations about the world, starting from thoughts, emotions

and memories. However, such systems are only theoretical as of now but they may become reality very soon.

These machines must understand thoughts and emotions that affect the behaviour of one's self. Theory of Mind machines make decisions keeping in view the feelings experienced through self-reflection and determination.

1.3.6 Self-Awareness

Once Theory of Mind is established, then the next step would be incorporating self-awareness in AI systems. Self-awareness in machines enable them to possess human-level consciousness, understand the current state—its own existence in the world and use that information to deduce others feelings. Self-AI systems will be able to understand what others may need. They could interpret user's feelings by learning not only what they communicate to them but also how they communicate it. Knowledge imbibed by researchers and its learning of the conscious context will help them to respond to an event. However, such systems do not yet exist but may become available in the near future.

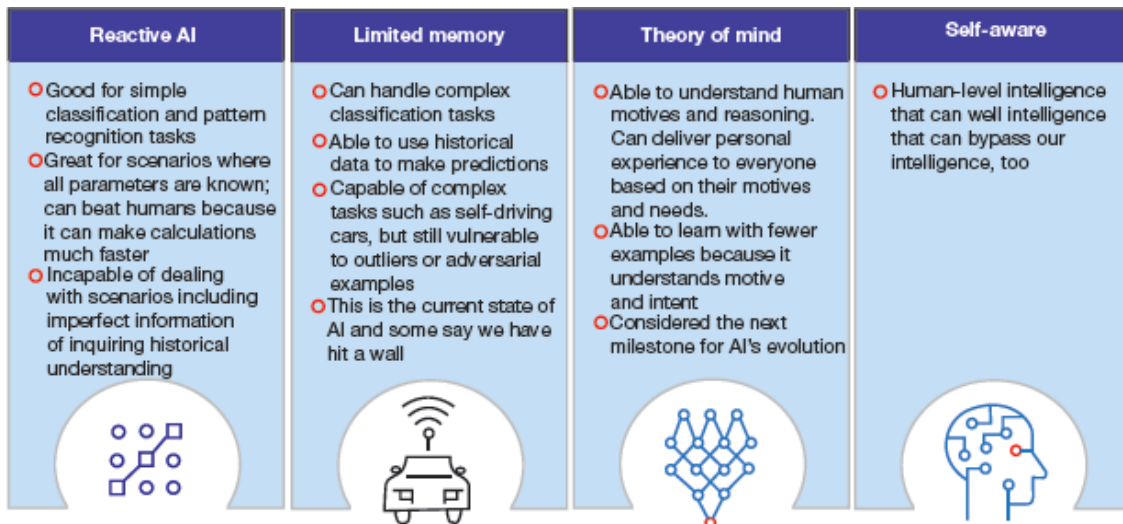


FIGURE 1.3 Types of AI Systems Based on Functionalities

1.4 Is Artificial Intelligence Same as Augmented Intelligence and Cognitive Computing?

Some experts think that *artificial intelligence* is the same as augmented intelligence. But this is not true. Let us understand the difference between

the two terms.

Augmented intelligence is weak AI that simply improves products and services. For example, automatically highlighting vital information in business is a part of implementing augmented intelligence.

Artificial intelligence or true AI, strong AI or artificial general intelligence, is the future AI that would far surpasses the human brain's ability to understand and implement complex tasks. As of now, though lot of work is being done in this domain, AI remains within the realm of science fiction. Technologies like quantum computing can play an important role in making AGI a reality.

The term AI when used in reference to machines, makes machines obtain human intelligence by simulating how humans sense, learn, process and react to information in the environment.

Cognitive computing is a term that is especially coined for products and services that mimic that augments human thought processes.

MACHINE LEARNING

Let us try to understand the term 'machine learning' with the help of some real-world scenarios.

Scenario 1: Imagine that you are given the task of finding the missing number in the series-10,20,30,40, ?

Yes, you are correct. That's 50. But how did you get the answer?

You quickly did some computations in your mind and observed the pattern in the data values. The whole idea of machine learning is to imbibe the same thinking process in machines.

Scenario 2: Before going for playing a cricket match, you always use your past experiences to understand how a batsman/bowler plays. For example, if you know a bowler gives a straight ball after every 2 left balls and 1 right ball, then you can better use this information to prepare yourself for

the next ball. In machine learning, we make machines also to learn from past experience/data.

Scenario 3: When you buy a new mobile phone, you can yourself compute the price of that phone vis-à-vis the configuration and brand. For example, you know a phone with 128 GB RAM, stereo speakers, iOS 16, hex-core would cost around 80K. So, given the details, we can predict the price. Similarly, a machine learning program is developed to predict values based on certain specifications.

Now let us go deeper in this terminology. Machine Learning is a branch of computer science that analyses data and identifies patterns to teach a machine to deduce results and make decisions without any human intervention.

ML algorithms learn from experiences rather than instructions.

They automatically learn and improve by learning from their output. For this, humans do not have to write instructions for them to produce the desired output. They learn by analysing data sets and comparing the final output. In case of any error, they repeat the learning process until the accuracy of the outputs improve.

This automation not only saves human time and effort but also makes better decisions.

Thus, machine learning is an application of artificial intelligence (refer **Fig. 1.4**) that enables machines to automatically learn and improve from experiences without being explicitly programmed.

It enables computers (machines) to make data-driven decisions rather than being explicitly programmed for carrying out a certain task. These algorithms are designed to learn and improve over time when are exposed to new data. Thus, the process of teaching the machines is a structural process that can be broken down into three steps (**Fig. 1.5**).

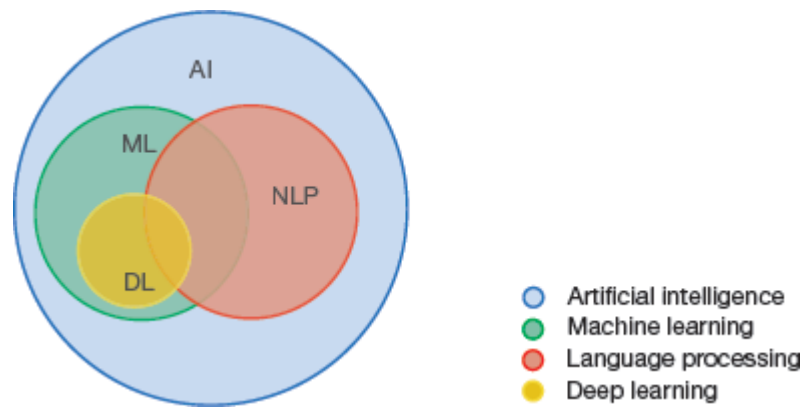


FIGURE 1.4 Relationship between Artificial Intelligence, Machine Learning, Deep Learning and Natural Language Processing

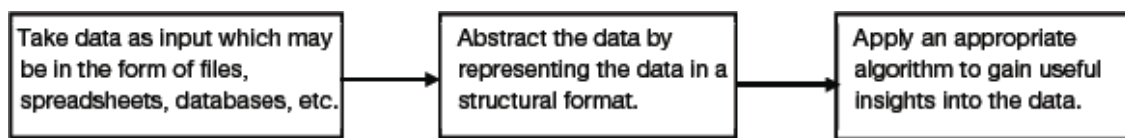


FIGURE 1.5 Making the machines learn

HOW IS AI RELATED TO MACHINE LEARNING?

Machine learning and artificial intelligence are two different techniques where machine learning is a tool for achieving artificial intelligence. The aim of AI is to create intelligent machines that can recognize human speech, have vision to see, assimilate knowledge, strategize and solve problems as humans do.

Machine learning provides machines the ability to learn, forecast and progress on their own without specifically being programmed. ML system improves by learning from more data and experiences.

Imagine you have been given the task of making a robot see, talk, walk, sense and learn. Which technique will you use—ML or AI? Of course, AI. This is because to achieve this task, multiple technologies come into picture (like NLP, computer vision, voice recognition, reinforcement learning, etc). Machine learning is just one of them that will be used only for learning purpose. Therefore, AI is the superset and ML is the subset of AI. That is, ML is only a way to achieve AI (**Fig. 1.6**).

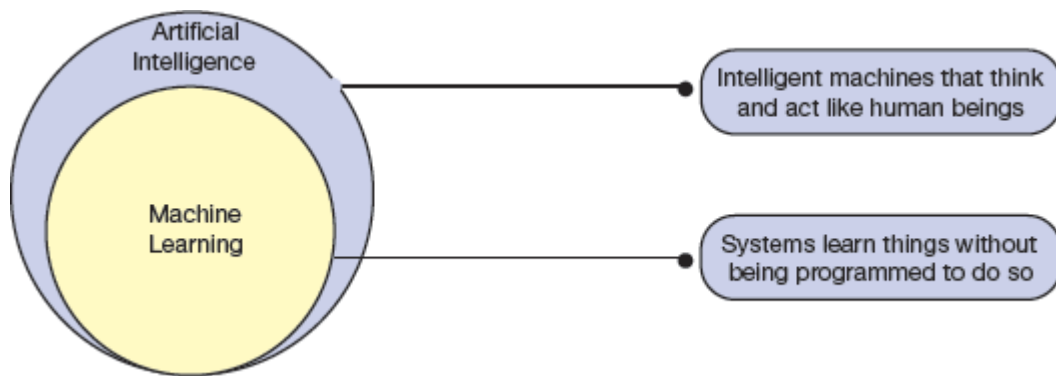


FIGURE 1.6 Difference between AI and ML

Traditional Programming vs Machine Learning

Traditional programming involves manually creating a program that accepts data and returns an output. It rests on a manual process as one has to manually formulate or code rules.



Traditional programming uses conventional procedural language like assembly language or a high-level language such as C, C++, Java, JavaScript, Python, etc. These languages code the rules and write lines of code manually. The program accepts input data, and based on the programming logic, generates the desired output.

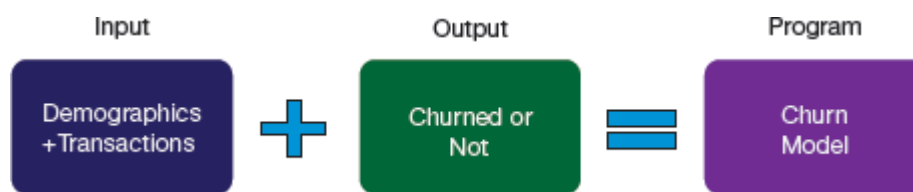
The conventional programming approach is algorithm dependent, and for a program, multiple algorithms can be used. The programmer can analyse the performance of all the algorithms to choose the best that meets the current requirements.

Recently, an advanced type of programming has revolutionized business by adding power to the applications working on intelligence and embedded analytics. **Machine learning** programming, also known as augmented analytics, accepts both data as well as output. It then generates the corresponding program for giving the specified output on that input. This gives insights that can be used to predict future outcomes.

The machine learning algorithm automatically formulates the rules from the data. Unlike traditional programming, machine learning is an automated process that adds the capability of embedded analytics in areas including natural language interfaces, automatic outlier detection, recommendations, etc. All of these features give better insights and reduce decision bias.



For example, if we give customer demographics and transactions as input and use historical customer churn rates as output, then the algorithm will formulate a program (also known as **predictive model**) that can predict if a customer will churn or not.



Therefore, the steps to create a model to predict outcomes can be given as follows:

Step 1: Identify the business question you would like to ask.

Step 2: Identify the historical input.

Step 3: Identify the historically observed output (when the condition is true and when it is false).

Machine learning language goes beyond traditional algorithmic solutions and trains the machine to solve different complex tasks by itself.

Unlike conventional programming, machine learning uses a pre-written algorithm that learns how to solve the problem itself. It is a more sophisticated way of solving a problem.

For Programmers: A data engineer cannot work as a substitute to a conventional programmer and vice versa.

Every data engineer must know at least one coding language. However, a conventional programmer may not learn the machine learning algorithms.

The capabilities of both traditional programming and machine learning are altogether different and cannot replace each other (refer **Table 1.2**). However, machine learning supplements conventional programming. For example, while machine learning can build predictive algorithms, traditional programming, on the other hand, can be used to design user interface or perform data visualization.

Case Study: It is not easy to filter Amna's photos from a set of 100 images using traditional programming. Even comparing every pixel value may not give accurate results. But the same task can be easily done with machine learning. We just need to input different photos of Amna, the machine learning model will learn how she looks like.

TABLE 1.2

Traditional Software	Machine Learning
The primary objective is to meet functional and non-functional requirements.	The primary goal is to optimize the performance of the model as every 0.1 % improvement result in significant business value creation.
The quality of the software primary depends on the quality of the code.	The quality of the model depends upon various parameters related to the input data and hyperparameters tuning.
Traditional software is created using traditional programming languages like C, C++, Java.	Machine learning models can be created using different algorithms. Each of these algorithms may perform differently.
As compared to machine learning, few parameters are required.	To get an accurate prediction, a large number of parameters may be required.
Change in data does not change/impact the business functionality. Only new business rules may have to be accommodated.	Machine learning models may have to be retrained with every new data to ensure same or better performance.
Example: Look at the steps given below to convert Celsius scale to Fahrenheit scale Step -1: Take input (Celsius) Step-2: Apply the conversion formula: $Fahrenheit = Celsius * 1.8 + 32$ Step -3: Print the Output (Fahrenheit) Here, the programmers tell the computer what to do on the input data (multiply Celsius with 1.8 and add 32) to get the result.	Example: Look at the steps given below. Step 1: Feed values in Celsius (i.e., -40, -10, 0, 8, 15, 22, 38) Step -2: Feed corresponding Fahrenheit values (-40, 14, 32, 46, 59, 72, 100) Step -3: Pass these 2 sets of values to Machine Learning (ML) model Step- 4: The ML model computes (convert) the logic to predict any other Fahrenheit value from the given Celsius value. The ML model automatically generates the relationship between Celsius and Fahrenheit to do the conversion.

1.5 Machine Learning and Deep Learning

Majority of AI systems are supported by breakthroughs in machine learning and deep learning techniques. These techniques together go hand-in-hand to deliver an efficient system that sometimes it is just difficult to understand the difference between artificial intelligence, machine learning and deep learning. However, venture capitalist Frank Chen highlighted this difference by stating ‘Artificial intelligence is a set of algorithms and intelligence to try to mimic human intelligence. Machine learning is one of them, and deep learning is one of those machine learning techniques.’

In simple words, machine learning techniques are used to feed data in to the computer so that statistical techniques can be applied to it to help machines ‘learn’ how to solve tasks or which it had not been specifically pro-

grammed. These computers free programmers from writing several lines of code. Machine learning techniques can be further classified into supervised learning (using labelled data sets) or unsupervised learning (using unlabelled data sets).

Deep learning is an advanced machine learning technique that passes data inputs through a biologically-inspired multiple layers of neural network. Such **a neural network architecture consists of a number of hidden layers through which the data is processed**. Data processed in this architecture allows machines to go ‘deep’ in its learning, making connections and weighting input for the best results.

How does AI work?

AI algorithms solve specific tasks in the following manner:

1. Studying very large amounts of data.
2. Making generalizations or statistical estimations model (Refer **Fig. 1.7**).

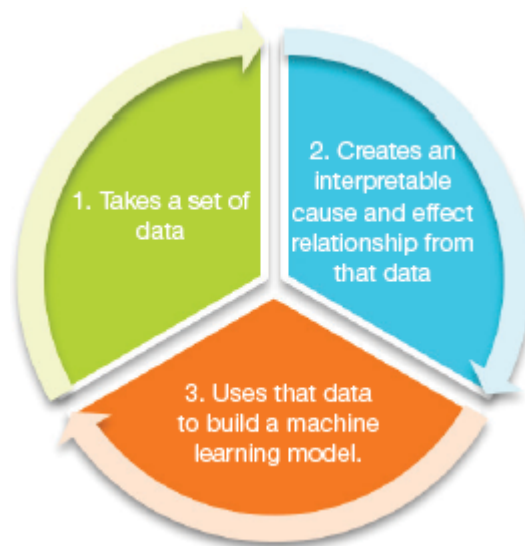


FIGURE 1.7 Building a ML model

Features of AI

1. Artificial intelligence is **autonomous** and can make independent decisions without any human intervention. These systems work silently in the background without the user’s knowledge. The best part of an AI

system is that they don't have to be programmed manually to respond to every situation. Rather, these systems themselves learn through input data and past experiences.

Baidu, a search engine that works in the Chinese language, has a voice cloning tool, which can clone human voice with just 3-4 seconds of audio, as opposed to the previously taken 30 minutes

2. **Predict and adapt: AI systems can** understand data patterns to make decisions and predictions.
3. An AI system **continuously learns** from patterns in data.
4. An AI system is said to be **reactive as it** perceives a problem and acts on perception.
5. **Data driven:** In the last several years, the cost of technology has gone down considerably. With the availability of cheap data storage (hard disk, etc.), fast processors (CPU, GPU or TPU) and sophisticated deep learning algorithms, huge volumes of data can now be easily extracted. This has led to the rise of data centric AI systems.
6. Data driven AI systems can make accurate predictions based on past experiences have outperformed humans. However, success of these systems depends largely on the availability of correctly labelled large datasets.
7. Finally, an AI system is **futuristic** since the scope of it being applied in different areas is continuously expanding.

1.6 Applications of AI

Artificial intelligence has become the cornerstone of innovation. When AI is used along with machine learning and deep learning techniques, it can easily analyse and interpret abundance of data available to make useful predictions and/or automate excessively complex or mundane tasks.

AI technology is helping organizations perceive the world as humans do and then react to real-time events. Most of the companies across different areas are making the best use of AI while avoiding its implications to create an AI ecosystem in which the following are observed:

Business analysts and data scientists work together to define the problems and its objectives.

Data engineers manage the data and the underlying technology to make it operational for analysis.

Data scientists plan, develop, analyse, visualize and model data using a machine learning model.

IT architects take care of the underlying infrastructure required for supporting data science projects in the cloud or on the organization's servers.

Application developers deploy the models into applications to build data-driven products.

In big and small organizations, AI is being used in the following areas.

1. **Automation:** When used in automation tools, AI technologies can enhance the number, efficiency and types of tasks performed. For example, Robotic Process Automation (RPA), is a software that automates repetitive, rules-based data processing tasks traditionally done by humans.
2. **Machine vision:** As the name suggests, this technology allows machines to see. For this, the machine captures visual information using a camera and then analyses using analog-to-digital conversion and digital signal processing techniques. Unlike human eyesight, machine vision is not bounded by biology. For example, it can be programmed to see through walls. Nowadays, machine vision is used in various applications ranging from signature identification to medical image analysis.

Machine vision or computer vision facilitates computers to derive meaningful information from digital images, videos and other visual inputs. Information gained through these inputs is then used to take action. When used with convolutional neural networks, computer vision is used in applications involving photo tagging in social media, radiology imaging in healthcare and self-driving cars in the automotive industry.

3. **Recommendation systems:** A recommendation system recommends or suggests products, services, information to users after analysing data based on a number of factors including history, behaviour, preference and interests of the user. Such systems are an example of data-

driven AI as they need a lot of data for training. For example, you have seen that when you watch a video on YouTube, it suggests many videos that match the videos that you generally search for, prefer to watch or have been watching in the past. In this case, the data about your preferences and history of your activities and behaviour are analysed. Similarly, online shopping websites (like Ajio, Flipkart, Amazon, etc.) recommends you products that you normally prefer or buy (refer **Fig. 1.8**).

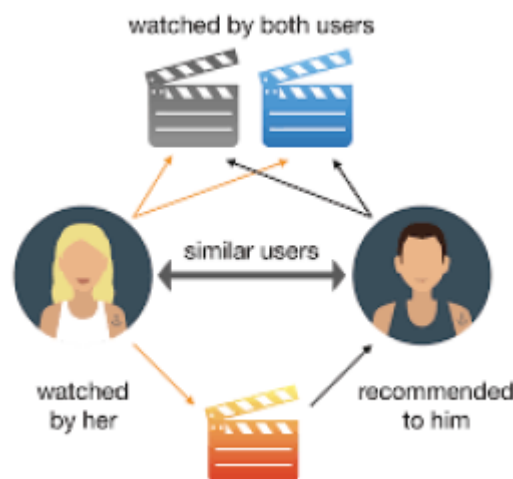


FIGURE 1.8 Recommendation System

4. **Natural language processing (NLP):** Enabling machines to understand and respond to commands given in human language (in digital assistants) has become possible because of NLP—the capability in machine to process natural language (refer **Fig. 1.9**). Another popular example of NLP is spam detection, which looks at the subject line and text of an email and decides if it is a junk mail or not. Practically, NLP

techniques come under machine learning.

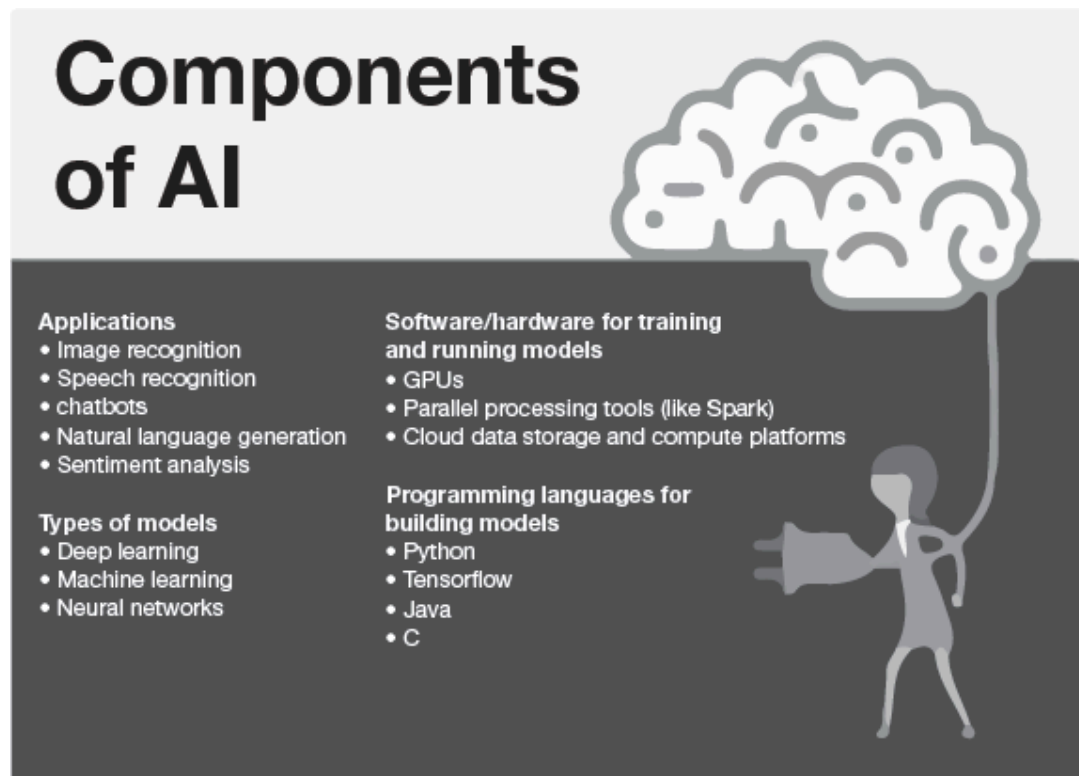


FIGURE 1.9 Components of AI

All smartphones today use voice-to-text feature with which users can start speaking and the phone converts the audio into text. You must have observed that you can ask Google to search a particular keyword.

- Robotics:** Robotics is the science of designing and manufacturing robots. Robots are often used to perform tasks that are either risky for humans or require great precision. Industrial robots now function as cobots performing smaller, multiple tasks while collaborating with humans in warehouses, factory floors and other workspaces.

For example, robots are used in assembly lines for car production, by space scientists to move large objects in space or to click images of areas deep under the sea. These days, extensive research is being done to make robots that can interact in social settings.

- Self-driving cars:** Autonomous vehicles or driverless cars use a combination of techniques including computer vision, image recognition and deep learning to drive a vehicle while staying in a given lane and avoiding unexpected obstructions, such as pedestrians.

The AI system in these vehicles collects data from the vehicle's radar, cameras, GPS and cloud services to generate control signals required to operate the vehicle. From Google's self-driving car Waymo LLC to

Tesla's 'autopilot' feature, AI has revolutionized the automotive industry.

How self-driving cars work? Autonomous vehicles have LIDARS (light detection and ranging) and remote sensors installed that gather information from the vehicle's surroundings. LIDAR uses light from a radar to see objects in front of and around the vehicle. It interprets this information to quickly determine the presence of objects, distances between vehicle and the object. On-board computers combine this information with the data from the sensors to compute the likelihood of the vehicle hitting something. In case the vehicle experiences dangerous conditions, it would either shift lanes, slow down or stop completely.

7. **Personalized online shopping:** AI is extensively being used these days to provide users a personalized experience by showing them products based on multiple filters and using recommender system (that displays them products similar to what they had been searching for or they had purchased in the past).

You must have observed that when you buy a product, say smart-phone from an online shopping website, then AI algorithms on that site will automatically suggest various add-on products for the same model such as screen guards, back covers, earphones, compatible pairable devices, extended warranty plans, damage protection plans and much more.

8. **Enhanced images:** These days, cameras and apps use AI to apply a range of effects on images to enhance their quality, add lighting, blur out the background, apply filters, identify scope of the picture and capture details to let users tag their friends and family.
9. **Social media:** These days, social media is generating an immeasurable amount of data through chats, tweets, posts, etc. In this context, AI is used for applications involving face verification and detecting facial features. To do so, every small detail from an image is extracted and analysed. Moreover, every time you access Facebook, you must have observed that news feed pushes articles based on your interests.
10. **Surveillance:** Traditionally, images from security cameras are monitored manually. Even a small error in this task may be detrimental. In such a scenario, an application based on AI can be developed that takes input from security cameras and trains the system using super-

vised learning algorithms. Such a system can identify potential threats (like intrusion, invalid access, unidentified individuals on defined premises, etc.) and generate an alarm for human security officers to investigate further. This maturing technology is playing a big role in helping organizations fight off cyber-attacks.

The DRDO has developed a robot named ***Daksh*** that safely destroys life-threatening objects. This is a big step to safeguard our soldiers in combat scenarios. Robots are also deployed in anti-submarine operations, fire support, battle damage management, strike missions and laying machines.

11. **Agriculture:** With the increased demand for food owing to population growth, climate change and concerns for food security concerns, organizations are shifting towards automation and use of robotics technology to help farmers explore better ways to protect their crops against bad weather, weeds, market consumption rates, etc. With image recognition capability, AI systems can be used to capture images of crops through farmer's camera and then identify possible defects in them. Early detection of potential threats and then providing soil restoration techniques, tips, and other possible solutions to deal with the identified defects have proved to be very beneficial to farmers.

12. **Prevent natural disasters:** AI systems can be fed with data about thousands of previous disasters. This can help it to accurately predict the occurrence of that disaster in future. For example, some AI systems are being specifically designed to study data about previously occurred earthquakes and similar disasters like tremors and volcanic eruptions, to create a ***neural network***.

Similarly, AI systems are also fed with **seismic information** (information regarding movements in the tectonic plates that reside beneath the Earth's surface) and data about ash particles that arise from lava after a volcanic eruption, and other geological information to predict sudden eruptions.

Other AI systems are designed to accurately predict other natural disasters like cyclones, floods, etc. Did you know that cyclone Fani hit Bangladesh and the south eastern state of India, Odisha in May 2019? The Indian Meteorological Department predicted and kept a stringent vigilance on the situation and evacuated about 1.2 million residents of Odisha on time, and moved them to cyclone shelters built on higher al-

titudes. This saved countless lives. Surprisingly, the death toll due to the cyclone was only 72 people.

13. **Customer service:** These days, businesses are cautious to retain existing customers and also acquire new customer base. You must have observed that almost every website/application has a provision for customers to chat with its representatives. Chat support bots or chatbots are AI systems that use NLP to understand and then extract information from the site and present it to the customer as and when required. These chatbots answer frequently asked questions (FAQs) around topics, like shipping, or provide personalized advice, cross-selling products, etc. Messaging bots on e-commerce sites with virtual agents, messaging apps, such as Slack and Facebook Messenger are all examples of chatbots.

Virtual assistants like Siri, Cortana, Alexa, Amazon's Echo translate human language into desirable actions using NLP. NLP allows them to user's commands (to control other devices, book cabs, make phone calls, order food, etc.) given in user's language.

14. **Video games:** With AI, the gaming industry has reached new heights. The AI system can play with human counterparts and defeat them easily. They are trained to take optimized decisions at every step and also learn from their mistakes to avoid repeating the same mistake again in future. Besides learning, AI is also used for designing the game, developing the characters and for framing the story to a certain extent (refer **Fig. 1.10**).



Credit: franz12 / Shutterstock

FIGURE 1.10 Playing games in a virtual world

- 15. Healthcare:** The capability of AI to crunch numbers fast and learn from historical data, makes this technology very useful in the health-care industry. Detailed analysis has fostered research and quick decision-making. Pathologists are using AI to analyse samples of tissue to diagnose the problem more accurately.
- Moreover, chatbots that can be used as digital health assistants ensure proper medication and treatment of patients. Such systems can help patients find medical information, schedule appointments, understand the billing process and complete other administrative processes. Thus, AI systems have been doing well to improve patient outcomes and reduce costs. For example, IBM Watson is one of the best-known healthcare technologies that understands natural language to respond to questions asked of it. The system also mines patient data and other available data sources to form a hypothesis, and then evaluate it with a confidence scoring schema.
- An array of AI technologies including drones are being used to fight against pandemics such as COVID-19. Apart from providing personalized medicine and X-ray readings, personal health care assistants are used to remind patients to take their medicines, exercise or eat healthier.
- Robots are also used in laparoscopy, neurosurgery, orthopedic surgery, disinfecting rooms, giving medicines to patients and for lot more other tasks.
- 16. Banks:** Banks and other financial institutions are using AI-based systems to provide customer support, detect anomalies and credit card fraud. AI systems can also analyse data to provide financial advice and suggest stocks to buy. Some AI systems are specifically designed to optimize stock portfolios. They can make thousands or even millions of trades per day without human intervention.
- Banks are also using chatbots to give information about their services and offerings and to customers so that they can easily handle transactions that do not require human intervention. They are also using AI to improve their decision-making for loans, and to set credit limits and identify investment opportunities.
- 17. Smart homes:** More and more tech savvy people today are increasingly using numerous smart devices like smart locks, switches, refrigerators, etc.

erators, washing machines, etc. to experience the comfort of a smart home.

Smart devices that make homes smart, learn user's behaviour patterns to save their time and resources, and money by implementing cost-optimized operations to provide a smarter way of living. For example, thermostats and building management systems can be used to automate building heating and cooling. These systems learn and can then predict when to turn on/off the boiler based on weather conditions inside as well outside the house.

18. **Travel:** Travel companies are using AI to provide customer support, analyse historical and real-time data to determine prices to different destinations, evaluate the level of traffic on the roads to give an accurate estimated time to reach a particular point that too through different modes of travel, including bus, trains and flights.
19. **Space exploration:** Space expeditions and discoveries require analysing vast amounts of data using machine learning and deep learning techniques. For example, astronomers used AI to interpret data obtained by the Kepler telescope to identify a distant eight-planet solar system.
AI was used by NASA's rover mission to Mars to perform investigations on Mars.
20. **Mining:** In mining, robotics is used in applications like robotic dozing, excavation and haulage, robotic mapping and surveying, robotic drilling and explosive handling, etc. A mining robot can easily navigate flooded passages and use cameras and other sensors to detect valuable minerals. Robots can also be used to protect humans from any harm or injury in the excavation process by detecting gases and other materials that are a potential threat to humans.
21. **Education:** Robo-readers are used in schools to evaluate answer sheets and grade them. Students and teachers are also notified about the topics in which a particular student is doing well as well as topics in which he/ she needs more practice.

Banking organizations use AI to improve their decision-making for loans, and to set credit limits and identify investment opportunities. From microbiological research to archaeological advancements, all these tasks that were difficult earlier have become much easier today

due to AI. And the scope will only grow with time. Applications that utilize Virtual Reality (VR), training simulations and exploration technologies are already paving the way for building a better future.

22. Law: In the legal field, AI is used to automate labour-intensive processes to save time and improve client service. Machine learning techniques can be used to describe data and predict outcomes. In addition, computer vision can be used to classify and extract information from documents and NLP can be used to interpret requests to provide the requested information.

23. Retail and manufacturing: Even in showrooms, AI is being used to provide virtual shopping capabilities that offer personalized recommendations, make the customers aware about different payment options available to them, manage stock, etc. AI units deployed in manufacturing units can be fed with data streaming from IoT devices to forecast expected load and demand.

In the manufacturing sector, robots are used for cutting, welding, assembly, disassembly, organizing printed circuit boards, packaging, labelling, palletizing, product inspection, testing, colour coating, drilling, polishing and handling materials.

AI can be used everywhere where there is a need to increase productivity and profitability by reducing human efforts. The industrial robots are accurate, flexible, provide less expensive labour cost, generate less noise, give less defects, minimize damages and considerably improve productivity.

1.7 Robotics—an Application of AI

Robotics is a branch of engineering that involves the conception, design, manufacture and operation of robots. It is an inter-disciplinary field that includes the study of electronics, computer science, artificial intelligence, mechatronics, nanotechnology and bioengineering.

Science-fiction author Isaac Asimov, first used the term ‘robotics’ in the 1940s. According to him, robots must ensure the following:

Rule 1: Never harm humans.

Rule 2: Always follow instructions from humans without violating rule 1.

Rule 3: Protect themselves without violating the other rules.

Some important aspects of robots include the following:

1. They have electrical components for providing power and control the machinery.
2. They have mechanical construction and their shape or design depends on the task they are intended to accomplish.
3. They are programmed. Instructions fed to them helps them to determine what, when and how it would do something.

These days, robots are being extensively designed as bots that explore the Earth's harshest conditions and assist in almost every facet of healthcare. Though the robotics industry is still evolving, we already have robots that work in the deepest depths of our oceans to the highest heights of outer space. They are being used to do everything that humans could not dream of doing.

1.7.1 Types of Robots

Pre-programmed robots operate in a controlled environment to perform simple, monotonous tasks. For example, a mechanical arm on an automotive assembly line that is used to weld a door, or insert a certain part into the engine. This type of robot can perform that task for longer hours, faster and more efficiently than a human.

Humanoid robots look like humans and/or mimic human behaviour. They can perform human-like activities (like running, jumping and carrying objects). Sophia and Atlas are two popular examples of humanoid robots.

Autonomous robots operate without human operators. They are designed to perform tasks in open environments that do not require human supervision. For example, the Roomba vacuum cleaner uses sensors to roam freely throughout a home to clean it.

Teleoperated robots are mechanical bots controlled by humans. They are used in extreme geographical conditions, weather, circumstances, etc.

For example, human-controlled submarines used to fix underwater pipe leaks during the BP oil spill or drones used to detect landmines on a battlefield are examples of teleoperated robots.

Robots are also used for tasks varying from diffusing bombs to performing surgeries. VR robots are also becoming popular these days.

Augmenting robots either enhance current human capabilities or replace the capabilities a human may have lost. Robotic prosthetic limbs or exoskeletons used to lift hefty weights are examples of such robots.

1.7.2 Uses of Robotics

Robots are used in the following areas.

1. Helping fight forest fires
2. Assisting humans in manufacturing plants (known as co-bots)
3. Provide companionship to the elderly
4. Work as surgical assistants
5. Deliver food orders or other packages
6. Perform household tasks like vacuuming and mowing the grass
7. Locate and transfer items throughout warehouses
8. Perform search-and-rescue missions after natural disasters
9. Landmine detectors in war zones

Industry-wise applications can be given as follows:

The manufacturing industry is amongst the first well-known users of robots. They use robots and co-bots (bots that work alongside humans) to efficiently test and assemble products, like cars and industrial machinery.

Logistics companies use robots in their warehouses to perform tasks like shipping, handling goods and ensuring quality control. Robots are supposed to take items off the shelves, transport them across the warehouse floor and package them. They are now also used for last-mile delivery of packages for faster and efficient delivery.

Self-driving cars are a result of integrating data science with robotics. Automakers like Tesla, Ford, Waymo, Volkswagen and BMW are all working to provide users an ultimate experience of travel that will let them sit back, relax and enjoy the ride. Companies like Uber and Lyft are also developing autonomous rideshare vehicles that will be operated without humans.

The healthcare industry uses robots to perform complicated surgeries, deliver everything from medicines to clean linens.

Space agencies like NASA use robots in different ways. Robotic arms on spacecraft can move large objects in space. Robotic spacecraft can visit other worlds like the Moon or Mars. For example, Mars rovers Spirit and Opportunity are robots. Cassini studies Saturn and its Moons and rings. Robots like The Voyager and Pioneer spacecraft are now travelling beyond our solar system. People on the Earth use computers to send messages to the spacecraft. The robots have antennas that pick up the message commands and work as per the instructions given to it.

Robotic airplanes can fly without a pilot aboard. NASA is also developing Robonaut to help people in space. The upper body of Robonaut looks like a person. It has a chest, head and arms. They can work outside a spacecraft and work like an astronaut on a spacewalk.

NASA is also working on robots that might help an astronaut in an emergency. For example, when the astronaut is seriously hurt, a doctor on Earth could use the robotic arm to perform surgery. This technology can help doctors on Earth, as well. Doctors can help people in faraway places where there are no surgeons to perform complicated surgeries.

Robots are also used as **scouts** to check out new areas to be explored. They take photographs, measure the terrain, look for dangers and find the best places to walk, drive or stop. This not only helps scientists and engineers make better plans for exploring but also helps astronauts to work more safely and quickly.

Note:

- ASIMO is a humanoid that has the ability to recognise moving objects, postures, gestures, understand its environment, and interact with humans.
- Pepper is the world's first robot capable of recognising human emotions. It is social, can converse with people, give them directions and even dancing with them
- Romeo helps with everyday tasks, assist when people have fallen over, make conversations and play games.
- Buddy is designed to entertain the family, help you with your everyday activities, offer reminders, provides recipes in the kitchen, make video calls, keep an eye on your home while not at home, connect all your smart home devices together and even help your children learn.
- Panasonic Egg uses NLP Robot to communicate with you. It's an intelligent assistant that can be controlled by your voice, play video footage via a built-in projector and even engage in interactive games. This robot is Wi-Fi connected and promises software updates in future to improve it further.
- REEM is a full-size humanoid service robot that can act as a receptionist, provide entertainment for guests, make presentations and give speeches in different languages and help with a variety of different chores. REEM is able to self-navigate, interact with people it encounters and keep on running for up to eight hours.

1.8 Drones Using AI

The term 'drone' generally means any unpiloted aircraft that operates using a combination of technologies including computer vision, artificial intelligence, object avoidance tech and others. Also known as 'unmanned aerial vehicles' (UAVs), these drones can be as large as an aircraft or as small as the palm of your hand. With drones becoming readily accessible, they are now increasingly being used for the most dangerous and high-paying jobs that are discussed below.



Credit: Slavoljub Pantelic / Shutterstock

FIGURE 1.11

Emergency Response

Drones outfitted with thermal imaging cameras are used by emergency response teams to identify victims who are difficult to spot with the naked eye.

Humanitarian Aid and Disaster Relief

During times of natural disaster, drones are used to assess damage, locate victims and deliver aid. In addition, they are also used to prevent disasters altogether (refer **Fig. 1.14**).

Surveillance

Drones outfitted with thermal imaging cameras can be used to monitor and combat forest fires. Thermal cameras can measure and detect abnormal forest temperatures. This information can then be used to identify areas that are more prone to forest fires or identify fires just minutes after they begin (refer **Fig. 1.13**).

Conservation

Poaching and activities resulting in climate change have adversely affected wildlife worldwide. In fact, according to the World Wildlife Fund, thousands of species are estimated to extinct each year. To help combat

this trend, conservationists are extensively using drones for geospatial imagery to monitor and track animals to protect our biological ecosystem.

Disease Control

Many infectious diseases spread through animals. In such a scenario, drones can be used to capture and test mosquitoes for infectious disease. This initiative can not only protect local residents, but can also prevent epidemics before they begin.

Moreover, drones are also being used in remote areas to provide quick access to vital drugs, medicines and medical equipment. All these initiatives have a profound impact on preventing disease, increasing life expectancy and raising general standards of living.

Bomb Detection

Small sized drones fitted with effective cameras can easily penetrate into constricted spaces to detect live bombs and save lives of thousands of people.

Air Strikes

Drones are used for conducting air strikes. Former US President Barack Obama had been using drones regularly to attack militants in the tribal areas of Pakistan. While being controlled by the defense personnel, drones can be made to fly around suspected areas to fulfill military operations (as shown in **Fig. 1.12**). However, use of drones for military operations has also raised numerous moral and ethical concerns as they lack accountability and failure to fully grasp the consequences of actions.



Credit: Alex Yuzhakov / Shutterstock

FIGURE 1.12 Using Drones for Air Strikes

Agriculture

Farmers in some advanced countries are extensively using drones to gather data, automate redundant processes and improve efficiency to reduce costs and expand yields. Drones also help farmers to predict their potential harvest.

Weather Forecasting

Scientists are using drones to collect data about temperature, humidity, wind speed and other climatic parameters that could help them to accurately predict future changes to global weather systems. Drones used as autonomous sailboats are used to collect oceanic and atmospheric data from the ocean surface.

Maritime

Drones are used to inspect ships above the surface as well as hulls from below. Countries like the Netherlands, Denmark, and Norway are already using drones to identify ships committing emissions infractions.

Waste Management

Drones are being used to clean oceans, collect waste in ports and harbours and maintain systems for wastewater management.

Energy

Drones are used by energy generating companies to set up new sites for the production of energy. For this, the drones are made to survey areas and gather topographic detail that can be used to help oil and gas companies identify new drill sites. Drones are also used to extract, refine and transport oil and gas while ensuring compliance with regulations and standards.

Drones fitted with specialized thermal sensors can detect leaks faster than a human inspector. Even companies generating solar energy can use drones to design configurations for new arrays.

Mining

Mining activities require constant measurement and assessment of stock-piles of ore or rock or minerals. Drones fitted with unique cameras can capture large amounts of data from the air, thereby reducing the risks associated with having surveyors on the ground.

Drones are being using by the Indian government to deliver COVID vaccines even in the remotest areas.

Construction Planning

Drones are used to improve construction planning, enhance project monitoring and site management. Cameras fitted in these drones monitor buildings and gauge topography and soil type throughout the construction lifecycle.

Urban Planning

With increasing urbanization, cities are overburdened to accommodate more people in already congested spaces. In such a scenario, urban planning is being done with drones to implement data-driven improvements. For example, drones are used to gather data in populated areas. This data when analysed using ML algorithms can suggest areas that may get bene-

fitted from green space or classify the different types of structures and regions on the map.



Credit: 102 / Shutterstock

FIGURE 1.13 Drones for Surveillance

Personal Transportation

China-based EHANG was started as an autonomous aerial vehicle (AAV) that operates with four rotors (quadcopter) for vertical takeoff. The vehicle is used to help passengers reach their destinations. Such a vehicle is especially very useful in an urban environment with plenty of obstacles. Personal transportation drones require minimal inputs from the passenger and aims to allow safe landings even in case of engine failure or a collision.

Big companies like Uber, Airbus, Boeing and Rolls-Royce are working on developing flying drones (robotaxis) for ferrying passengers around.

Space

Drones are being used to be moved in the space. For example, NASA is using a drone-like helicopter in its Mars 2020 mission to help look for signs of life on Mars. In this mission, the helicopter will act as a scout for the rover, gathering data about the planet's terrain and surveying areas the rover cannot reach.

NASA is also using a nuclear-powered drone for exploring Titan, one of Saturn's moons. The drone will arrive on Titan by 2034, autonomously traverse the planet for about 2 years, taking photos and sending data back for analysis.



Credit: koya979 / Shutterstock

FIGURE 1.14 Drones for delivering medical help

Telecommunications

Telecommunication towers must be inspected frequently to ensure service reliability. However, this process is too dangerous and time-consuming to do manually. In such a scenario, drones are able to quickly assess damage to help guide repair teams in restoring service.

Internet

Drones are being used to provide Internet access in remote areas and over irregular landforms. For example, Facebook has designed a solar-powered drone called Aquila, to provide Internet access to rural parts of the world. However, in 2018, Facebook halted the use of Aquila and used only third-party drones instead.

Even SoftBank, in collaboration with the drone manufacturer AeroVironment, is planning to develop drones that will operate in the stratosphere to serve as ‘floating cell towers’ to provide internet service to customers.

Outdoors

Drones are used outdoors to perform aerial landscape photography and extreme sports footage and map the entire mountain face to help climbers and skiers to better understand the terrain.

Tourism and Hospitality

These days, flying drones providing luxury accommodations can be used to travel to new locations on demand or to remote and traditionally inaccessible locations for guests. They are also being used to deliver packages and room service quickly.

Live Entertainment

Drones are already being used by Disney for entertainment through synchronized lights shows, floating projection screens and as drone puppeteers.

Journalism and News Coverage

News companies are using drones to gather news, especially from areas that are difficult to visit due to safety issues, high costs or physical barriers. For example, drones are used to get aerial footage of the aftermath of hurricane, wildfires and assess flood-ridden areas in the midwest.

Food Services

Drones are being used by online food ordering and delivery services for faster, cheaper delivery. This helps restaurants downsize their physical locations and lower real estate expenses.

1.9 The Future of AI

Accenture presented a report, AI: Built to Scale, in which it was concluded that 84% of business executives believed that they required AI to achieve a higher growth but 76% struggled to use AI across their business. When asked about the future of AI, Dr Kai-Fu Lee, AI oracle and venture capitalist said, in 2018, '[AI] is going to change the world more than anything in the history of mankind. More than electricity.'

Today, the most important topic that restrains companies to go deep into AI is cybersecurity. Spike in cybersecurity breaches throughout 2020 when attacks rose 600% is a big concern. During the pandemic, hackers capitalized on people working from home, using less secure technology

and Wi-Fi networks. In such a scenario, AI and machine learning can be used to identify and predict potential threats and catch instances of fraud especially those involving finances. Moreover, every application, these days, collect a lot of data about users. A person with disruptive mindset can train a machine to even threaten user(s) with solutions that can even endanger their being.

AI is also considered to be a threat to privacy as it violates human rights. For example, voice assistants and facial recognition techniques invade privacy of humans and probably eavesdropping into their lives. Thus, AI needs to evolve as a trustworthy system in the future.

AI will change the way managers search suitable candidates for a job. AI is already being used in the hiring process so much that up to 75% of resumes are rejected by an automated Applicant Tracking System (ATS) before even reaching a human being.

In the past, recruiters have to invest considerable time to analyse resumes and shortlist relevant candidates. But now, this is easily done by AI-powered programs. However, the concern is to develop AI systems that are not biased. HireVue is a startup (established in 2018) that used facial recognition software and psychology to determine the potential effectiveness of a candidate in a certain role. But after the Electronic Privacy Information Center filed a lawsuit alleging that it could perpetuate bias and prejudice, HireVue had to discontinue use of facial recognition software in early 2021, and now uses audio analysis and natural language processing. This clearly indicates that the use of AI in the hiring process or any other process may become controversial.

As of now, AI systems like **Jobscan** is an excellent resource that provides similar resume scanning by comparing a candidate's resume to a job description, thereby performing better than the ATS. Similarly, **Jobseer**, a browser add-on, is an AI-based tool which scans a resume, as well as keywords and skills related to the desired jobs, to match with the job listings that best fit a candidate's experience. For each listing, the candidate gets a rating as well as recommendations of skills to be added to improve the capabilities.

AI and machine learning are one of the most desired skills in today's job market. Jobs demanding AI or machine-learning skills are expected to increase by 71% in the next five years. Even an understanding of how AI can be used in different sectors is considered to be an added skill. Some facts and myths that prevail about this technology are listed in **Fig. 1.15**.

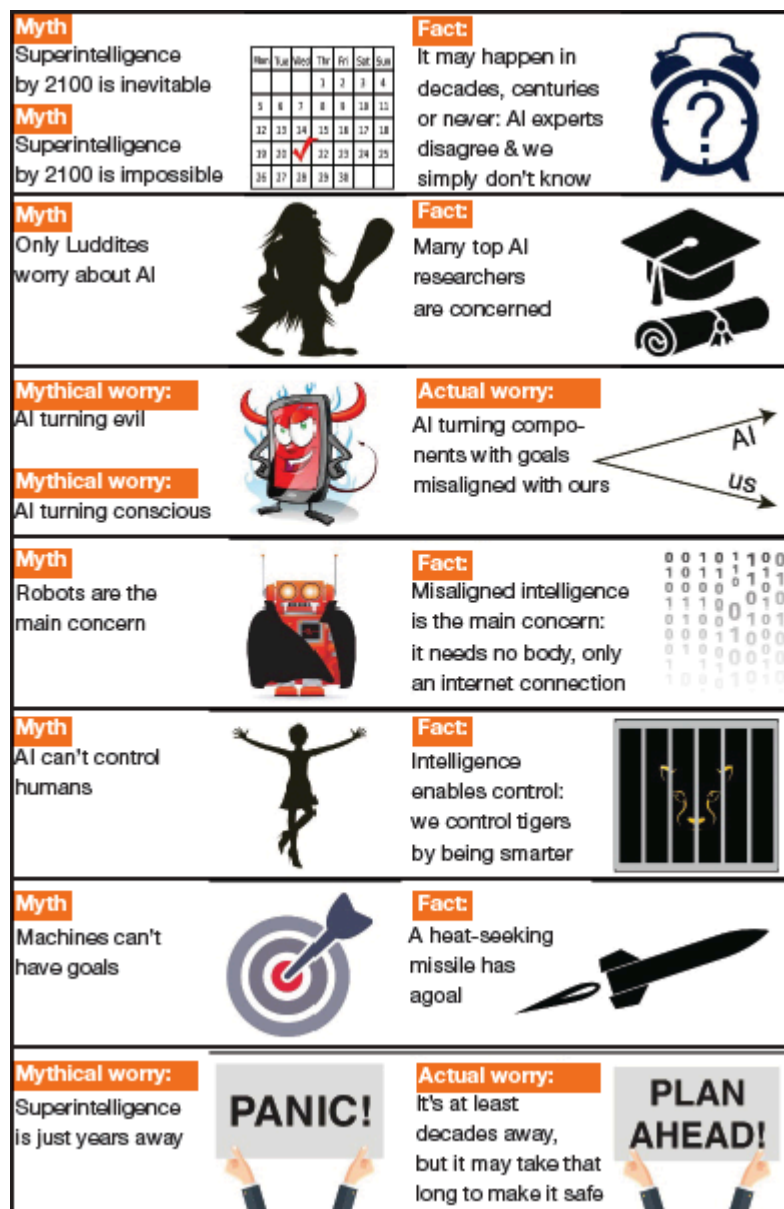


FIGURE 1.15 Interesting Facts and Myths about AI

AI technology is still evolving. Companies would strive to implement ethical AI or responsible AI. Business organizations are becoming keener to earn better revenues and profits. Small businesses are considering AI solutions of larger companies as a case study to understand how they can delve into it at a lower cost for maximizing their success. Startups are coming up to offer agility, growth and better ideas to stay competitive.

In a few years' time, advances in AI will reach the superintelligence stage. But it should be adopted without harming the goodwill of loyal people. No doubt, the future of artificial intelligence is exciting and extremely promising.

In fact, recently, a paper published by researchers at AMOLF's Soft Robotic Matter group, discussed self-learning robots that can easily adapt to changing circumstances. For this, small robotic units were connected to each other for them to learn to move on their own. Techniques like reinforcement learning and generative adversarial networks will be in great demand. Moreover, sustainable technologies for fighting climate change will be developed using AI. The term already coined for this is 'Green AI'.

To conclude, AI as a technology is not at all a threat to human existence. But the real threat is from humans who may misuse AI to harm an individual or a society. For example, robotic army can be fed to realize harmful motives. In order to alleviate the potential risks of AI in the future, AI systems must be ethical and responsible.

1.10 No Code AI

Right from the first generation of computers, exhaustive efforts have been made to make programming easier, faster, less technical so that it can become accessible to a much larger audience.

No-code tools allow people to build applications and systems without having to write long and complex program codes. Applications can now be designed using visual interfaces and guided user actions. No-code tools are pre-integrated with other tools to exchange information as needed. Given below are some applications that can be built entirely using such tools.

1. Websites and landing pages with Webflow.
2. Web or mobile applications with Bubble, Adalo, Mendix or Thunkable

Number and applications of No Code AI tools will grow steadily

3. Chatbots or virtual assistants through Octane AI, Kore.ai, Landbot or mindsay

4. Databases through Airtable
5. Connecting your tool stack with Zapier, tray.io, Integromat, Parabola or Paragon
6. E-commerce through Shopify or Weebly
7. Manage memberships with Memberstack
8. Newsletters with Mailchimp or Mailjet

1.10.1 Why No-Code AI?

It has been seen that majority of businesses struggle to implement AI to its full potential and scale as it is still considered to be a core-engineer's task. In such a scenario, use of no code AI tools can be a great help due to the following reasons:

1. These tools allow automation through plug-and-play or drag-and-drop UI.
2. Even users with absolutely no coding skills can leverage the benefits of AI algorithms to solve business issues.
3. Businesses that cannot afford to invest additional time and resources to develop AI applications can build such systems from the ground up.
4. **Improves accessibility** to intensify use of data science or AI by even small and mid-sized companies.
5. **Enhances usability** as more and more non-technical users can create AI based solutions to a problem in low-cost and less-time.
6. No-code AI solutions are of fairly good quality and reduces human error(s) when setting up such systems. Refer **Fig. 1.16** to know about some No Code AI Tools. **Figure 1.17** list their applicabilities.

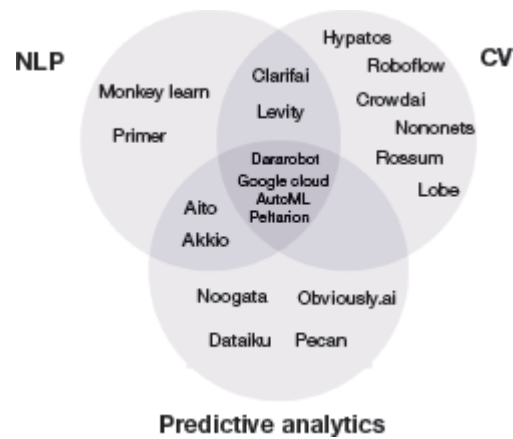


FIGURE 1.16 No-code AI landscape 2022: Technology

	Text	Image	Docs	Video	Audio	Tabular
No Code						
Levity	✓	✓	✓			
Aito	✓					✓
Akkio	✓					✓
Lobe		✓				
Monkey learn	✓					
Obviously.ai						✓
Pecan						✓
Low code						
Clarifai	✓	✓		✓	✓	
Crowdai		✓			✓	
Dataiku						✓
Dararobot	✓	✓		✓	✓	✓
Google cloud AutoML	✓	✓		✓		✓
Hypatos			✓			
Nononets			✓			
Noogata						✓
Peltarion	✓	✓			✓	✓
Primer	✓					
Roboflow		✓		✓		✓
Rossum			✓			

FIGURE 1.17 No-Code AI landscape 2022 : Data type

When running this exercise, it made sense to group the tools by the technology they use. Here's a quick round of definitions:

Computer Vision

It allows machines to obtain information from digital images, videos, pdfs and other visual data, and take actions based on their learnings.

NLP

This allows machines to understand and process language both spoken and written, for example, text messages.

Predictive Analytics

This refers to predictive modelling based on structured (i.e., tabular data), for example, predicting churn rate, forecasting and stock prices.

We also made a distinction between no-code tools and low-code tools. The no-code tools follow basic criteria: they are end-to-end tools that are usable without coding knowledge. In that sense, low-code tools are better suited if you have someone on your team who *speaks data*. **Figure 1.18** categorizes these tools based on their usage.

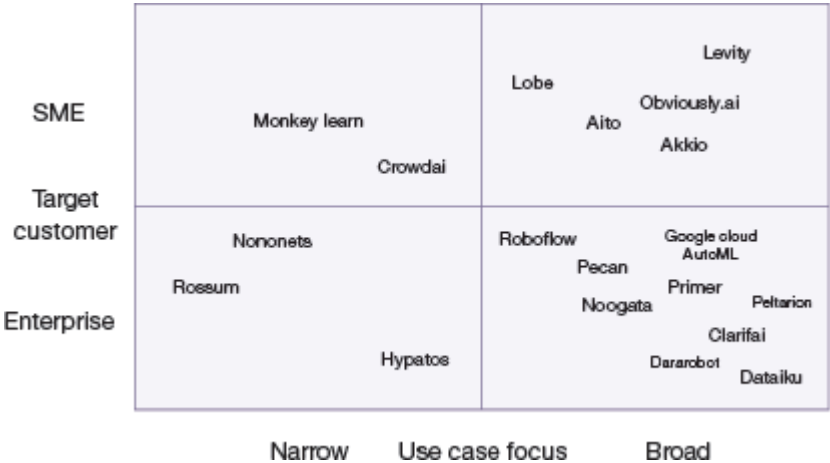


FIGURE 1.18 No-code AI landscape 2022: Customer vs use case

Let us look at some popularly used no-code tools (**Table 1.3**).

TABLE 1.3 No-Code AI Tools and their Usage

Tool	Description
Aito	Performs predictive analytics and NLP automation Has a simple UI Furnished with APIs that integrate well with many automation platforms Focus on tabular datasets Offers automated re-training system Usually designed for sales, marketing and financial activities
Clarifai	An NLP and Computer Vision tool Modelling unstructured data for the entire AI lifecycle Offers image, video and text recognition solutions
Crowd AI	Uses Computer Vision for both images and video Used for applications like vegetation management or disaster responses
Levity	Focuses on image, text and document classification Enables users to train custom models on their use-case-specific data automatically learn from user interactions Easily integrates with all the tools
Google AutoML	Trains high-quality models specific to business needs AutoML Image for image annotation and labelLing AutoML Video for video annotation and labelLing. AutoML Text for semantic analysis and text classification
Lobe	Free, private desktop application by Microsoft Used for image classification, with object detection, and data classification coming soon Has pre-trained solutions (e.g., offers emotional reactions to enable the program to react to different expressions) Allows people to send emoji reactions using just their faces
MonkeyLearn	Performs text analysis Facilitates data visualization Supports unstructured text-based data Used to get topic, sentiment, intent, keywords, etc. Also used for automatically tagging business data, visualizing actionable insights and trends Provides simplified processes for text classification as well as extraction
Nanonets	Used in the domain of Computer Vision Used to build an ID card verification model for any country, format or language
Noogata	Predictive analytics tool Helps to customize your model Used for making data-driven decisions
ObviouslyAI	Uses NLP processing to perform tasks on user-specific text data. DragS and drop CSV file or integrate with HubSpot, Salesforce, or MySQL for getting predictions Performs tasks like forecasting company revenue, optimizing supply chain, personalizing marketing campaigns, etc.
Primer	NLP model builder Powerful integrations with other tools Offers pre-trained models that are ready to be used Offers features to visualize your model performance in one go
Apteo	Helps to make more repeat sales AnalySes customer database to find out which products certain customers are

	likely to buy next Using this information, crafts better messages for more accurate targeted upselling It integrates with ecommerce tools like Shopify, Stripe and Square
Prevision	Increases the productivity of data science projects Focused on AI modelLing Users can upload data, train the model, analyse performance and then get predictions
causaLens	Finds cause-and-effect relationships in data Uses an innovative approach to AutoML
Causaly	Same as causal AI field, but is focused on finding causal relationships in biomedical science research Uses machine learning to analyze over 30 million papers, clinical trials and side effect databases
PredictNow.ai	A vertical specific no-code AI solution focused on financial machine learning Computes the probability of profit for your next investment
Accern	Functions for credit risk, ESG investing, quantitative research, financial crimes analysis, etc. For all applications involving heavy-duty finance needs
RunwayML	Made for creators and creatives, with features for image, video and text data Allow users to create synthetic images and videos, cut objects out of videos and more
IBM Watson	Helps in model creation, training, and deployment, without any serious integration features
MLJar	Highly focused on modelling, automating feature engineering, algorithm selection, documentation and explanations
Dataiku	Supports visual AutoML functionality More focused on general data science tasks like data integration, building data pipelines, data visualizations and statistical analysis Provides users with a drag-and-drop interface for more advanced users, Dataiku also provides the ability to code in Python and R within the platform Offers a wide range of tools for data preparation, machine learning and deployment For data preparation, Dataiku offers an interface for ETL tasks like data cleaning, transformation and enrichment
Noogata	Has pre-built AI blocks Used for predictive maintenance, fraud detection, and customer segmentation. Users can simply drag and drop these blocks to create an AI solution
Kinetix	Uses no-code AI to create high-quality 3D animations Users provide any video, which gets converted into 3D animated avatars In the Web3 world, these avatars can then be used for a variety of purposes, such as in video games, social media or even as customer service agents
Loris	Provides conversational AI software to help human agents make customer support more human Help businesses automate customer support tasks with a high-touch, human experience Crafts real-time suggested responses based on the customer's tone and sentiment
Neptune Software	Uses AI-augmented workflow to support users in creating apps, either from scratch or from a library of customizable templates and components
Evisort	Embeds AI into contract management software

Webio	Helps businesses automate the contract management process, from data extraction to analysis and approvals Uses AI to understand the contents of contracts, as well as to identify risks and opportunities Allows businesses to quickly use conversational AI Integrates with SMS, WhatsApp and other channels, to allow communication with business customers

1.10.2 Future of No Code AI

Businesses are steadily moving towards no-code platforms. Research shows that by 2024, nearly 65% of application development will be done through either through low-code or no-code platforms. And, no code AI will play a big role in this. Some advantages of these tools are as follows.

1. No-code AI is a code-free technology.

Google AutoML is the no-code AI solution that provides a feature-rich suite of AI products that helps to train high-quality ML models tailored to unique business needs.

2. It facilitates non-AI experts also to implement and test their ideas without any help of AI experts
3. AI experts can create ML solutions in less time using minimum efforts.
4. It supports collaboration between AI experts and domain experts.
5. It has drag-and-drop or wizard-based interface to build applications.
6. It helps to solve problems with improved productivity and efficiency.

All in all, no code AI ensures that **AI technology can be adopted by everyone and everywhere.**

1.10.3 Why No-Code AI Must be Used?

No code AI (refer **Fig. 1.19**) will become more important as it solves the following problems.

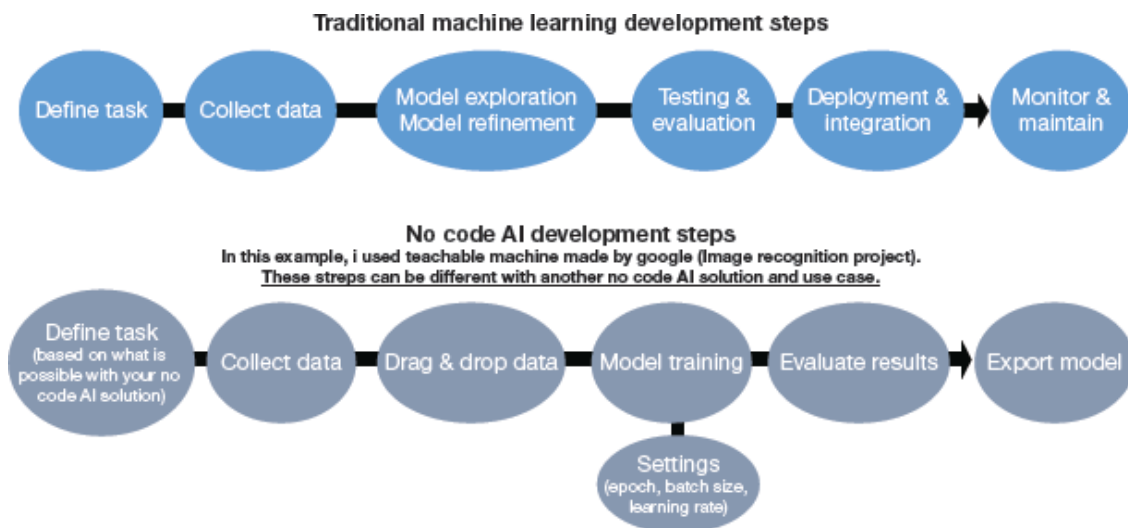


FIGURE 1.19 Steps in a Traditional ML Model Development and in a No-Code AI Model Development

Gap Between AI Experts and Domain Experts

Every year, more and more companies are adopting AI technology. There is a huge gap between AI experts and domain experts. They rarely understand terminology used by others. While AI experts talk using terms of machine learning (ML) models and optimization algorithms, domain experts, on the other hand, use only business-specific words to **solve their problems**. This gap is critical to the success of AI initiatives. Many projects fail due to lack of understanding. In such a scenario, no-code AI platforms allow domain experts to test their ideas and communicate their issues with AI experts in a better way. At the same time, these tools also help AI experts to add value for the domain experts in less time with less effort.

We can categorize no-code AI platforms based on the type of interface: drag-and-drop or wizard-based.

Limited AI Experts and More Problems

We know that many problems can be solved fully or partially using AI technology. But to solve them, we need AI experts. These experts are still

less in number and charge a heavy fee. So, here, domain experts can run their experiments and get AI solutions using No-code AI platforms.

Steve Jobs said, ‘the line of code that’s the fastest to write, that never breaks, that doesn’t need maintenance, is the line you never had to write.’

It is interesting to note that only 0.25% population of the world can do coding but anyone can be taught to quickly and easily use no-code tools.

1.11 Low Code AI

The use of no-code as well as low-code platforms is constantly on the rise. Today, the growing demand for AI solutions in business applications has paved the way for low-code and no-code AI tools. These tools give companies the flexibility and agility to create new applications.00:00

Low-code is a software development technique that was introduced in 2011 to promote application development in less time with little to no coding required. For this, it supports visual development of applications using intuitive modeling with a graphical user interface (GUI).

Low-code development platforms are based on the concepts of model-driven design, automatic code generation and visual programming. These platforms provide integrated tools to eliminate the need to write code line-by-line. When working on these platforms, the user draws flowcharts in a visual editor, and the code is produced automatically. Users can even create software applications by dragging and dropping pre-defined components. Today, some of the most popular low-code software platforms that are used by both programmers and non-programmers are Mendix, Outsystems, Creatio, Appian and Creator.

According to Gartner, the demand for information systems will increase five times faster than IT departments’ ability to provide them because the number of employees is not growing at a sufficient pace.

Lower the difficulty level of application development processes, broader is the audience. Low-code solutions not only enables fast delivery of applications with minimum effort but also ensure least possible effort for the installation and configuration of environments, training and implementation. Apart from other reasons, low-code AI is also preferred as it provides built-in security and maintenance, saving costs and time. It also lowers development risks at a high return on investment.

The key differences between traditional and low-code development are given in **Table 1.4**.

TABLE 1.4 Differences between traditional and low-code development

Feature	Traditional Tool	Low-Code
Time to market	Slow to launch, and is slow to change	Quick to launch
Uses	Manual coding	Drag and drop design, pre-built components
Scope	Planned and developed as large, complex projects	Small and independent solutions
Development	Difficult to align to business demands and has slow turnaround times	Allows up to 5x quicker building, efficient testing, and higher ROI.
Maintenance	Requires additional development and involves expensive support	Development enables easy and quick updating or extending of the solution. This makes it excellent for prototyping
Integration	Software integrations are time and investment-intensive, documentation is required, as well as extensive testing phases	Offers pre-built connectors live debugging and creating web services with no coding
Deployment	Slow and complex; it requires multiple steps and development resources	offers one-touch deployment, an application is sent to production in just a single click
Low-risk/ high ROI	High risk is involved	robust security measures, data integration and cross-platform support are already built-in and can be easily customized

1.11.1 Low-Code vs No-Code AI Development

Low code development requires very less or minimal programming effort from the user, whereas no-code software does not require any program-

ming at all.

Though both low-code and no-code use building blocks to create applications, low-code facilitates the integration of custom software in the form of such building blocks.

1.11.2 Who Uses Low-Code Development?

Low-code development is beneficial for different types of users to speed up their projects and simplify the deployment process. Low-code AI platforms enable rapid development, testing and scaling of apps. In this section, we will discuss who use these platforms and for what purpose.

Low-Code for AI Beginners

Low-code and no-code AI platforms make AI technologies more accessible to technical and non-technical users. Businesses no longer need to hire a full-time AI developer. These platforms have made it much easier for beginners to get started with AI technologies.

Low-Code for Developers

Low-code platforms help AI developers (like Computer Vision Engineer) to develop AI applications, add their own custom software or AI models as integrations to the platform. This helps programmers and researchers to spend less time on repetitive coding tasks, write less code and focus resources on enhancing and optimizing the functionality of the application.

Low-Code for Researchers

Researchers can use low-code platforms to quickly develop prototypes, speed up their experiments and test each change without the need to invest in expensive infrastructure. Moreover, they can easily compare different approaches and benchmark multiple versions.

1.11.3 Low-Code AI Platform for Computer Vision

A low-code AI platform for automating computer vision applications is a difficult task. Lot of research is done in this area to overcome the chal-

allenges and technological problems. The latest AI low-code platform, specifically designed for computer vision applications is, viso.ai.

viso.ai is used to create custom computer vision and deep learning applications that process video feeds of numerous cameras in real-time with deployed AI algorithms. Using fine-tuned, pre-trained AI algorithms and pre-built modules in a visual editor, viso.ai quickly creates a custom AI vision solution. The platform continuously optimizes AI applications to achieve progressively better results.

1.11.4 Components of Low-Code AI Platforms

A low-code AI platform includes the following elements:

Graphical User Interface (GUI)

An intuitive GUI with clear visual elements can be easily used non-technical users to construct AI vision applications. The purpose and usage of the element must be easily understandable by the users.

Pre-Built Integrations

The AI vision application may use visual input of cameras and some pre-trained AI models, so these functionalities are available as ready to use building-blocks to the user. Pre-built integrations of modules that ensures secure data storage and communication are available with these platforms.

Application Manager

Developing an application with a low-code AI platform is just one portion of the software development process. The application needs testing, deployment and maintenance on a regular basis. For this, the platform offers tools to update code, remove bugs and improve its functionality and performance.

1.11.5 Disadvantages of Low-Code/No-Code Platforms

Though no-code/low-code AI platforms sounds interesting to use, there are some concerns which should be addressed before taking the final call in selecting a platform.

Security

Some platforms may not be able to design access protocols. In such a case, security may be compromised and this is a serious concern for many companies where security is at the utmost priority. Therefore, it is crucial to read, research and clearly understand the terms and conditions to interpret where and how data will be used.

Lack of Customization

Low-code/no-code platforms offer limited functionality as they are specifically designed to solve a particular problem.

Requires Consultation or Training

All users, technical or non-technical, must be able to use the low-code/no-code platforms. But usually, it is used by an ML engineer who also need lot of training and consultations to guide other team members to solve problems using AI.

Lack of Trust

According to Google Trends, the interest in no code ML is increasing but people interested in traditional ML are far ahead. Support for libraries for ML and computer vision are much more than low-code/no-code AI platforms.

Lock-In Strategy

Considerable amount of switching cost is involved while moving from one vendor to another. Thus, most of the times, a user is entirely dependent on a particular software vendor.

Limitations on Personalization

Some no-code and low-code solutions do not allow users to change certain parameters.

Data Management

Even while using no-code solutions, businesses may have to rely on the expertise of data scientists and data engineers for data processing tasks.

Scalability

As of now, creating scalable solutions for solving complex problems using a no-code machine learning platform is far from reality.

1.11.6 is Low-Code the Future of Software Development?

Low-code AI solutions are having a deep impact on the overall market right now, and its use is quickly expanding. For example, if you want to enhance your existing application, then either you can use a coding language such as Python or a low-code/no-code (LCNC) framework that has pre-designed and tested code blocks that can be instantly incorporated to use its functionality.

The forecast about the use of low-code/no-code development platforms follow a strongly positive curve. By the end of 2024, it is expected that more than 65% of applications will be developed using the low-code/no-code development approach. Also, more than 75% of large enterprises will be using at least four low-code/no-code development tools.

The demand for low-code/no-coding development platforms has increased tremendously and the worldwide low-code development technologies market is estimated to be \$65 billion market cap by 2026. Businesses will benefit from low-code/no-code AI platforms in more data-driven sectors, such as marketing, sales, and finance. AI can help in predicting churn rates, analyzing reports, adding smart suggestions, automating invoicing and for lot more applications.

By 2030, the low-code development platform is expected to generate a revenue of \$187 billion. As compared to no-code tools, low-code platforms provide a better scope of customization as per business requirements.

Chatgpt (Chat Generating Pre-Training)

ChatGPT is a language-generation software that can easily converse with people. For this, it answers follow-up questions, rejects inappropriate queries and even admits its mistakes (Source: OpenAI summary of the language model). To perform its task, ChatGPT has been extensively trained on an enormous amount of text data.

Using NLP techniques, training data and crawling the web to study archived books and Wikipedia, ChatGPT can recognize patterns to create text that mimics various writing styles. Some features of this amazing tool are as follows.

1. Facilitates communication in conversational format.
2. Uses Reinforcement learning techniques to respond more accurately and dynamically.

ChatGPT is developed by OpenAI using GPT-3.5 language.

We know that reinforcement learning learns by rewards and punishments received. In case of ChatGPT, a question is asked from the software. Then, a number of answers are sampled which are later manually ranked by humans. In fact, these ranks serve as training data for the reward model. The output of the reward model is then improved by further training a fine-tuned language model using reinforcement learning to react to queries. Therefore, human feedback plays a crucial role in the success of deploying this software.

Once trained, ChatGPT can respond to follow-up inquiries, acknowledge its errors, refute false assumptions and refuse unsuitable proposals.

Besides conversing, ChatGPT can respond to all types of writing, including theoretical essays, mathematical solutions and stories.

The underlying technique of ChatGPT can also alter how people use search engines by delivering answers to complex problems.

ChatGPT is a good debugging tool and can also fix the bug when you are asking a question.

ChatGPT is still in the research review stage. But still users can sign up and test it out for free.

Did you know that as soon as Elon Musk learned that OpenAI was using Twitter's database to train ChatGPT, he immediately stopped it from doing so because OpenAI is no longer open-sourced and non-profit, and it should eventually pay for this knowledge.

Chat GPT-3 be used to create more complex chatbots. Those chatbots can provide detailed information or perform difficult tasks like book flights, order food, automate simple tasks (like scheduling meetings, tracking appointments), notify about upcoming events, find local businesses, give personalized recommendations and advice based on preferences. Imagine a fitness freak person getting healthy recipes and exercise routines for a chatbot.

Big Concern: Human Replacement

Most significantly, ChatGPT has demonstrated the ability to build complex Python code and compose college-level essays. This has raised concerns that such technologies may eventually replace human workers like journalists or programmers.

Limitations

1. Chatbot is not a new thing. Several companies including Microsoft have developed them but did not get much success. We have read that in less than 24 hours, Twitter users taught Microsoft's Tay bot misogynistic and racist language. Meta released BlenderBot 3 which also disseminated racial, antisemitic and misleading information, including the assertion that Donald Trump won the 2020 presidential elections.

2. To prevent these kinds of incidents, OpenAI used a Moderation API (an AI-based moderation system) that notifies when language violates the company's content policy to stay away from communicating harmful or unlawful content. However, its moderation still has issues and is not perfect.
3. Like any other AI tool, ChatGPT is only as good as the data it is trained on. If the training data contains biases or inaccuracies, the model may reproduce these biases in its outputs.
4. ChatGPT sometimes write nonsensical answers and fixing this issue is challenging because there may be no source of truth and building more restrictions may result in declining questions that it can answer correctly.
5. ChatGPT gives different results when the questions are either slightly rephrased or asked multiple times.
6. Text generated by ChatGPT excessively uses certain phrases especially when trainers want longer answers that look more comprehensive.
7. Ideally, ChatGPT should ask clarifying questions when the user asks ambiguous queries but instead, it gives answers by guessing what the user intended.
8. Attempts were made to make ChatGPT refuse inappropriate requests, but it sometimes respond to harmful instructions or exhibit biased behaviour.
9. ChatGPT writes long paragraphs, poems, stories, etc. but it has not mastered the art of creative style, so its output has not yet topped best-seller lists. However, the program does write good bedtime stories for kids. But models like ChatGPT can never replace children's books that have been traditionally published.

Conclusion: Chat GPT is a powerful and versatile NLP tool that has the potential to revolutionize the way humans interact with machines.

Wishing you and everyone in your family a very healthy, wealthy, happy and prosperous 2023. Stay abundantly blessed always.



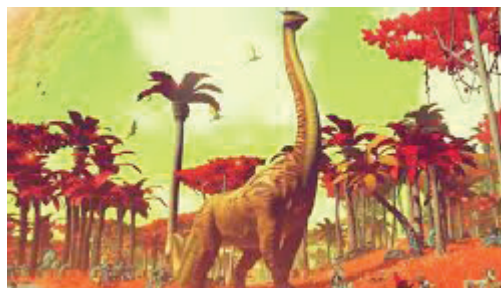
The AI Game – Mystery Animal

Mystery Animal Game is a very interesting game in which the computer pretends to be an animal and the user tries to guess it by using his/her voice.

Rule 1: Users can only ask Yes-No questions such as ‘Are you a reptile?’ or ‘Do you have feathers?’ e ‘Can you swim?’, ‘Are you a carnivore?’, ‘Are you a mammal’

Rule 2: Users have to guess the animal in 20 questions.

Visit the link to play the game. <https://mysteryanimal.withgoogle.com/>



Credit: Screenshot of STORY SPEAKING IN A GOOGLE DEVICE, <https://experiments.withgoogle.com/story-speaker>, is a trademark of Google LLC

AI Lab Session - Talk to Transformer

AI text generator tool, **Talk to Transformer** is used to create human-like text by predicting the next word from the 40 GB internet data (around 8 million web pages). The tool extensively uses Neural Network and NLP techniques (more specifically NLG) by learning from the data and by predicting the context of the previous data. Users just have to type a single line of text and the tool will in turn generate multiple results based on the prediction of what is already written.

Just visit the URL, <https://app.inferkit.com/demo>. Select a topic by clicking the down arrow and see the multiple results you get. Just read the

story. You will feel as if it had been written by humans only.

Fun Activity – Story Speaking in a Google Device

Use the following link to go to story-speaker page.

<https://experiments.withgoogle.com/story-speaker>

Story Speaker lets anyone create talking, interactive stories with no coding required. The story can then be played anytime.



Credit: Screenshot of STORY SPEAKING IN A GOOGLE DEVICE, <https://experiments.withgoogle.com/story-speaker>, is

a trademark of Google LLC

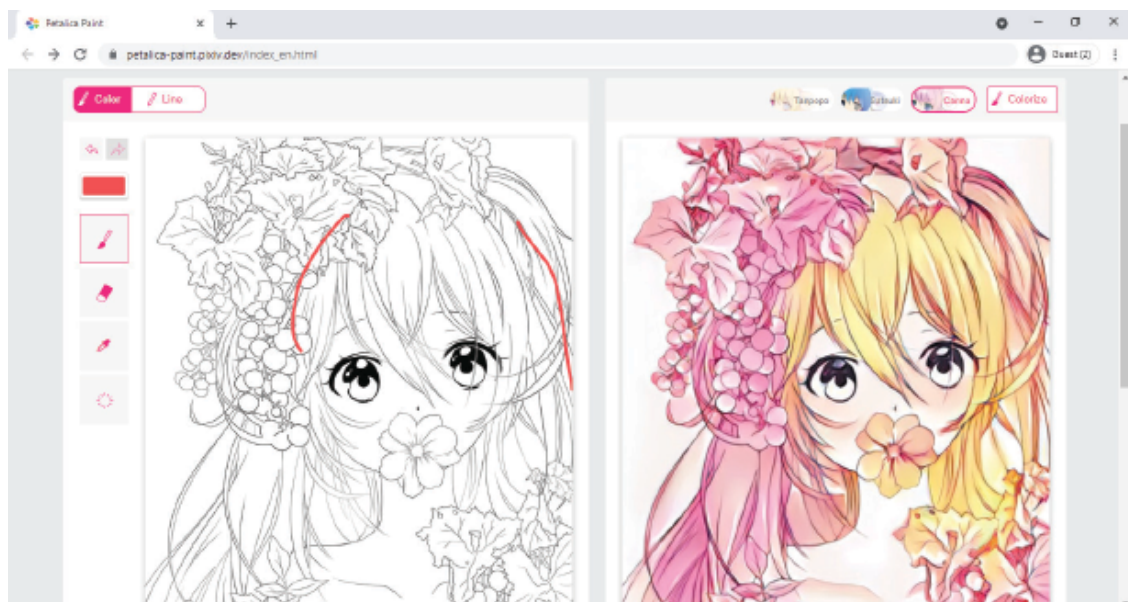
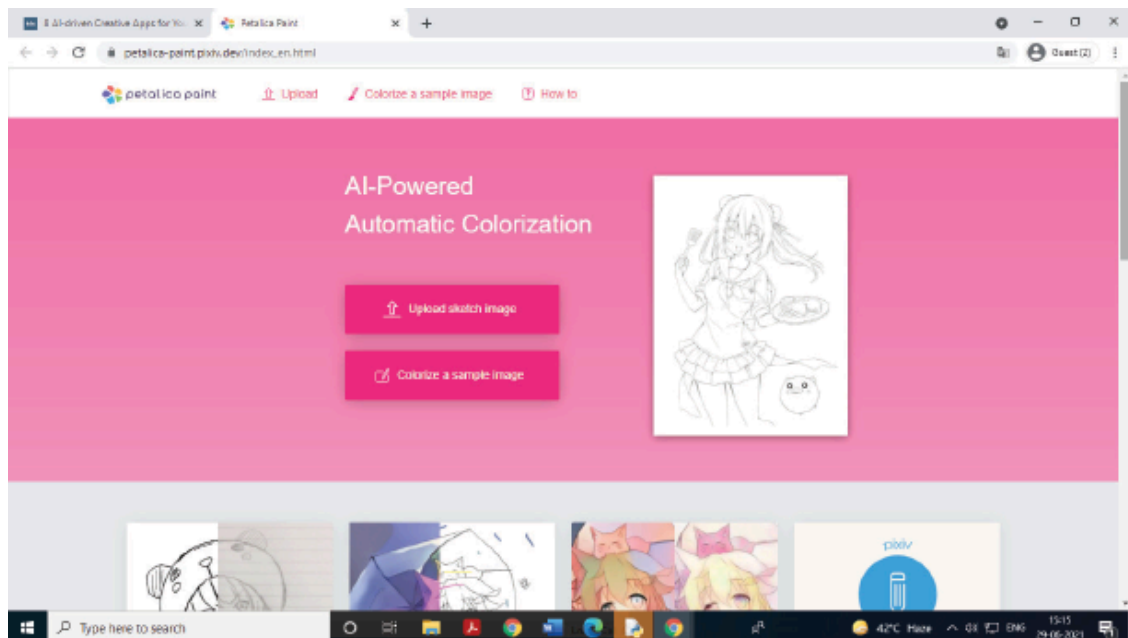
AI Lab Session for Painting Images

Visit the URL, <https://mitmedialab.github.io/GAN-play/>.

The pix2pix application converts images from one style to another using a machine learning model that is specifically trained on pairs of images. For example, you need to input a sketch/ drawing with outlines (or edges), select a model and then translate to see how the resultant image looks when filled with the selected pattern.

AI Lab Session – Colorizing Sketches

Visit the URL, https://petalica-paint.pixiv.dev/index_en.html You will be amazed to see that with Artificial Intelligence, you can draw within the tool or upload your sketches without worrying about the quality of the original file and the color them automatically in different ways.



AI Lab Session – Generate New Images

Visit the URL, <https://www.artbreeder.com/> The AI application allows users to keep selecting the most interesting image to discover totally new images. Therefore, Artbreeder turns the simple act of exploration into creativity by helping users make images by mixing any Artbreeder images together. Users can use any shared image for editing and mixing.



Credit: Joel Simon <https://www.artbreeder.com/>

AI Lab Session – Creating Comics from Video

Visit the URL <https://comixify.ai/?last=month&sortBy=trend> Upload a video and convert it into a comic. You can use this website to tell your story by making videos (less than 25 minutes) and then convert it into interesting story using comics text.

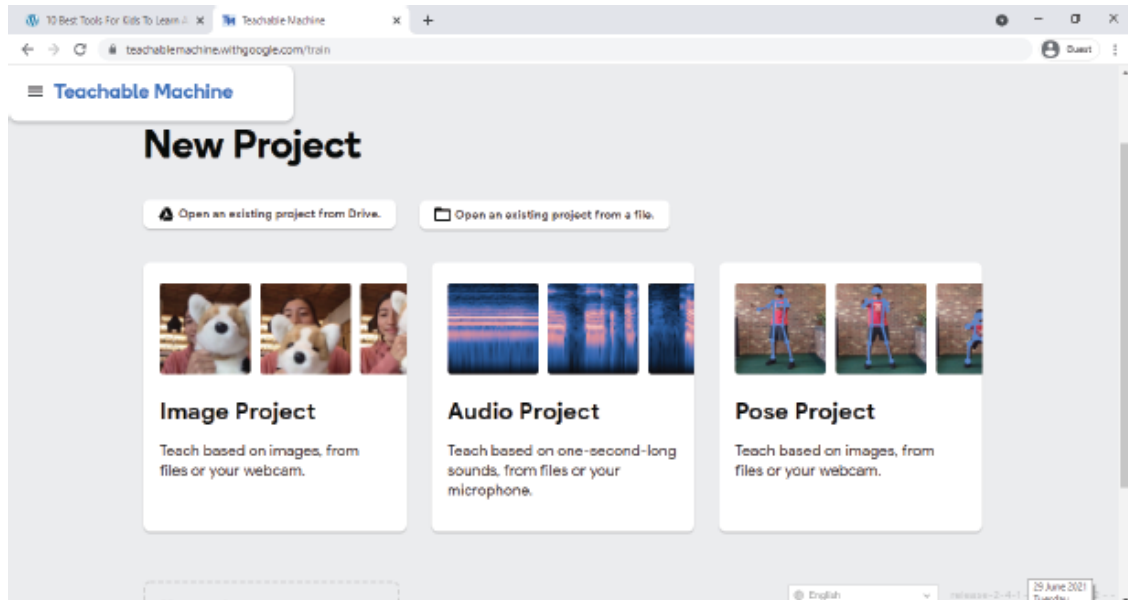
AI Lab Session – Train a Machine Learning Model

Visit the link <https://machinelearningforkids.co.uk/> and create a project that can classify text, images and numbers. You can train the model and test how accurate the results are.

A screenshot of the Machine Learning for Kids website. The browser window shows the URL 'machinelearningforkids.co.uk'. The website has a navigation bar with links: About, Projects, Worksheets, Pretrained, Book, News, Help, Log Out, and a Language dropdown. The main content area features the heading 'Teach a computer to play a game' with a blue button 'Go to your Projects'. To the right, there is a numbered list of three steps: 1. Collect examples of things you want to be able to recognise; 2. Use the examples to train a computer to be able to recognise them; 3. Make a game in Scratch that uses the computer's ability to recognise them.

AI Lab Session – Train a Machine Learning Model

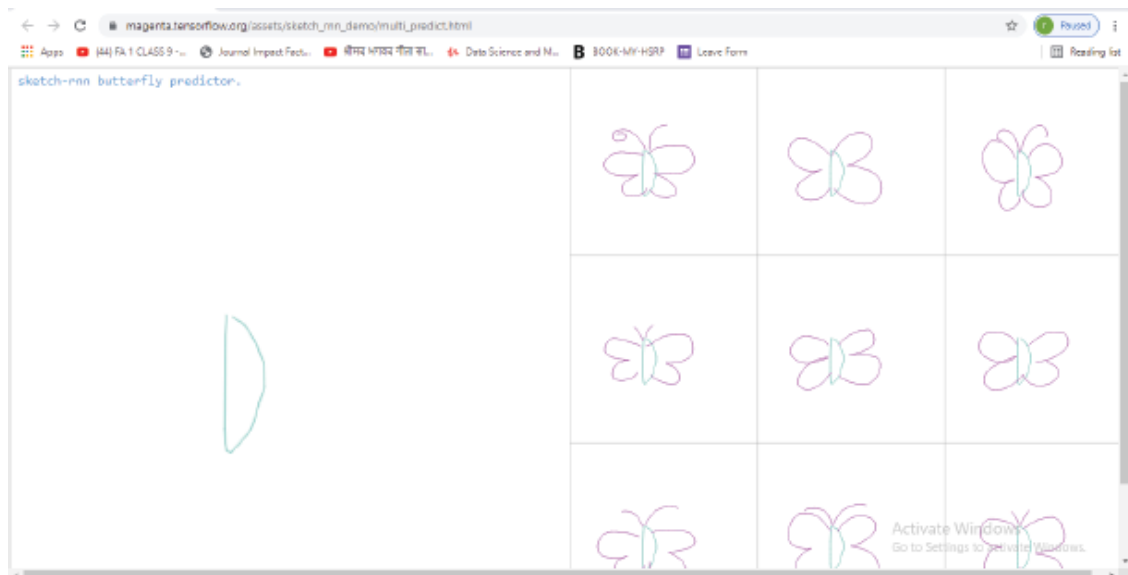
Visit the link <https://teachablemachine.withgoogle.com/> and create a project that can classify images, poses and sounds. You can train the model and test how accurate the results are.



Credit: Screenshot of TRAIN A MACHINE LEARNING MODEL, <https://teachablemachine.withgoogle.com/>, is a trademark of Google LLC

The experiment predicts the drawing which the user wants to create based on partial drawing made by the user. The user draws a sketch on the left side of the screen. On the right side, AI is used to predict the complete drawing based on user's inputs.

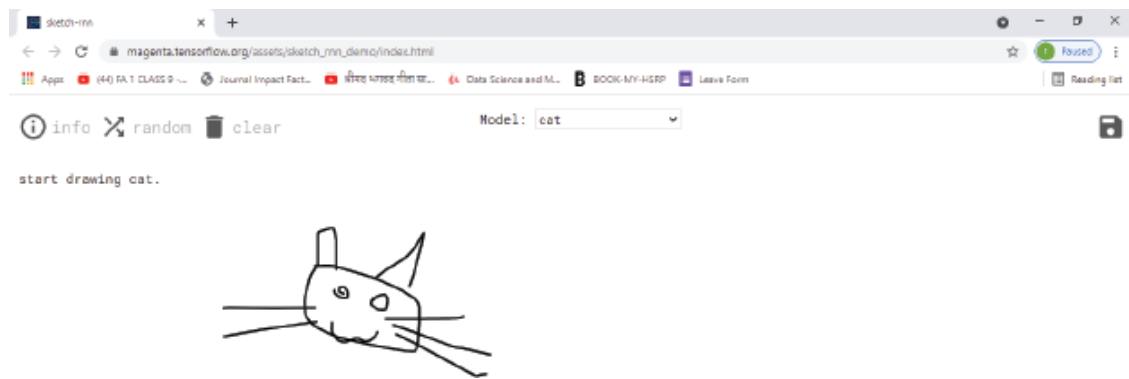
https://magenta.tensorflow.org/assets/sketch_rnn_demo/multi_predict.html



The experiment takes the drawing made by user as input and uses this input to make similar drawings. Visit the link

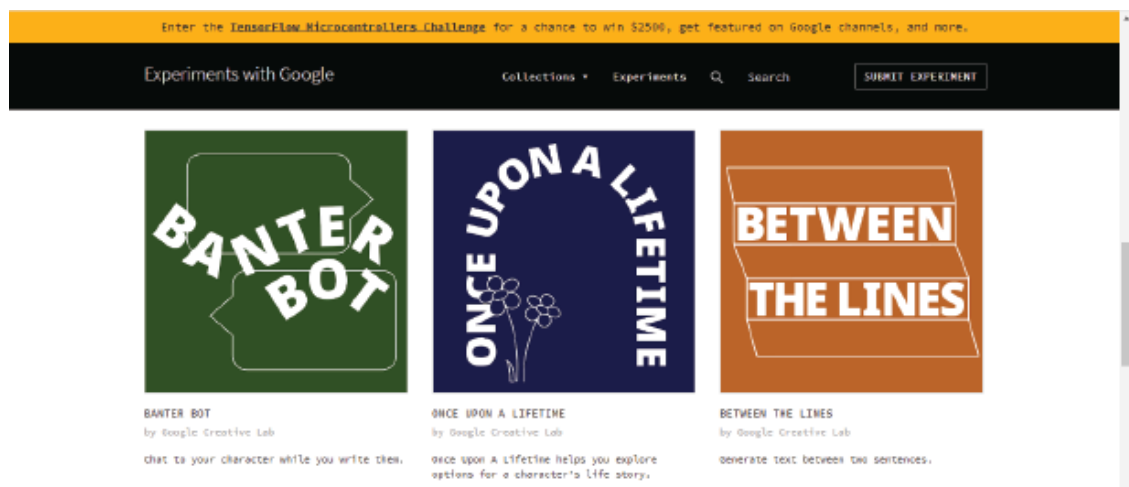
https://magenta.tensorflow.org/assets/sketch_rnn_demo/index.html to draw interesting sketches. For convenience, users can select a category of the sketch before drawing. However, do not forget to select a category

and then draw entirely different sketch. For example, select cat in the category and draw a butterfly to see how the AI system behaves.



AI Lab Session – Experiments with Google

Google uses ML techniques to support authors create interesting stories with the help of three main tools that can be accessed by clicking on the URL <https://experiments.withgoogle.com/collection/aiwriting>



Credit: Screenshot of EXPERIMENTS WITH GOOGLE, <https://experiments.withgoogle.com/collection/aiwriting>, is a trademark of Google LLC

Machine learning models detect meaningful patterns in huge quantities of complex data. ML models can learn from larger patterns of grammar and semantics and use this learning on completely different contexts. The three tools that extensively assist users in creative writing include:

Between The Lines — The program takes the first and last line of a story and then uses machine learning techniques to interpolate between the

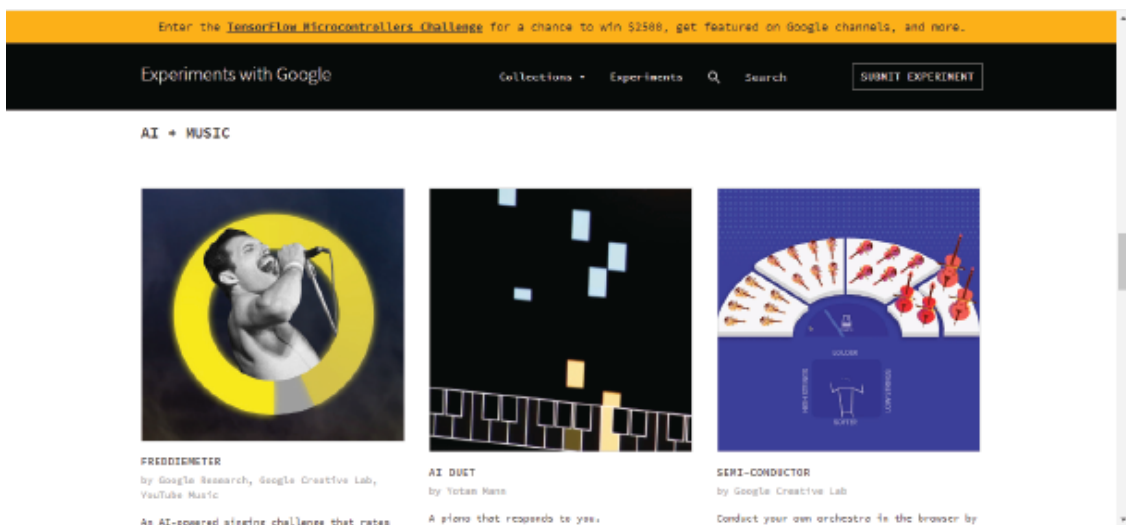
lines and generate an interesting story specifying what would happen in the middle.

Once Upon A Lifetime— This is a character life story generator in which the user input keywords describing the kind of life they want to generate. This is especially important when writing biography of a character. The program creates the entire life story using the specified keywords.

Banter Bot— Banter bot is a character chatbot. Users can input information about what a character is like, and then can converse with it through text. As the user converses, the character evolves by learning from it.

AI Lab Session – AI for Music By Google

Visit the link to know about AI applications developed by Google to support music. <https://experiments.withgoogle.com/collection/ai>



Credit: Screenshot of AI FOR MUSIC BY GOOGLE, <https://experiments.withgoogle.com/collection/aiwriting>, is a trademark of Google LLC

FreddieMeter: It is an AI-powered singing challenge that rates how closely your singing matches the voice of Freddie Mercury. This application uses machine learning models to compare user's timbre, pitch, and melody with that of Freddie's. Since user's audio is not uploaded to servers for analysis, so the vocal information stays private.

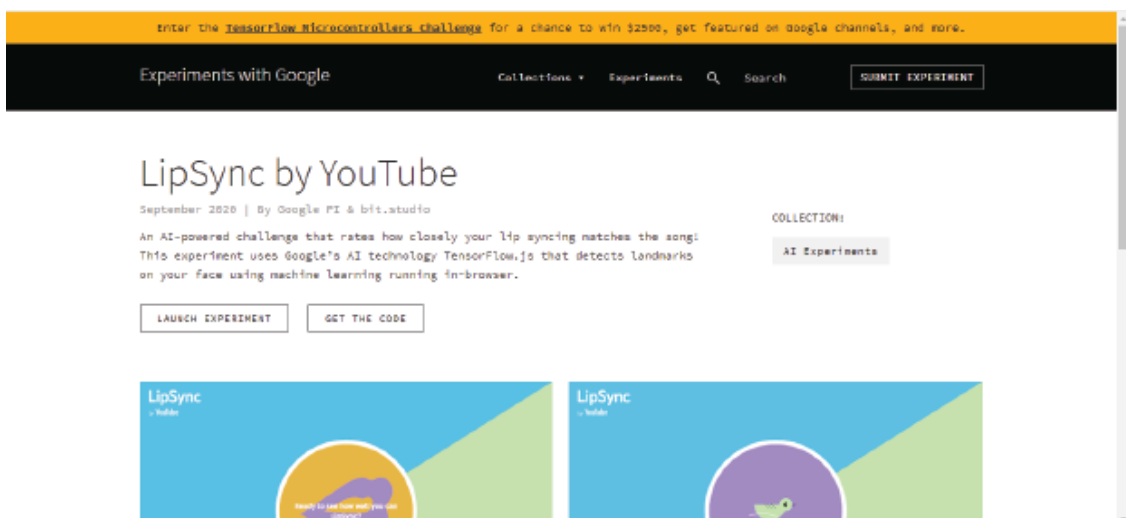
Semi-Conductor: This experiment by Google allows users to conduct their own orchestra in the browser just by moving their arms. Movement of the arms creates variation in the tempo, volume, and instrumentation of a piece of music.

AI Duet: It is another AI application by Google that plays a piano based on the inputs given by the user. Users can play some notes, and the computer will respond to the melody. Users may not even know how to play piano. They just have to press some keys and listen to what comes back.



AI Lab Session – Lip Sync

Go to the URL <https://experiments.withgoogle.com/lipsync> to launch the lip sync experiment developed by Google. The application uses AI techniques to user's lip syncing matches the song!



Credit: Screenshot of LIP SYNC, <https://experiments.withgoogle.com/collection/aiwriting>, is a trademark of Google

LLC

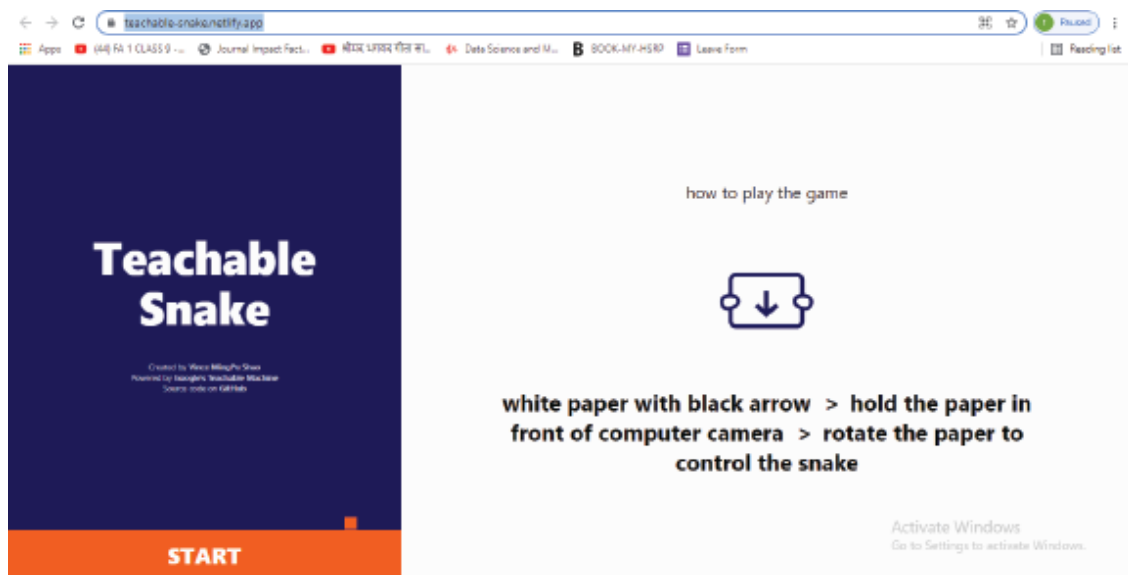
AI Lab Session – Touch Type By Google

Touch Type is an interactive type projection application that tracks simple body movements to replicate the effect on screen. Visit the link to <https://pose.yee.gd/> to use the application.

AI Lab Session – Teachable Snake

Teachable Snake is an interactive web game in which the users have to draw a black arrow on a piece of white paper as controller, and move the snake by turning the paper in different directions in front of webcam.

Visit the URL <https://teachable-snake.netlify.app/> to start the game.



Credit: Screenshot of teachable snake

AI Lab Session – Nsynth Sound Maker

The AI powered application help users create unusual new sounds using machine learning techniques. The application trained a neural network on over 300,000 instrument sounds. NSynth combine sounds, like a bass and flute, into a new, hybrid bass-flute sound. Visit the link

<https://experiments.withgoogle.com/sound-maker> to explore work with NSynth.

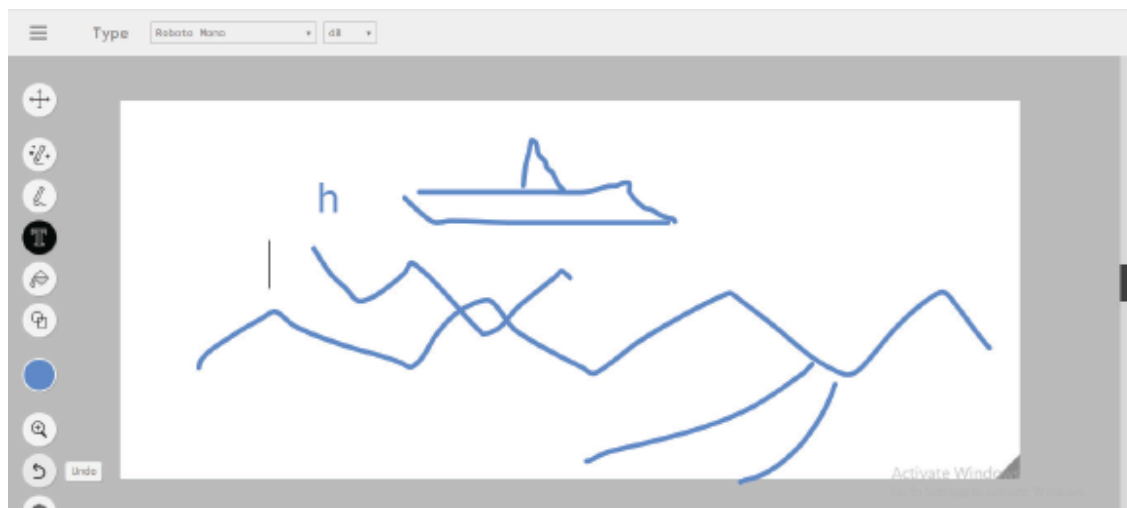


Credit: Screenshot of Nsynth SOUND MAKER, <https://experiments.withgoogle.com/sound-maker>, is a trademark of

Google LLC

AI Lab Session - Autodraw

AutoDraw is an AI powered, fast drawing application for everyone. It is a new kind of drawing tool that uses ML techniques along with drawings from talented artists to help everyone create anything visual, fast. Users can use this application on their smartphone, tablet, laptop, desktop, etc free of cost without uploading/ downloading any data from the internet. Just visit the link <https://www.autodraw.com/> to use this application.



AI Lab Session - Scroobly

Scroobly is a Google experiment that is used to create fun animations in real-time with your camera. The application uses machine learning (ML) models to map user's live motion to the screen doodle. This means that the ML system updates the animation seen on screen according to user's movement. Using Scroobly, users are extensively use artificial intelligence as a creative tool to become a digital animator. All this can be easily done

without any prior knowledge of computer coding. Visit the link <https://www.scroobly.com/> to experiment with Scroobly.

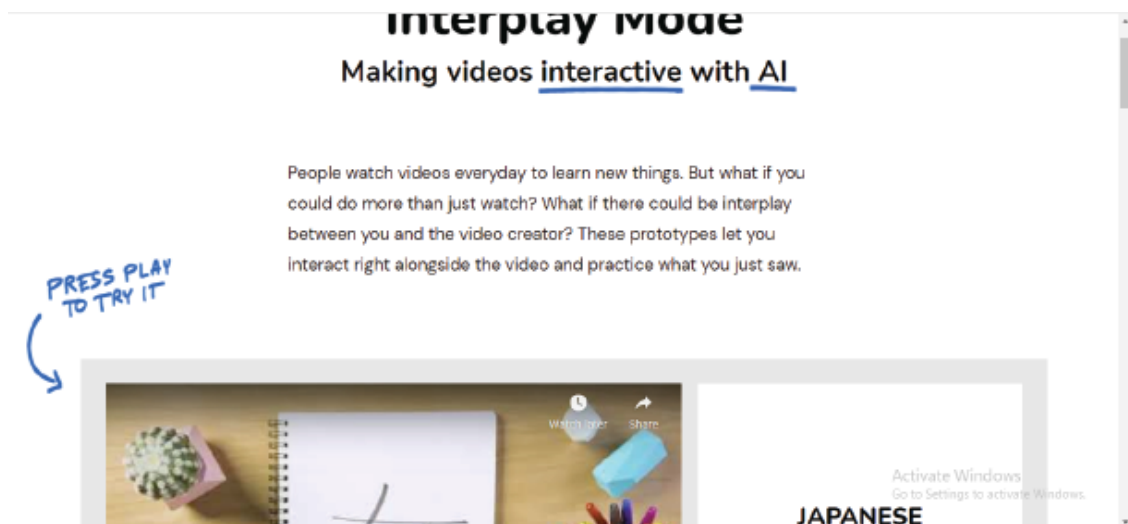


Credit: Screenshot of SCROOBLY, <https://www.scroobly.com/>, is a trademark of Google LLC

AI Lab Session – Interplay Mode

Interplay Mode is a Google experiment that allows users to make videos interactive with AI. With online education, students often learn something new from videos. But videos have a limitation of being a one-way communication mode. Therefore, the Interplay mode is extremely useful as it enables users to right alongside the video and practice what you have just learnt. So, now there is an interplay between the user and the video creator, to facilitate interaction. Visit the given link to launch the Interplay Mode application by Google.

<https://experiments.withgoogle.com/interplay-mode/view/>

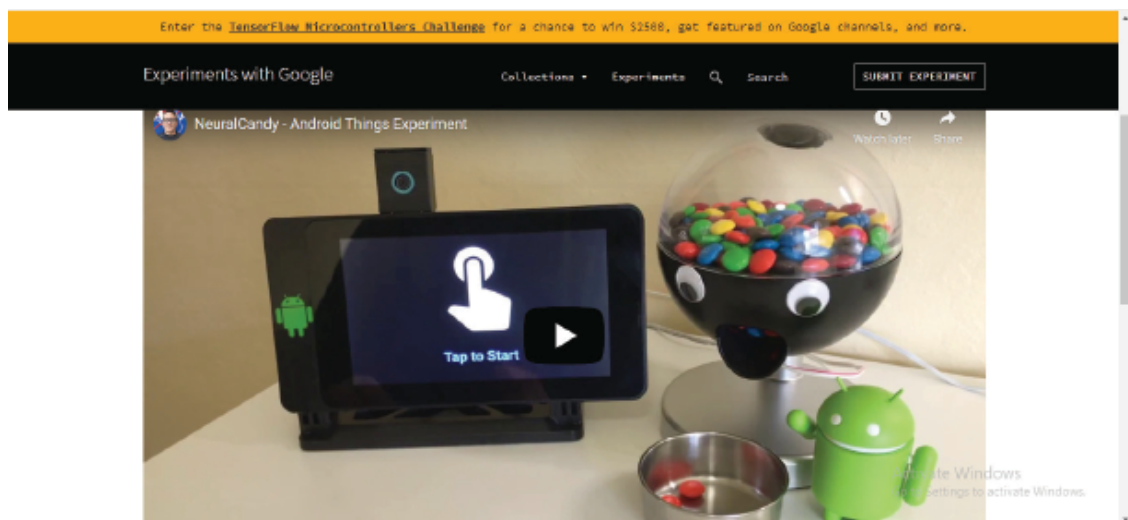


Credit: Screenshot of INTERPLAY MODE, <https://experiments.withgoogle.com/interplay-mode/view/>, is a trademark of Google LLC

AI Lab Session - Neuralcandy

NeuralCandy is an interesting experiment in which a machine dispenses candy when the user displays the requested image in front of the trained image model. The application prompts the user for a random image to be placed in front of the camera module. If the image matches the request; then the motor of the candy dispenser is activated to release the reward.

You can use a smartphone to search on Google for the requested target image and put it in front of the Pi camera. The game can be started by visiting the URL <https://experiments.withgoogle.com/neuralcandy>

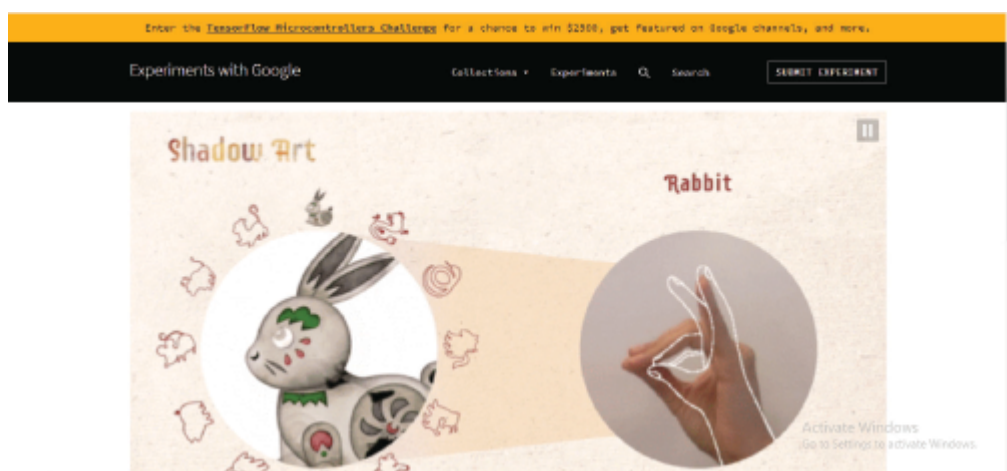


Credit: Screenshot of NEURALCANDY, <https://experiments.withgoogle.com/neuralcandy>, is a trademark of Google

LLC

AI Lab Session – Shadow Art

Shadow Art is an application by Google that allows users to use AI to play with the ancient art of shadow puppetry. To play with this game, users have to create shadow puppets with hands in front of laptop or phone camera. The puppets correspond to one of the lunar cycle's 12 zodiac symbols. The ML model identifies the animal formed by user's hands. If the user is accurately able to form the first puppet within the specified time then the next zodiac symbol has to be tried until all 12 puppets are correctly formed by the user. Visit the link to launch the Google experiment <https://experiments.withgoogle.com/shadow-art>.



Credit: Screenshot of SHADOW ART, <https://experiments.withgoogle.com/shadow-art>, is a trademark of Google LLC

AI Lab Session – Imaginary Soundscape

Imaginary Soundscape is a Web-based sound installation on Google Street view. Created using deep learning models, the application facilitates users to imagine sounds by taking a glance at a photo. For example, a beach scene is displayed with the sound of crashing waves. Similarly, when viewing a busy road, the sounds of horns and street advertising can be heard. 'Imaginary Soundscape' allows users to freely walk around Google Street View and immerse themselves into imaginary soundscapes generated using deep learning techniques. To use this application, visit the URL <http://www.imaginarysoundscape.net/>

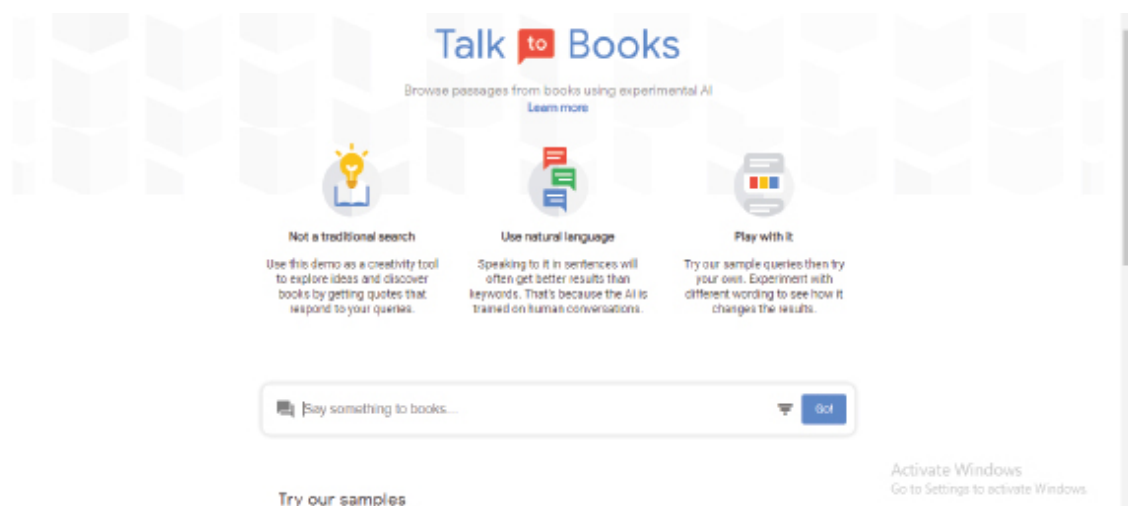


Credit: Qosmo, © 2023 Qosmo, Inc

AI Lab Session – Talk to Books

Talk to Books is an AI powered application that allows users to browse passages from books thereby providing an absolutely new way to interact with books. Users can speak directly with all 100,000 books, ask questions or make statements. Users can even filter their search results depending on category of books. Visit the link to use this application

<https://experiments.withgoogle.com/talk-to-books>



Credit: Screenshot of TALK TO BOOKS, <https://experiments.withgoogle.com/talk-to-books>, is a trademark of Google

LLC

AI Lab Session - Cartoonify

Cartoonify is an experiment by Google that use Artificial Intelligence to turn a photo into a cartoon using neural networks. For best results, upload images with clear objects on a solid background. So visit the link <https://www.kapwing.com/cartoonify> to cartoonify yourself !!!

What happens when artificial intelligence tries to draw your photo?

Cartoonify turns your photo into a cartoon drawing, powered by Draw This by Dan Macnish and the Google QuickDraw dataset. Upload any image below to try it out!

[or try a sample!](#)

Gallery

Activate Windows
Go to Settings to activate Windows.

Credit: Cartoonify, © 2023 Kapwing

AI Lab Session - Fontjoy

Fontjoy is an AI powered Google experiment that allows users to generate font combinations using deep learning techniques. Visit the link <https://fontjoy.com/> and click on Generate button to create a new font pairing. Click on the Lock icon to lock fonts that you want to keep, and click on Edit to choose a font manually. The text is editable, so users can anytime replace the default text with their own piece of text.

The application is mainly used to select fonts that share an overarching theme yet have a pleasing contrast. With this application, we are using neural networks to visualize Which fonts work together.

Classifying Images Activity

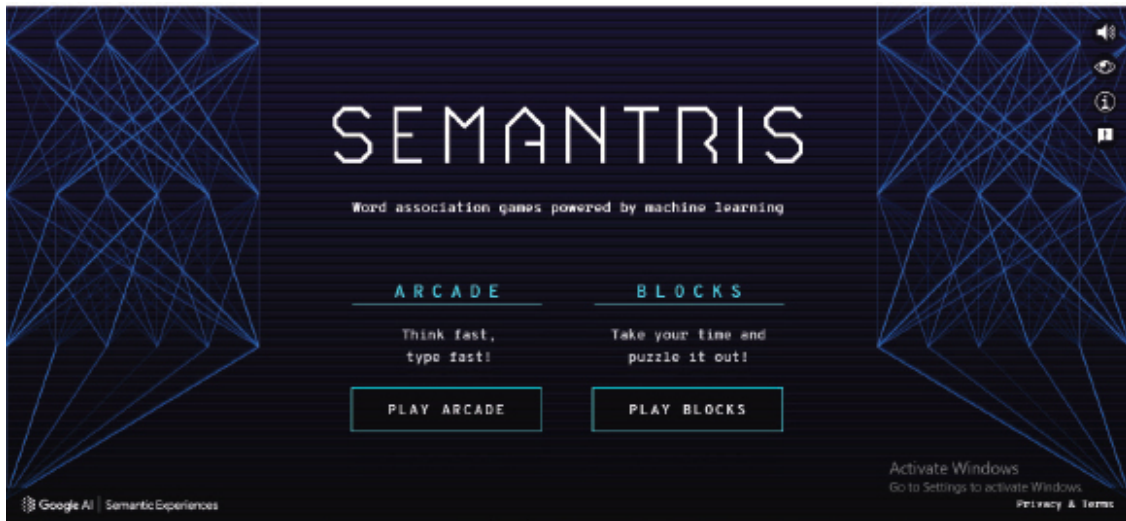
Visit the link <https://www.cs.ryerson.ca/~aharley/vis/conv/flat.html> to explore an online application of classifying different numbers. You can visualize how analysis is done using multiple convolution layers to finally classify the input digit.

AI Lab Session - Semnatris

Semnatris is a word association games powered by semantic search. The application uses machine-learning and more specifically NLP techniques. Each time the user enters a clue, the AI looks at all the words in play and chooses the ones it thinks are most related. To develop this game, the AI system was trained on billions of examples of conversational text that

span a large variety of topics. So, the application can make many types of accurate associations. Visit the link

<https://research.google.com/semantris/> to launch this experiment.

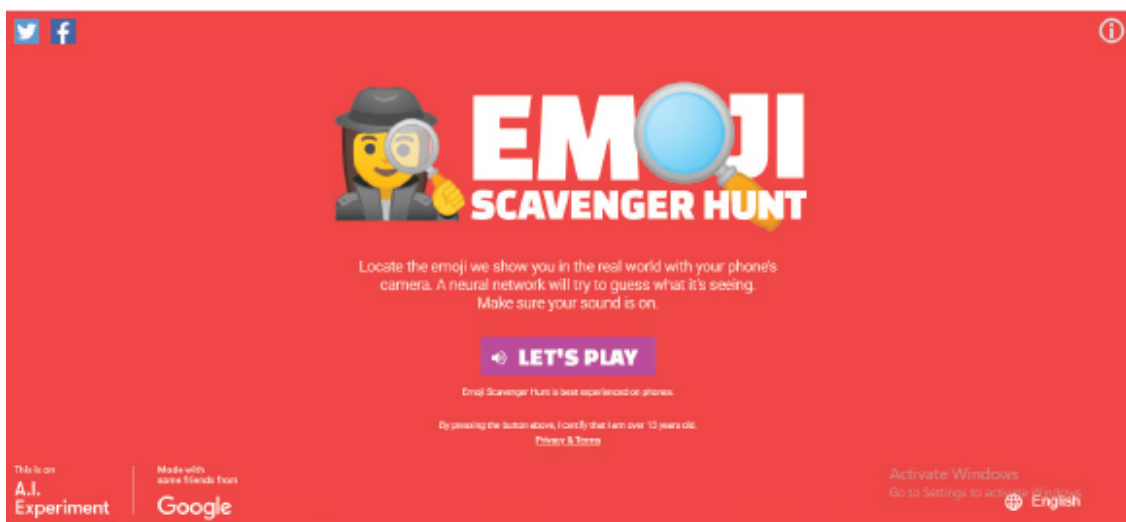


Credit: Screenshot of EMNATRIS, <https://research.google.com/semantris/>, is a trademark of Google LLC

AI Lab Session – Emoji Scavenger Hunt

Emoji Scavenger Hunt is an AI powered Google experiment that identifies emojis in the real world with user's device's camera. The application uses neural networks to identify the real-world versions of the emojis we use every day. To launch this application, visit the link

<https://emojiscavengerhunt.withgoogle.com/>



Credit: Screenshot of EMOJI SCAVENGER HUNT, <https://emojiscavengerhunt.withgoogle.com/>, is a trademark of

Google LLC



KEY TERMS

Artificial Intelligence (AI): The science and engineering of making intelligent machines, especially intelligent computer programs.

Learning processes: Acquiring data and creating rules for turning that data into actionable information.

Reactive machines: Basic machines with no memory to store and thus use past experiences to determine future actions.

Limited memory machines: Machines that retain data but only for a short period of time and can use it only for a limited time. They cannot keep adding data permanently to a library of their experiences.

Robotics: A branch of engineering that involves the conception, design, manufacture and operation of robots.

Drone: Any unpiloted aircraft that operates using a combination of technologies including computer vision, artificial intelligence, object avoidance tech and others.



RECAPITULATION POINTS

- AI systems work effectively when fed with a large amount of labelled training data.
- AI programming focuses on three cognitive skills: learning, reasoning and self-correction.
- Long Short-Term Memory (LSTM) models use past data to help predict the next item in a sequence. More recent information is given a higher priority for prediction than past data.
- Augmenting robots either enhance current human capabilities or replace the capabilities a human may have lost. Robotic prosthetic limbs or exoskeletons used to lift hefty weights are examples of such robots.

- Robotic airplanes can fly without a pilot aboard. NASA is also developing Robonaut to help people in space. The upper body of Robonaut looks like a person. It has a chest, head and arms.
- AI and machine learning can be used to identify and predict potential threats and catch instances of fraud especially those involving finances.
- No-code tools allow people to build applications and systems without having to write long and complex program codes. Applications can now be designed using visual interfaces and guided user actions.
- Low-code development platforms are based on the concepts of model-driven design, automatic code generation and visual programming.



ANSWER THE QUESTIONS

1. Define AI and give at least three advantages and three disadvantages of this technology.
2. How do AI systems work?
3. What are the three cognitive skills on which AI focuses?
4. Differentiate between the following:
 1. Weak AI and strong AI
 2. Reactive machines and limited memory AI systems
 3. Augmented and artificial intelligence
 4. No-Code and low-code AI
 5. Traditional tools and low-code AI tools
5. How will you categorize AI systems based on their functionality?
6. Explain machine learning models that apply limited memory artificial intelligence.
7. Give any ten applications of AI.
8. What are the different types of robots?
9. Give any five uses of robots.
10. How are drones useful in the field of military, agriculture and personal transportation?
11. Give some pros and cons of no-code and low-code AI tools.
12. What is the reason behind tremendous growth of no-code AI tools?



1. In _____, _____ coined the term artificial intelligence.
2. _____ AI is specifically designed to perform a specific type of task.
3. _____ focuses on imitating the human brain.
4. _____ machines interpret user's feelings by learning not only what they communicate to them but also how they communicate it.
5. Sophia and Atlas are two popular examples of _____ robots.
6. _____ allows machines to obtain information from digital images, videos, pdfs and other visual data, and take actions based on their learnings.
7. _____ is a free desktop application by Microsoft that is used for image classification with object detection.
8. _____ is a low-code AI tool that quickly creates a custom AI vision solution.



1. AI systems work well with labelled data.
2. Siri and Alexa are weak AI systems.
3. Weak AI makes full attempt to resemble human brain.
4. Autonomous vehicles use reactive machine systems.
5. Long short-term memory (LSTM) models do not consider past data while making predictions.
6. Pre-programmed robots operate in a controlled environment to perform simple, monotonous tasks.
7. Drones are being used to provide Internet access in remote areas and over irregular landforms.
8. Predicting churn rate is an application of computer vision.



MULTIPLE CHOICE QUESTIONS

1. Select a language that is not used to code AI applications.
 1. R
 2. Python
 3. Java
 4. C++
2. The CEO of which company said 'AI First'?
 1. Microsoft
 2. Google
 3. Amazon
 4. Twitter
3. Choosing the right algorithm to reach the desired outcome is a part of which process?
 1. Learning
 2. Reasoning
 3. Self-correction
 4. AI
4. In 1958, John McCarthy developed the AI programming language _____.
 1. Lisp
 2. Prolog
 3. R
 4. Python
5. Which company made the first self-driving car that could pass a state driving test?
 1. Microsoft
 2. Google
 3. Amazon
 4. Twitter
6. Google search, image recognition software, self-driving cars and IBM's Watson are some examples of which systems?
 1. Weak
 2. Strong
 3. Both of these

4. None of these
7. IBM's Deep Blue is an example of _____
 1. Limited memory
 2. Reactive machines
 3. Theory of mind
 4. Self-awareness
8. Which machine makes decisions keeping in view the feelings experienced through self-reflection and determination?
 1. Reactive
 2. Limited memory
 3. Theory of mind
 4. Self-awareness
9. _____ robots are designed to perform tasks in open environments that do not require human supervision.
 1. Pre-programmed
 2. Humanoid
 3. Augmenting
 4. Autonomous
10. Drones are examples of _____ robots.
 1. Pre-programmed
 2. Tele-operated
 3. Augmenting
 4. Autonomous



FIND OUT THREE AI TOOLS FOR

Content writing

Text paraphrasing

Note making during a meeting

Low code AI tools

No code AI tools



APPLICATION BASED QUESTIONS

1. List two real-world applications of supervised and unsupervised learning.
2. Imagine that you are supposed to train a machine to play chess. What kind of learning algorithm do you think would work better?
3. Your Instagram posts are unstructured. Justify this statement.
4. What parameters will you use to train the machine learning model to identify a cycle, auto and a car?
5. Given an image having both dogs and cats which the machine has not seen before, how will the ML model categorize them if no training data or examples were provided for prior training?
6. If you meet a stranger at a public place, how will you classify this person—gentle man/woman or a fraudster?
7. In a cricket match, mention any two parameters that you will use to classify players?
8. Can you think of any three job profiles that existed in the 80s but no longer relevant now?
9. Can you think of any three job profiles that did not exist in the 80s but are very demanding these days?
10. Can you think of any three job profiles that do not exist now but may become popular in next 10 years?
11. Think of any two problems that you would like to solve using machine learning.
12. Imagine you have been assigned the task to design a recommendation system for the school library. What data do you need to collect for training the recommendation system?
13. Fill in the boxes to predict customers who will pay the bills late.





LINKS TO EXPLORE

Go through these links to learn more about IBM Watson, Google Driverless car and Sophia, the humanoid.

https://www.youtube.com/watch?v=s_wgf75GwCM

<https://www.youtube.com/watch?v=cdgQpa1pUUE>

<https://www.youtube.com/watch?v=cdgQpa1pUUE>

Visit these links to know more about Honda Asimo and Boston Dynamics AI Robot. Make a beautiful presentation to highlight their capabilities.

(https://www.youtube.com/watch?v=1urL_X_vp7w)

<https://www.youtube.com/watch?v=NR32ULxbjYc>

Open the URL Crime in India - 2021 | Open Government Data (OGD) Platform India (<https://data.gov.in/catalog/crime-india-2021>).and answer the following questions.

1. What kind of data does it hold?
2. Why does the Government of India publish this data?
3. Why is this data available on public domain?
4. What do you think, from where did the government get this data?



ANSWERS

Fill Up

1. 1956, John McCarthy
2. Weak/Narrow
3. Theory of mind
4. Self-awareness AI

5. humanoid
6. Computer vision
7. Lobe
8. viso.ai

State True or False

1. True
2. True
3. False
4. False
5. False
6. True
7. True
8. False

Multiple Choice Questions

1. d
2. b
3. c
4. a
5. b
6. a
7. b
8. c
9. d
10. b

Previous chapter

< [About the Author](#)

Next chapter

[2. Artificial Intelligence Technologies](#) >

Table of contents expanded

2

Artificial Intelligence Technologies

2.1 Techniques in AI

Artificial intelligence (AI) works by combining massive data with fast, iterative processing and intelligent algorithms, allowing the software to automatically learn to deduce patterns in data. Building an AI system is a process of reverse-engineering human traits and capabilities in a machine. AI is a broad field of study that includes many theories, methods and technologies. To understand how artificial intelligence actually works, we must look into its various sub domains. All these domains have one thing in common. They all process large amounts of data with fast and intelligent algorithms to allow the software to learn automatically from patterns or features in the data. The sub-domains of AI are given in **Fig. 2.1**.

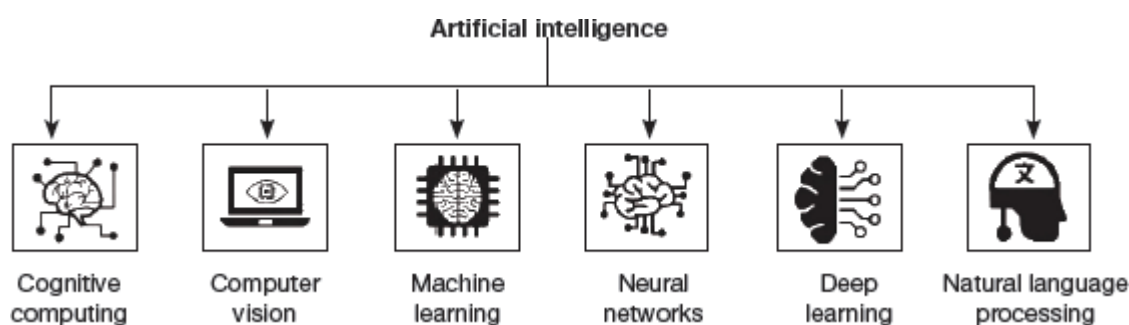


FIGURE 2.1 Sub-Domains of AI

1. **Neural Networks:** Neural networks work in the same way as human neural (brain) cells work. A series of nodes that capture the relationship between various underlying variables and processes the data. Every node (or neuron) process information by responding to external inputs and relaying information between each unit. The entire process

of learning requires multiple passes at the data to derive meaning from undefined data (refer to **Fig. 2.2**).

Note that even the most basic neural network consists of the following layers:

An *input layer*, which is the layer from where data enters the network.

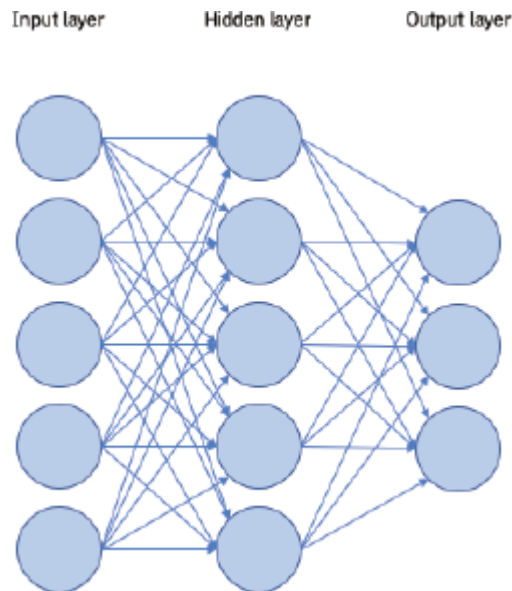


FIGURE 2.2 Basic neural network

At least one *hidden layer*, where machine learning algorithms process the inputs. Weights, biases and thresholds are applied to the received inputs and the results of processing are passed to the output layer.

An *output layer*, is the layer that gives the final result to be displayed.

2. **Machine learning:** Machine learning (ML) is a branch of computer science that analyses data and identifies patterns to teach a machine to deduce results and make decisions without any human intervention. ML algorithms learn from experiences rather than instructions. They automatically learn and improve by learning from their output. For this, humans do not have to write instructions for them to produce the desired output. They learn by analysing data sets and comparing the final output. In case of any error, they repeat the learning process until the accuracy of the outputs improve.

This automation not only saves human time and effort but also make better decisions. In **chapter 1**, we have already seen that technologies like Machine Learning, Natural Language Processing, Deep Learning are all sub domains of Artificial Intelligence (refer **Fig. 2.3**)

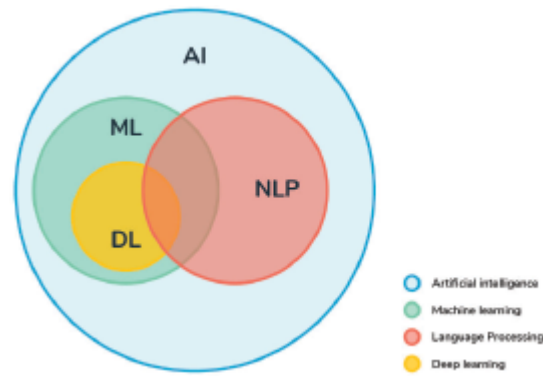


FIGURE 2.3 Relationship between AI, ML, DL and NLP.

3. **Deep learning:** Deep learning is an ML technique that teaches a machine to process inputs through layers to more accurately classify, infer and predict an outcome. DL creates huge neural networks with several layers of processing units to take full advantage of advances in computing power and improved training techniques. This helps the algorithm to learn complex patterns in large amounts of data. Some applications of DL include image and speech recognition.

Deep learning models are based on *deep neural networks* (refer **Fig. 2.4**), that is, neural networks with multiple hidden layers. In such a network, each hidden layer further processes the temporary outputs received from the previous layer. This movement of computations through the hidden layers to the output layer is known as *forward propagation*.

Once the final result is produced by the output layer, its accuracy is calculated. In case of unsatisfactory results, errors are identified, weights assigned to each node are updated, and pushed back to the previous layers to refine or train the model. This process of moving backward to update weights of all nodes is known as ***backward propagation***.

Deep learning models can work with labeled as well as unlabelled data. This means that deep learning supports both supervised and *unsupervised learning*.

4. **Natural language processing:** NLP is a science in which a machine is made to read, understand, interpret and respond to a human language. This is specifically done to make machine capable of communi-

cating with a human.

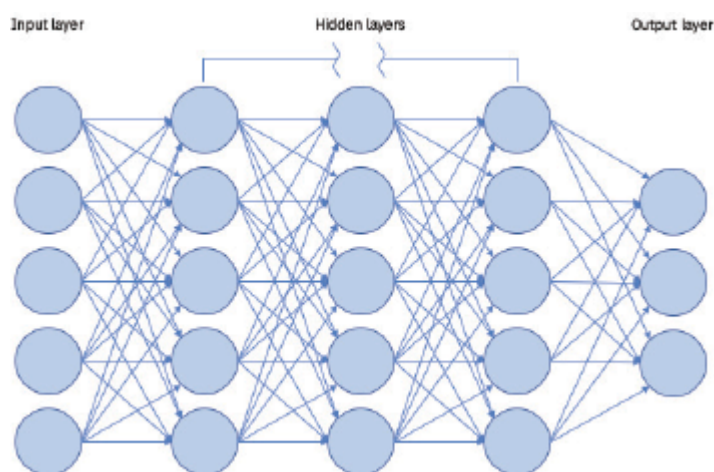


FIGURE 2.4 Deep neural network

5. **Computer vision:** Computer vision is a branch of AI that tries to understand an image by breaking it down into several parts and then studying each part of the image. This helps machine to classify and learn from a set of images so that it can make better decisions based on previous observations. For example, when a machine can process, analyse and understand images, they can capture images or videos in real time and interpret their surroundings.

Computer vision techniques are used today for facial recognition that helps in surveillance and security systems, autonomous vehicles, retail stores for tracking inventory and customers, in medicine for diagnosing diseases, in financial Institutions to prevent fraud, and so on.

6. **Cognitive Computing:** Cognitive computing algorithms try to mimic human brain by analysing text, speech, images or objects in the same way to give the desired output. It is basically a subfield of AI that is used to provide a natural, human-like interaction with machines. The ultimate goal of using cognitive computing is to make the machine speak coherently in response to a human.

Apart from the above-mentioned techniques, some additional technologies that enable and support AI include the following:

1. **Graphical processing units** that provide heavy computing power required for iterative processing and training neural networks.
2. **Internet of Things** to generate massive amounts of data from connected devices. Usually, this data remains unanalysed. Automating models with AI helps to analyse this data and extract useful informa-

tion from it. Advanced algorithms **can be used** to analyse data faster at multiple levels to identify and predict rare events, understanding complex systems and optimizing unique scenarios.

2.2 Machine Learning Model

We can better understand the role of machine learning techniques through a very simple definition given by professor Mitchell –

‘A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.’ Based on this definition, the Machine Learning Model can be given as shown in **Fig. 2.5**. Here,

Task(T), is the real-world problem to be solved. For example, predicting sales of a product, classifying an email as spam or not a spam, etc. Technically, examples of ML based tasks are classification, regression, clustering, recognition, etc.

Experience (E) is the knowledge gained from data provided to the algorithm or model. Once data is provided, the model runs iteratively to learn some inherent patterns. Therefore, like humans, machines now learn from experience by analysing situation, relationships etc. Supervised, unsupervised and reinforcement learning are some ways to learn or gain experience. Experience acquired through ML model or algorithm is used to solve task T.

Performance (P) is a measure that indicates how well a particular ML algorithm has performed the given task T using experience E. Performance is analysed based on well-defined metrics including accuracy, F1, confusion matrix, precision, recall, sensitivity, etc.

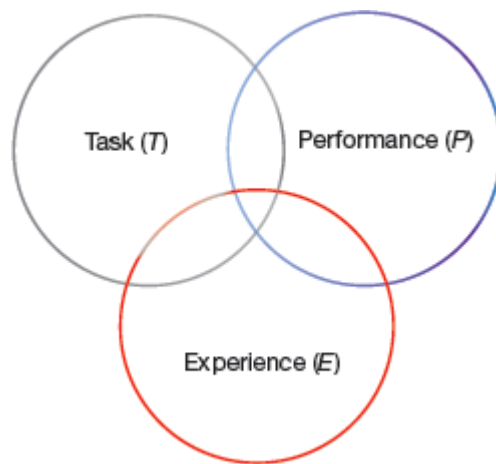


FIGURE 2.5 ML Model is a combination of Task, Performance and Experience

2.2.1 Types of Machine Learning Algorithms

Machine learning algorithms can be categorized as supervised or unsupervised.

2.2.1.1 Supervised Machine Learning Algorithms

As the name indicates, such algorithms have a supervisor as teacher. These algorithms apply learning from past data (or experiences) to new data using labelled examples.

Technically, supervised learning algorithms learn an association between input data and output. For example, if we have input variable(s) X and output variable (Y), then the mapping function from input to output can be given as, $Y=f(x)$. This mapping function can be used to predict the output value for any new input after learning from the existing data.

For example, if you have a basket filled with different varieties of fruits then the first step is to train the machine to identify a fruit. Now, our machine can easily identify an apple and a banana. Supervised learning algorithms can be further classified into two categories:

Supervised learning should be used when output of data in the training set is known.

Classification Algorithm

A classification algorithm classifies data into a particular group. Classification techniques predict discrete categories. The output will be based on what the model has learned in training phase.

For example, a fruit as either an apple or a banana. In real world applications, classification can be used in medical imaging, speech recognition, hand-writing recognition, credit scoring, predict if an incoming email is authentic or spam, or whether a tumour is cancerous or benign.

Classification algorithms are best used if data can be tagged, categorized, or separated into specific groups or classes.

Regression

A regression algorithm predicts a real value. The output value is based on what the model has learned in its training phase. In contrast to classification algorithms, regression predict continuous values. For example, the cost of a product, the value of a stock, changes in temperature or fluctuations in power demand.

Thus, in supervised machine learning algorithm, data input and desired output, along with a feedback about the accuracy of predictions during algorithm training are provided. Data scientists can select variables or features that can be used by the model to analyse data and make predictions. For example, supervised learning can be used by a company to identify customers who are likely to churn. It can also be used by insurance companies to predict the likelihood of occurrence of a mishap and determine the total insurance value.

In supervised learning, clear instructions are given specifying what needs to be learnt and how it needs to be learnt.

2.2.1.2 Unsupervised Learning

Unsupervised learning trains the machine using information that is neither classified nor labelled. In this case, the machine learning algorithm works on that information without any guidance. The unsorted information is grouped based on similarities, patterns and differences without any prior training of data. For example, if we give an image of mango and an orange, then initially, the machine has no idea about how a mango looks and how the orange looks.

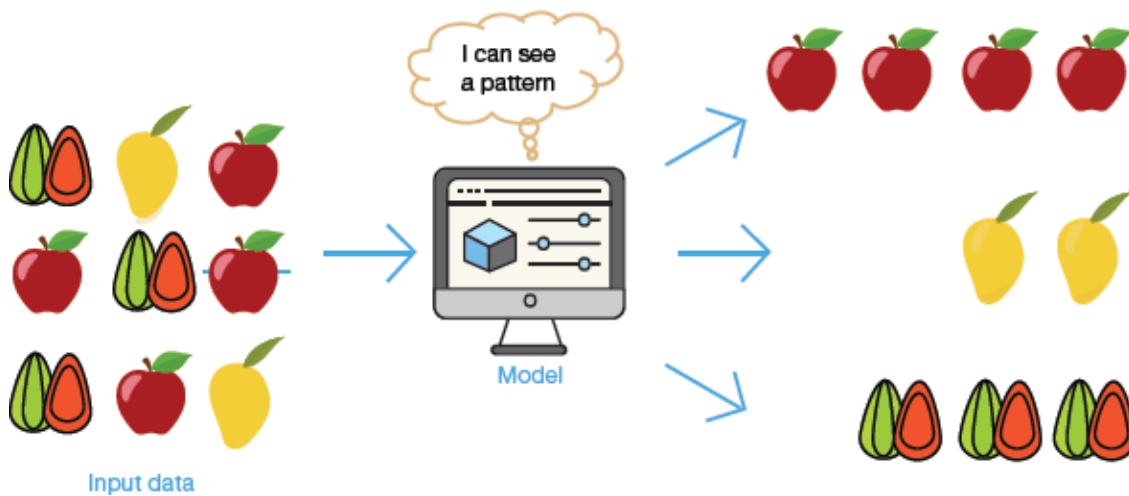


FIGURE 2.6 Grouping fruits by identifying patterns

Unsupervised machine learning algorithm learns through observation and finding structures in the data. That is, the model automatically finds patterns and relationships in the dataset by creating clusters in it. For example, if given a dataset of pictures of both mangoes and oranges, the algorithm can make two clusters- one containing only pictures of oranges and the other of mangoes (refer **Fig. 2.6**). What an unsupervised machine learning cannot do is specifying labels to the clusters. That is, it can only segregate the pictures but cannot tell that this is a real-world orange and that is a mango.

Unsupervised learning can be categorized into following sets of algorithms:

1. **Clustering:** In clustering, the aim is to discover inherent groupings in the data or discover hidden patterns (refer **Fig. 2.7**). It is one of the most useful unsupervised machine learning techniques as it finds sim-

ilarity as well as relationship in the underlying data. For example, a company may like to group its customers by their purchasing behaviour. A cell phone company can use clustering to optimally decide the locations where they can build cell phone towers (these clusters can depict the number of people relying on their towers). Other applications include gene sequence analysis, market research, and object recognition.

2. **Association analysis:** In association mining (or analysis), the aim is to discover rules that describe large portions of data. For example, a company can use association analysis to conclude that customer who buys X also buys product Y.
3. **Dimensionality reduction:** It is used to reduce the number of feature variables for the data set. It is done by selecting a set of principal or representative features. Dimensionality reduction is very important technique especially when the data set has a large number (millions).
4. **Outlier detection or anomaly detection:** This technique is used to find out the occurrences of rare events or observations that generally do not occur. Application of learned knowledge in anomaly detection techniques helps to differentiate between anomalous or a normal data point. Generally, clustering (more specifically, KNN) is used to detect anomalies based in data.

Thus, unsupervised machine learning algorithms (also called neural networks) are used when the information used to train is neither classified nor labelled. These algorithms use an iterative approach called deep learning to review data and arrive at conclusions. Unsupervised learning algorithms are used for more complex tasks than supervised learning systems. For example, these algorithms are used in image recognition, speech-to-text and natural language processing applications, predict the probability of presence of a particular disease. A retailer can use unsupervised learning technique to find out products that are frequently bought by customers tends to buy more frequently.

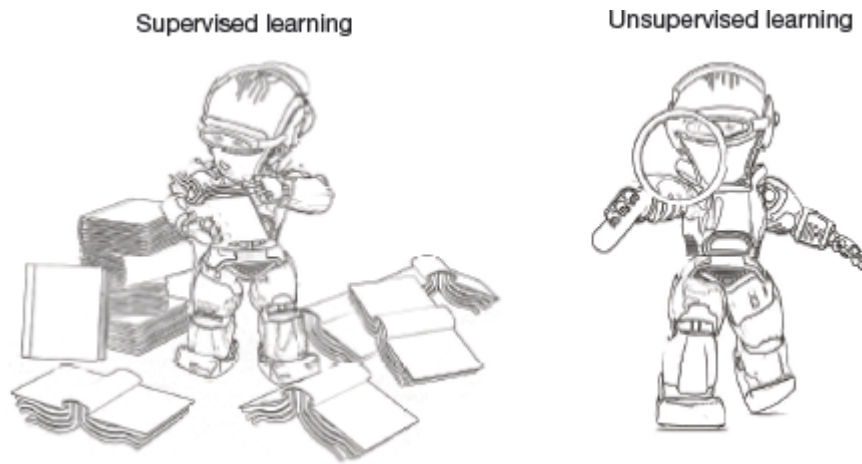


FIGURE 2.7 In supervised learning, we have some clue about what exactly we are finding but unsupervised learning explores data to find just any hidden pattern

2.2.1.3 Semi-Supervised Learning

Semi-supervised machine learning algorithms fall somewhere between supervised and unsupervised algorithms since they use both labelled and unlabelled data for training. These algorithms improve accuracy.

Semi-supervised learning algorithms use small amount of pre-labelled data and a large number of unlabelled data for training. Semi-supervised learning techniques can be applied using any of the two approaches given below.

First, to build the supervised model based on small amount of labelled data followed by building an unsupervised model by applying the same to the large amounts of unlabeled data. Experience gained by generating more labelled data is used to train the model. This process is repeated multiple times.

In the second approach, unsupervised methods are used to cluster similar data samples, annotate these groups and then use this information to train the model.

Application of the discussed techniques can be summarized in the flow-chart given in **Fig. 2.8** below.

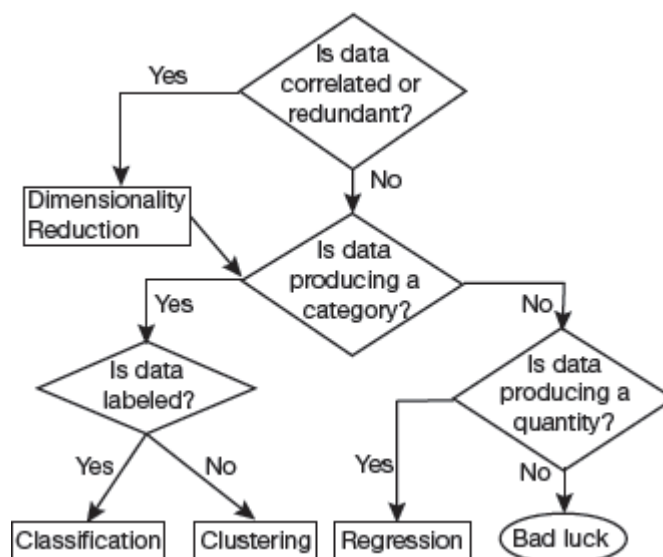


FIGURE 2.8 Summary of ML Algorithms

2.2.1.4 Reinforcement Learning (RL)

These techniques are different from the previously discussed techniques and are rarely used. In a reinforcement learning algorithm, an agent is trained over a period of time so that it can interact with a specific environment.

Reinforcement learning, is a type of dynamic programming that trains algorithms using a system of **reward and punishment**. The agent receives rewards by performing correctly and penalties for performing incorrectly. In this way, the agent learns without any human intervention to maximize its reward and minimize its penalty.

Since RL requires a lot of data, it is mostly used in areas where simulated data is readily available like gameplay and robotics. In these areas, RL is used to find the best possible behaviour or path that can be taken in a particular situation.

Reinforcement learning is different from supervised learning. In supervised learning, the training data has labels, so the model is trained with the correct answer but in case of RL, the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its own experience.

Example: In the game shown in **Fig. 2.9**, there is an agent (robot) and a reward (diamond), with many hurdles (fire) in between. The robot has to learn by trying all the possible paths and then choose the path that gives him the reward with the least hurdles. Each right step earns a reward and every wrong step will subtract the reward of the robot. The total reward is calculated when it reaches the final reward that is the diamond.

To summarize technically, in an RL algorithm, note the following:

1. Input is an initial state from which the model will start
2. Output is a list of possible outputs for a particular problem
3. Training is based on the input. The model returns a state and the user will decide to reward or punish the model based on its output. The model keeps learning this way until it finds the best solution (or the solution with maximum reward).

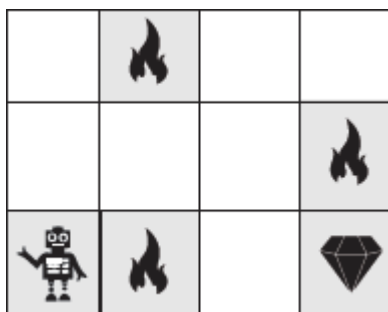


FIGURE 2.9 Robot and the Reward Game

2.2.1.4.1 Types of Reinforcement

There are two types of reinforcement as follows:

Positive reinforcement has a positive effect on behaviour. It occurs when a particular behaviour, increases the strength and the frequency of the behaviour. Such a reinforcement maximizes performance and sustain changes for a long period of time.

Negative reinforcement is defined as strengthening a behaviour by stopping or avoiding a negative condition.

2.2.1.4.2 Applications of Reinforcement Learning

RL is used in large environments when a model of the environment is known, but an analytic solution is not available.

1. RL can be used in robotics for industrial automation.
2. It is used to make machines learn
3. RL is used in data processing applications
4. RL can be used to create training systems that provide custom instruction and materials according to the requirement of students.
5. RL is used for creating game playing software (playing chess).
6. RL is used in a self-driving car where the car (agent) interacts with its environment, receives a reward depending on how it performs, such as driving to destination safely. Similarly, the agent receives a penalty for performing incorrectly, such as going off the road or hitting a hurdle.

An RL agent perceives and interprets its environment, takes actions and learns through trial and error.

2.3 Regression Analysis in Machine Learning

Regression analysis is a statistical method tool that allow users to study the relationship between a dependent (target) and one or more independent (predictor) variables as shown in **Fig. 2.10**. This helps the analyst to understand how the value of the dependent variable changes with respect to independent variables. The values to be predicted are usually continuous or real (like **temperature, age, salary, price**, etc.). For example, a company can analyse its data to find out the extent to which expenditure on advertisements increases sales of a particular product. Similarly, we can use regression analysis to predict rainfall using temperature and other factors; determine market trends; predict road accidents due to rash driving.

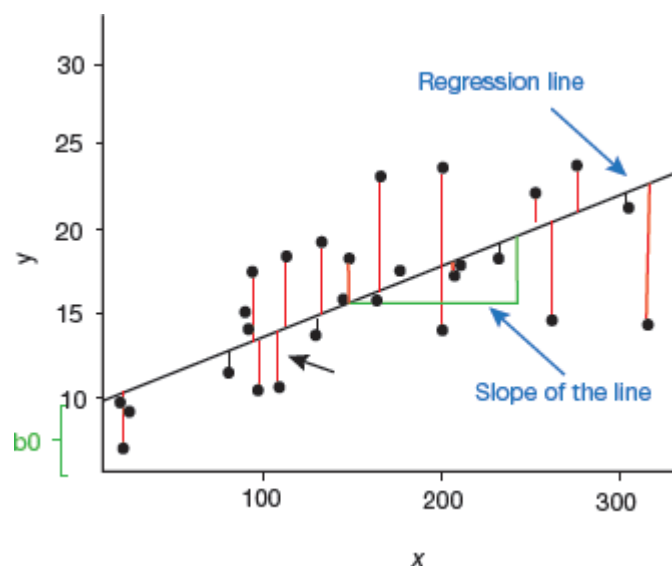


FIGURE 2.10 Plotting the Regression Line

Regression analysis is widely used for **prediction, forecasting, modeling time series data, and inferring the causal-effect relationship between variables.**

In regression analysis, a graph that best fits the given datapoints is plotted. This graph can either have a straight line or a curve that *passes through all the datapoints on the graph that is drawn depicting the dependent and independent variable(s). The vertical distance between the datapoints and the regression line should be minimum as this line indicates whether the model has captured a strong relationship or not.*

2.3.1 How Regression Analysis Works?

Regression analysis creates a mathematical equation that defines y as a function of the x variables. This equation is then used to predict the value of y on the basis of values of the predictor variables (x).

Linear regression is the most basic, simple and widely used technique for predicting values of a continuous variable. It assumes that there exists a linear relationship between the outcome and the predictor variables as shown in **Fig. 2.11**.

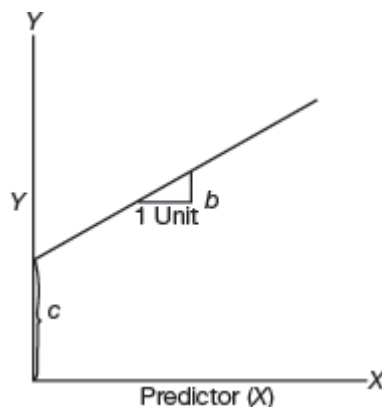


FIGURE 2.11 Relationship between outcome and predictor variable

The linear regression equation can be given as,

$$y = b_0 + b \cdot x + e, \text{ where:}$$

b_0 is the intercept,

b is the coefficient of x .

e is the residual error

Values of coefficients are determined in such a way that the residual error is minimized. This method of computing the beta coefficients is known as **ordinary least squares** (OLS) method.

In case of multiple predictor variables, say x_1 and x_2 , the regression equation can be written as $y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + e$. There may also exist an **interaction effect** between two or more predictor variables. For example, increasing the value of one predictor variable may in turn increase the effectiveness of the other predictor(s) in explaining the variation in the outcome variable.

When there are multiple predictors in the regression model, then the best combination of predictors needs to be chosen (like **best subsets regression** and **stepwise regression**) to construct the optimal predictive model. In such cases, we use model selection technique that compares multiple models built with different sets of predictors to select the best performing model that minimizes the prediction error.

Linear regression models work well with both continuous and categorical predictor variables. However, before applying linear regression model on our data set, we must first make sure that the linear model is suitable for the underlying data. It may happen that the relationship between the outcome and the predictor variables is not linear. In such cases, a **non-linear regression** model (like *polynomial and spline regression*) is preferred over a linear model.

When there is a large multivariate data set containing some correlated predictors, then such variables are summarized into few new variables that are a linear combination of the original variables.

These techniques based on **principal components** include **principal component regression** and **partial least squares regression**.

Alternatively, we can use penalized regression to simplify a large multivariate model. Such a regression model penalizes the model for having too many variables. **Ridge regression** and **Lasso regression** are popular examples of penalized regression model.

Thus, different regression models can be applied on the data and their performance can be compared to select the best one that explains the data. Statistical metrics are used to compare the performance of the different models in explaining the data and predicting the outcome of new test data.

Case Study 1: Let us compute auto fare. We know that auto charges are computed by adding a fixed amount and a variable cost. The fixed amount is set by the government. The variable cost depends on the distance travelled. It is usually specified as Rs 11 per km. If the fixed charge of an auto is Rs 30, then a linear regression equation can be used to find the cost of any auto trip. By using “x” to represent the number of kilometers travelled, “y”, the cost of that auto ride can be calculated as, $y = 11x + 30$.

Now, if I say that I took an auto to reach a destination that was 10 km away from my house, how much would I have to pay? Yes, Rs. 140. Because, $y \text{ (cost)} = 11 \times 10 + 30$.

Case Study 2: A company decides to carry out its business operations on a rented space. If the cost of the rental space is Rs 20000 plus Rs 500 per employee per day, then compute monthly rental for space given that the company is open 5 days a week.

The linear equation in this scenario, can be given as,

$$y = (500)(5)(4)x + 20000$$

$$y = 10000x + 20000$$

If 20 employees are to be present every day, then monthly rental would be

$$Y (\text{cost}) = 10000 \times 20 + 20000 = 220000.$$

Case Study 3: Let us make predictions using simple linear regression equation. If the annual expenditure of a bakery shop is Rs 500000 and the monthly sales is Rs 450000, then the linear equation to compute profit for x months can be given as,

$$y = 450000x - 500000$$

For example, after six months, the shop can expect a profit of,

$$450000 \times 6 - 500000 = 2200000.$$

2.3.2 Model Evaluation Metrics

The best regression model is the one with the lowest prediction error. The most widely used metrics for comparing regression models are discussed here:

Root mean squared error measures the model prediction error. It is calculated as the average difference between the observed known values of the outcome and the predicted value by the model. Mathematically,

$$\text{RMSE} = \sqrt{\text{mean}((\text{observeds} - \text{predicted})^2)}$$

Lower the RMSE, better is the model.

Adjusted R-square represents the proportion of variation in the data thereby reflecting the overall quality of the model. Higher the adjusted R^2 , better is the model

These metrics are computed on a new test data that has not been used to train or build the model. In a large data set with many records, rows can be randomly split in 80:20 ratio, where 80% of the rows are used to build the predictive model and rest of the 20% are used as test set or validation set for evaluating the model performance.

One of the best techniques for estimating a model's performance is **k-fold cross-validation**. This can be applied even on a small data set. Step followed to perform k-fold cross-validation are as follows:

Step 1: Randomly split the data set into k-subsets (or k-fold). For example, to generate 5 subsets, value of $k = 5$.

Step 2: Reserve one subset and call it as test data. Use rest of subsets to train the model.

Step 3: Test the performance of the model using the test data set and record the prediction error

Step 4: Repeat the above steps until each of the k subsets have been used as the test set.

Step 5: Calculate the average of the k recorded errors. This is also known as cross-validation error. Finally, the best model is the one that has the lowest cross-validation error, RMSE.

2.3.3 Types of Regression

The type of regression analysis that can be applied on a particular data set depends on the attributes, target variables, shape and nature of the regression curve that represent the relationship between dependent and independent variables. Some important regression techniques are as follows:

1. **Linear regression:** It is the simplest regression technique used for predicting values for a dependent variable that has linear relationship between the response and predictors (or descriptive variables). The general equation of linear regression can be given as,

$$Y = bX + C$$

where Y is a dependent variable

X is the independent variable

b is the slope of the regression line and

C is the intercept.

Though linear regression suffers from overfitting issues, it is still used as it is fast and easy to model and evaluate. It is best used when the target relationship is not complex or enough data is not available.

Linear regression (**Fig. 2.12**) is also very useful for detecting outliers.

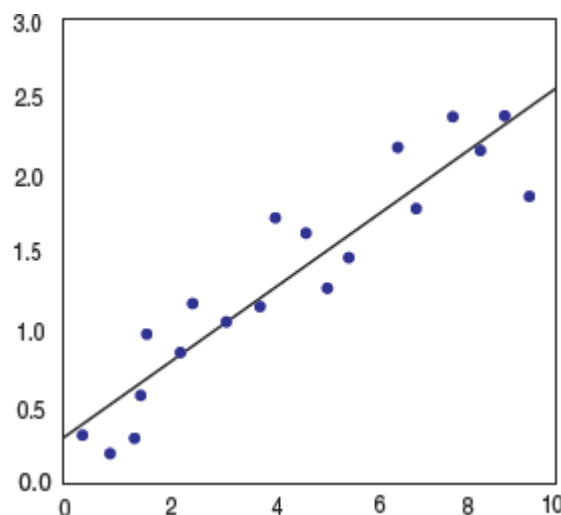


FIGURE 2.12 Linear Regression Graph

2. **Logistic regression:** It is preferred when the dependent variable is binary (dichotomous) in nature (like 0 or 1, true or false, yes or no). That is, logistic regression is preferred when there is a need to ascertain the probability of an event in terms of either success or failure. In such a case, the relationship between the dependent and independent variables are calculated by computing probabilities using the logit function (refer **Fig. 2.13**). Logistic regression is mostly used to analyse cate-

gorical data.

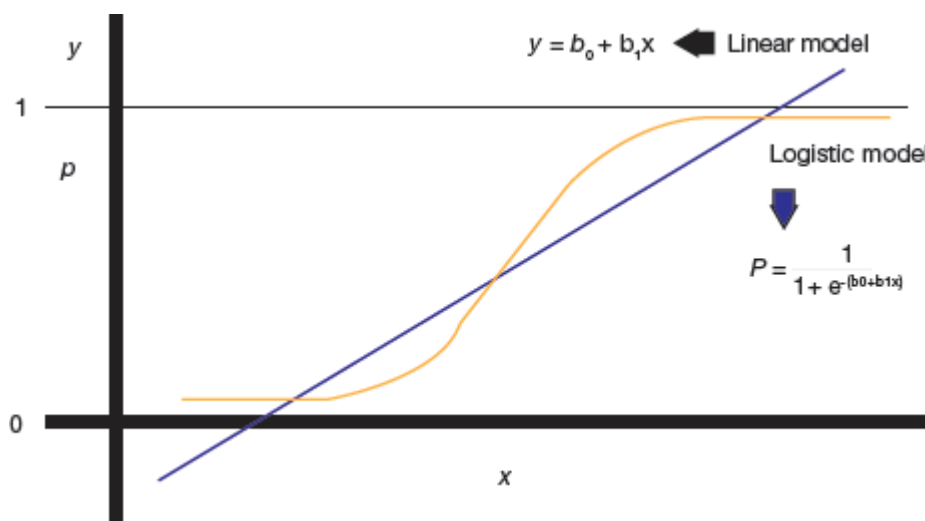


FIGURE 2.13 Logistic Regression Curve

3. **Ridge regression:** Ridge regression is basically used for analysing numerous regression data. In case of multicollinearity, least-square calculations get unbiased, so the regression model becomes too complex and approaches to overfit. In such a case, we need to minimize the variance in the model and save it from overfitting. Ridge regression helps us to do this by correcting the size of the coefficients. A bias degree is affixed to the regression calculations that in turn reduces standard errors.
4. **Lasso (Least Absolute Shrinkage Selector Operator) regression:** It is a widely used regression analysis technique for variable selection and regularization. Lasso regression uses thresholds to select a subset of the covariates given for the implementation of the final model. It reduces the number of dependent variables if the penalty term is huge. This is done by reducing the coefficients to zero so that features can be selected easily. Lasso regression is also known as L1 regularization.

Ridge regression eases collinearity in between predictors of a model

5. **Polynomial regression:** Polynomial regression is used to construct a model that fits non-linearly separated data. In such a case, the best-fitted line is not a straight line, but a curve.

The equation of polynomial regression is given as,

$$Y = b_0 + b_1x + b_2x^2 + \dots + b_n x^n$$

Polynomial regression is widely used to analyse curvilinear form of data (refer **Fig. 2.14**) and best fitted for least-squares methods.

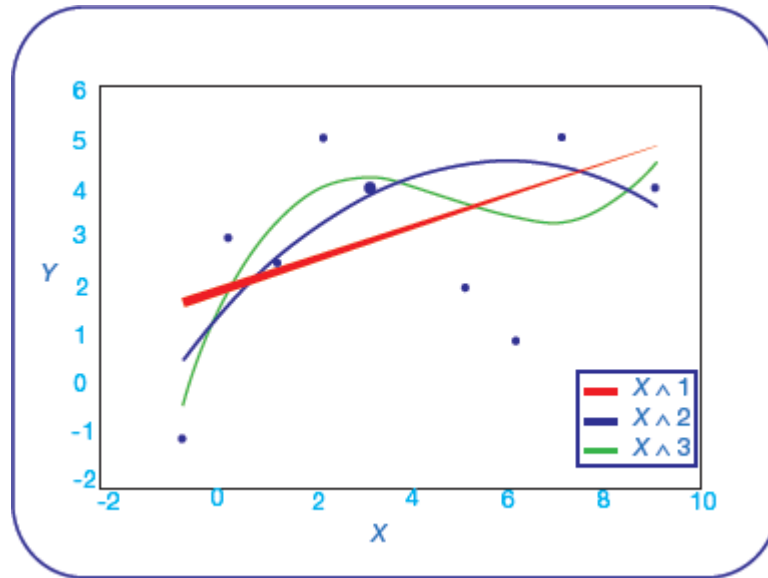


FIGURE 2.14 Polynomial Regression Curve

6. **Stepwise regression:** It is used to build predictive regression models that are carried out naturally. With every forward step, a variable gets added or subtracted from a group of descriptive variables.

We can use **forward selection**, **backward elimination** or **bidirectional elimination** for adding/removing variables at each step. As the name suggests, forward selection continuously adds variables to the set and reviews the performance of the model. It stops when no further improvement is needed (or being achieved). Backward elimination removes variables until no extra variable is left to be deleted without considerable loss. Bidirectional elimination is a good combination of both the approaches.

7. **ElasticNet regression:** Elastic regression is a mix of ridge and lasso regression. It produces a grouping effect when highly correlated predictors are either in or out in the model combinedly. This technique is preferred when there are too many predictors as compared to the number of observations. ElasticNet regression is usually used in SVM (Support Vector Machine Algorithm), metric training, and document optimizations.

2.4 Classification Techniques

In this section, we will read about some widely used classification algorithms. These algorithms predict the probability that the data that follows

will fall into one of the predetermined categories.

2.4.1 K-Nearest Algorithm

The k-nearest neighbour algorithm is a supervised learning algorithm in which the output value of data is known but how to get the output is not known. For example, if we have several groups of labeled samples and all items available are homogeneous, then to find which group an item with unknown label belongs, we will find similarities between the item (at hand) and with items in each group. Finally, we will conclude that the item belongs to the group to which it is most similar. The k-nn algorithm works in exactly the same way.

The k nearest neighbours' algorithm stores all available cases and classifies new cases by a majority vote of its k neighbours. The algorithm segregates unlabelled data values into well-defined groups.

Technically speaking, ***k-nn is a non-parametric supervised learning algorithm*** which classifies data into a particular category with the help of training set. Here, the word non-parametric means that it makes assumptions on the underlying data distribution. Non-parametric methods do not have fixed numbers of parameters in the model. The parameters in the model grows with the training data set.

In the k-nn algorithm, the value for a new instance (x) is predicted by searching the training set for the k most similar cases (neighbours) and summarizing the output values for those k cases. In other words, this is the mode (or most common) value.

2.4.1.1 Choosing an Appropriate K Value

The most important thing to do in the knn algorithm is to determine an appropriate value of k, that is, the number of nearest neighbours. A large value of k reduces the noisy data. But if we have more data points in one group then we may ignore the smaller patterns which may have useful insights.

k-nn uses all of the data for training while classifying a new data point or instance.

2.4.1.2 Example of Knn Algorithm

Suppose we have n number of students rated on two parameters – Academic Score and EC Score on a scale of 1 to 10 as given in **Table 2.1**.

TABLE 2.1 Academic Score and EC Score

Name	Academic Score	EC Score	Grade
Ria	8	8	Outstanding
Khushank	9	1	Academically Sound
Mehar	4	8	Sporty
Nagma	2	1	Below Average

Then, academic score is an indication of how well the student performs academically and EC score is the score obtained by the student in extra-curricular activities.

Now, if we have a new student to categorize then we will calculate distance between ‘New Student’ and its nearest neighbours (‘Outstanding’, ‘Sporty’, ‘Academically Sound’ and ‘Below Average’) using the Euclidean distance formula as shown in **Fig. 2.15**.

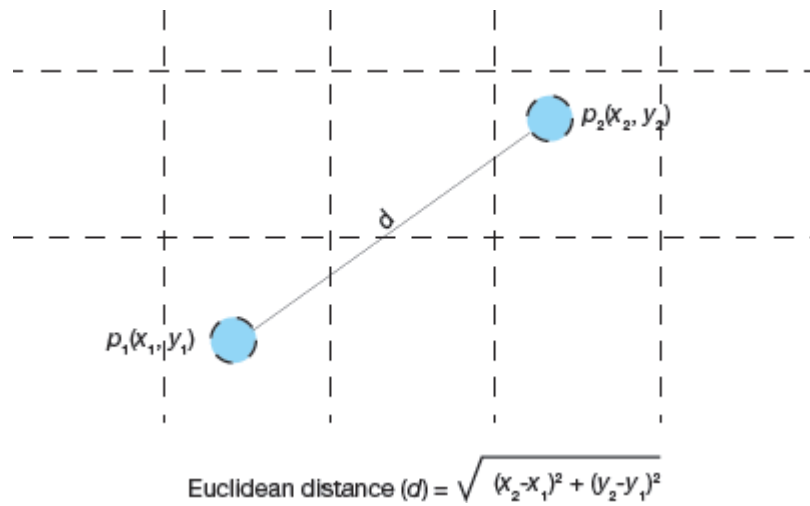


FIGURE 2.15 Euclidean distance between two points

Assuming that co-ordinates of the New Student are (8,2), we calculate the distance between Below Average (2,1) and New Student as:

$$\text{dist}(\text{Below Average, New Student}) = 6.08$$

Similarly, calculate the distance of New Student from each of its nearest neighbours as shown in **Table 2.2**.

TABLE 2.2 Scores including New Data

NAME	Academic Score	EC Score	Grade	Distance to new Student
Ria	8	8	Outstanding	6.08
Khushank	9	1	Academically Sound	1.41
Mehar	4	8	Sporty	7.21
Nagma	2	1	Below Average	6.08

We see that the distance between New Student and Academically Sound is the least so the New Student belongs to the group of academically sound students.

k-NN algorithm can also be used for prediction. It is extensively used in pharmaceutical industry to detect the growth of oncogenic (cancer) cells or presence of disease.

Another Example

Consider **Fig. 2.16**. If we have to two classes – class A denoted by red stars and class B indicated by blue triangles, then to classify a new green square object the value of k plays an important role. If $k=3$, then according to **Fig. 2.16**, the 3 nearest neighbours belong to class B. So, the new object is placed in class B. but if we take $k = 7$, then the new object will be a member of class A. If we take $k = 1$, then the new object will belong to class B. Hence, value of k greatly influences the classification.

The K-nearest neighbour can be used for regression when dependent variable is continuous. In this case, the predicted value is the average of the values of its k nearest neighbours.

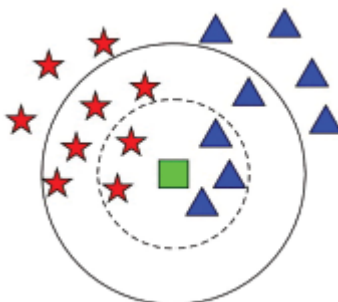


FIGURE 2.16 Finding nearest neighbor for the new data point (green). The new data will either belong to the group represented by red star or to the one represented as blue triangle

2.4.1.3 Pros and Cons of Knn Algorithm

Pros

1. The kNN algorithm is a very simple and effective algorithm.
2. Users can easily implement the algorithm which has made the algorithm quite popular with data professionals.
3. The algorithm is highly unbiased in nature and does not make any prior assumption of the underlying data.
4. The kNN algorithm works well with multi-class algorithms
5. The algorithm can be applied to both classification and regression.

KNN is a non-parametric learning algorithm, which means that it doesn't assume anything about the underlying data.

Cons

1. Usually, simple things are not very powerful; same is the case with kNN algorithm.
2. Although the algorithm trains the model really fast, the prediction time is high
3. Many a time, useful insights may get ignored.
4. The algorithm is sensitive to the scale of data so data must first be standardized
5. The performance of the algorithm deteriorated when there are multiple independent variables
6. The algorithm requires more memory and is computationally expensive
7. It does not work well if the target variable is skewed
8. The accuracy of the algorithm depends on k value. For any given problem, a small value of k will lead to a large variance in predictions. And setting k to a large value may lead to a large model bias.

Handling categorical variables in KNN

If you have categorical variables, then create k dummy variables out of that categorical variable. For example, if a categorical variable named “Designation” has 3 unique levels / categories then create 3 dummy variables. Each dummy variable has 1 against its designation and 0 otherwise.

2.4.2 Decision Trees

Do you remember the ‘Guess What’ game that we used to play in our childhood? One of us will think of something, and the others had to guess what it is. For that they can ask questions, answers of which will be either ‘Yes’ or ‘No’ based on these clues, the answer is given. For example, consider the tree given in **Fig. 2.17**. (Note, it is just a sample tree, not a gen-

eral tree). According to the rules specified, if you are thinking of an animal that is herbivorous and does not live-in forest then it is a rabbit.



FIGURE 2.17 Decision Tree

The same concept is used to create a decision tree for any data science problem. Now, let us study in detail the math behind it.

2.4.2.1 Basic Introduction

Decision tree is one of the most intuitive and popular techniques in data mining that provides explicit rules for classification. It works well with heterogeneous data and predicts the target value of an item by mapping observations about the item.

A decision tree represents choices and their results in form of a tree. The nodes represent an event or choice and the edges of the graph represent the decision rules or conditions. Some common applications of decision tree include the following:

1. Predicting an email as spam or not spam
2. Predicting of a tumour is cancerous or not
3. Predicting a loan as a good or bad
4. Predicting credit risk based on certain factors

Basically, decision trees can perform either classification or regression. For example, identifying a credit card transaction as genuine or fraudulent is an example of classification but using a decision tree to forecast prices of stock would be regression task.

Unlike linear models, decision trees map non-linear relationships quite well.

2.4.2.2 Principle of Decision Trees

Decision trees can be used to divide a set of items into n predetermined classes based on a specified criterion. They belong to a class of **recursive partitioning algorithms** that are simple to describe and implement. To create a decision tree, the following steps are necessary.

Step 1: Select a variable which best separate the classes. Set this variable as the root node.

Step 2: Divide each individual variable into the given classes thereby generating new nodes. Note that decision trees are based on forwarding selection mechanism; so, after a split is created, it cannot be re-visited.

Step 3: Again, select a variable which best separates the classes.

Step 4: Repeat steps 2 and 3 for each node generated until further separation of individuals is not possible.

Now, each terminal (or leaf) node consists of individuals of a single class. That is, once the tree is constructed and division criterion is specified, each individual can be assigned to exactly one leaf. This is determined by the values of the independent variables for the individual.

Moreover, a leaf is assigned to a class if the cost of assigning that leaf to any other class is higher than assigning the leaf to the current class. After all the leaf nodes are assigned a class, we calculate the error rate of the tree (also known as the total cost of tree or risk of the tree) by starting with an error rate of each leaf.

Thus, we see that a decision tree is a supervised learning predictive model that uses rules to calculate a target value. This target value can be either:

- a continuous variable, for regression trees. Such trees are also known as continuous variable decision trees. For example, predict the price of a product or income of a customer (based on his occupation, experience, etc.).
- a categorical variable, for classification trees. Such trees are also known as categorical variable decision tree. Example, to if the target value is 'Yes' or 'No' - whether a customer will buy a product or not.

Programming Tip: Decision tree works for both categorical and continuous input and output variables.

Pruning the Tree

When the decision tree becomes very deep, it must be pruned as it may contain some irrelevant nodes in the leaves. Therefore, pruning avoids creating very small nodes with no real statistical significance.

An algorithm based on decision tree is said to be good if it creates the largest tree and automatically prunes it after detecting the optimal pruning threshold.

The algorithm should also use Cross-validation technique and combine error rates found for all possible sub-trees to choose the best sub-tree.

A tree with each node having no more than two child nodes is called binary tree.

For example, if information gain of a node is -10 (loss of 10) and then the next split on that gives us IG of 20, then a simple decision tree will stop at step 1 but in pruning, the overall gain would be taken as +10 and both the leaves will be retained.

2.4.2.3 Key Terminology

Some important terms that are frequently used in decision trees (and demonstrated in **Fig. 2.18**) are as follows:

1. **Root node:** This is the node that performs the first split.
2. **Terminal nodes/Leaves:** These nodes predict the outcome.
3. **Branches:** They are depicted by arrows that connect nodes and shows the flow from question to answer. Technically, a branch is a sub section of entire tree
4. **Splitting:** This is the process of dividing a node into two or more sub-nodes. In a decision tree, splitting done until a user-defined stopping-criteria is reached. For example, the programmer may specify that the algorithm should stop once the number of items per node becomes less than 30.

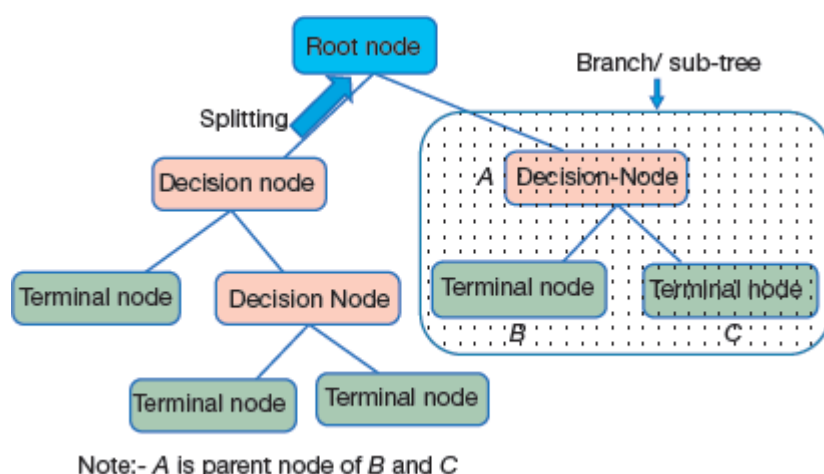


FIGURE 2.18

5. **Decision node:** This is a sub-node that splits into further sub-nodes
6. **Terminal or Leaf node:** It is a sub-node that does not split further.
7. **Parent node:** A node which splits into sub-nodes is called a parent node of the sub-nodes (or child of the parent node).

2.4.2.4 Advantages of Decision Trees

Decision trees are one of the most popular statistical techniques because it provides following advantages to users.

1. It is easy to implement

2. Results can be easily understood. It does not require any statistical knowledge to read and interpret them.
3. Programmers can code the resulting model
4. It executes faster when the model is applied to new individuals.
5. Outliers or extreme individuals can not affect decision trees.
6. Some decision tree algorithms can deal with missing data.
7. A decision tree allows users to visualize the results and represents all factors that are important in decision making.
8. Decision trees provide a clear picture of the underlying structure in data and relationships between variables.
9. They can also be used to identify the most significant variables in the data-set
10. Decision trees closely mirror human decision-making compared to other regression and classification approaches.
11. Decision trees requires less data cleaning as compared to other techniques.
12. They are not influenced by missing values to a fair degree.
13. They can handle both numerical as well as categorical variables.
14. Decision trees employs **non parametric method as they** have no assumptions about the space distribution and the classifier structure.

OUTLIERS

An outlier is an individual data item or observation that lies at an abnormal distance from other data values in a random sample of a data set. It is the work of the data analyst to decide what will be considered abnormal. Outliers should be carefully investigated as they may contain valuable information about the data being processed. Data analyst must be able to answer questions like the reasons for the presence of outliers in the data set, probability that such values will continue to appear, etc. for example, if an examination was held of maximum marks 100, then one or more students getting more than 100 will be treated as an outlier.

Outlier analysis is extensively used for detecting fraudulent cases. In some cases, outliers may be contextual outliers. This means that a particular data value is an outlier only under a specific condition. For example,

in hot summer season when temperature is 40 degrees and above, then an unexpected windstorm and rain may bring it to 32 degrees. Another example could be that a bank customer may not be withdrawing more than 50K in a month but because of a family wedding may withdraw 2.5 lakhs in a day.

2.4.2.5 Limitations of Decision Trees

The disadvantages of using decision trees are as follows:

1. The tree detects local, not global, optima.
2. All independent variables are evaluated sequentially, not simultaneously.
3. The modification of a single variable may change the whole tree if this variable is located near the top of the tree.
4. Decision trees lack robustness. Though this can be overcome by re-sampling, in which one can construct the trees on many successive samples and can aggregate by mean, but this will lead to a complex tree that is difficult to read and interpret.

The problem of overfitting can be done by tree pruning and setting constraints on tree size.

5. Since each split reduces the number of remaining records, later splits are based on very few records have less statistical power.
6. The property of forwarding variable selection and constant splits of nodes makes prediction by trees not as accurate as done by other algorithms.
7. Overfitting is a key challenge in modelling decision trees. If there is no limit set of a decision tree, it will give 100% accuracy on training set because in the worst case, there will be a single leaf for each observation.

2.4.3 Random Forests

Like decision trees, random forests are also a versatile machine learning technique that can perform both regression and classification. They give better performance than decision trees as it does everything for reducing the number of dimensions (or variables), treating missing values, outlier values and exploring data.

Random forests perform better than bagged trees as it *de-correlates* the trees. Like in bagging, random forests also create a number of decision trees on training data sets. For creating these trees, each split considers a *random sample of m predictors* as split candidates from the full set of predictors. A split uses only one of those m predictors. Therefore, the main difference between bagging and random forest is that while bagging chooses all predictors, random forest selects only one of the m predictors.

Let us summarize the possibilities.

1. If the number of cases in the training set is K , then a random sample of these K cases is taken as the training set.
2. Out of p input variables, specify a number m , such that $m < p$. At each node, m random variables out of p predictors are chosen. The best split on these m variables is used to split the node.
3. Each tree is subsequently grown to the largest extent possible
4. Take the average of all the predictions made by the target trees to finally predict new data.
5. To classify a new object based on attributes, each tree gives a classification and finally the classification occurring most frequently is chosen.

In simpler terms, we can understand the working of a random forest in the following steps.

Step 1: Random samples are selected from a given dataset.

Step 2: A decision tree is constructed for each sample to obtain a prediction result from each decision tree as shown in **Fig. 2.19**.

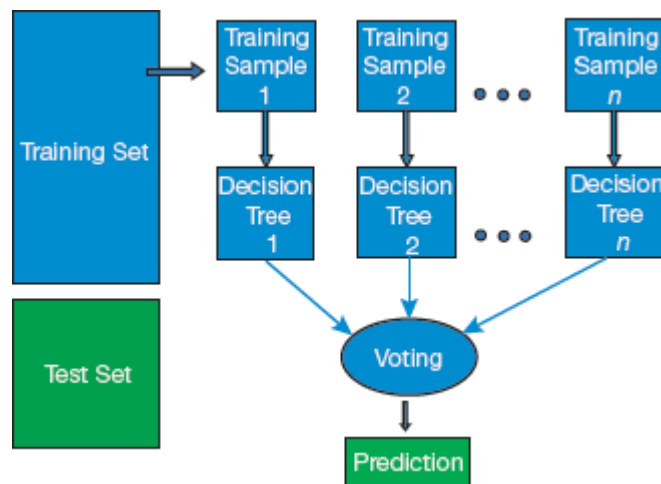


FIGURE 2.19 Random Forest

Step 3: Voting for each predicted result is performed.

Step 4: Prediction result with the maximum votes is selected as the final prediction.

Other major advantages of using Random forests include the following:

1. It can be very effectively used to estimate missing data
2. It maintains accuracy when a large proportion of the data is missing
3. It can balance errors in datasets where the classes are imbalanced
4. It can handle huge datasets with large number of dimensions

Limitations

1. Random forests might easily overfit noisy datasets, especially when performing regression.
2. Each tree is grown to the largest extent possible without pruning.
3. Random forests are slow in generating predictions as multiple decision trees are constructed and then the process of voting selects the best prediction result.
4. Random forests are difficult to interpret as and when compared to a decision tree model.

2.5 Clustering Techniques

Clustering is a set of techniques used to partition data into a number of groups, also called clusters. A **cluster** can be considered as a group of

data objects that are similar to other objects in the same cluster and dissimilar to data objects in other clusters. Clustering helps data scientists to identify two main qualities of data—meaningfulness and usefulness.

Meaningful clusters enhance domain knowledge. For example, in the healthcare industry, clustering is used to make groups of patients whose bodies responded differently to the same medical treatment.

Useful clusters serve as an intermediate step in a data pipeline. For example, businesses use this technique for making different clusters of customers depending on their purchasing patterns to create targeted advertising campaigns.

2.5.1 Overview of Clustering Techniques

Data objects can be clustered using a variety of techniques. Each of these techniques has its own strengths and weaknesses. Depending upon the input data, we need to select the most appropriate technique that result in more natural cluster assignment. To select the most suitable clustering algorithm for the dataset is not a trivial task. Some important factors that affect this decision include the characteristics of the clusters, the features of the dataset, the number of outliers, and the number of data objects. Based on these features, three popular categories of clustering algorithms include:

1. Partitional clustering
2. Hierarchical clustering
3. Density-based clustering

Partitional Clustering

It divides data objects into non-overlapping groups in such a way that every object belongs to one and only one cluster, and every cluster must have at least one object.

In such algorithms, users can specify the number of clusters (indicated by the variable k). These algorithms iteratively assign subsets of data points into k clusters. Two examples of partitional clustering algorithms are k -

means and k -medoids. Note that, these algorithms are **nondeterministic**. That is, such algorithms produce different results from two separate runs even if the same data was used in every run.

Advantages

1. Works well when clusters have a **spherical shape**.
2. **Scalable** with respect to algorithm complexity.

Limitations

1. Does not perform well with **complex cluster shapes** having different sizes.
2. Does not work with clusters of different **densities**.

Hierarchical Clustering

It creates clusters in a hierarchy either by using a bottom-up or a top-down approach.

In agglomerative clustering, bottom-up approach is used to merge two points that are most similar. This process is repeated until all points have been merged into a single cluster.

In divisive clustering, top-down approach is used in which the algorithm starts with all points as one cluster. In each iteration, the data objects that are least similar in the cluster are split until only single data points remain.

In such an algorithm, a tree-based hierarchy of points called a **dendrogram** is created. The number of clusters to be created, that is, k is determined by the user. Clusters are created by cutting the dendrogram at a specified depth that results in k groups of smaller dendrograms.

Hierarchical clustering is a **deterministic** process. Therefore, cluster assignments do not change when the algorithm is executed two or more times using the same input data.

Advantages

1. They highlight the **relationships** between data objects at a finer level.
2. Results are easily **interpretable**.

Limitations

1. They're **computationally expensive** with respect to algorithm complexity.
2. They are adversely affected by **noise** and **outliers** in the data set.

Density-Based Clustering

This technique creates clusters based on the density of data points in a region. A cluster is created where there are high densities of data points separated by low-density regions.

In contrast to other clustering categories, density-based approach does not require the user to specify the number of clusters. It uses a distance-based parameter that acts as a threshold. The threshold value helps the algorithm to determine how close points must be to be considered a cluster member.

Density-Based Spatial Clustering of Applications with Noise (or DBSCAN) and Ordering Points To Identify the Clustering Structure (or **OPTICS**) are popular examples of density-based clustering algorithms.

Advantages

1. Works well when clusters are of **non-spherical shapes**.
2. Performs well even on a data set having noise or outliers.

Limitations

1. Not well suited for clustering in **high-dimensional spaces**.
2. Difficult to identify clusters of **varying densities**.

2.5.2 K-Means Algorithm

We have seen that the clustering technique is used for finding sub-groups of observations within a data set. Clusters are created in such a way that observations in the same group are similar. Correspondingly, observations in different groups are dissimilar. K-means is an unsupervised learning technique as it finds relationships between n observations without being trained by a response variable. K-means clustering is the simplest and the most commonly used clustering method for splitting a dataset into a set of k groups.

2.5.2.1 Properties of Clusters

To understand the concept of clusters, let us take an example of a banking system that forms segments of its customers based on their income and debt. To start with, a scatter plot is used to visualize the customer data.

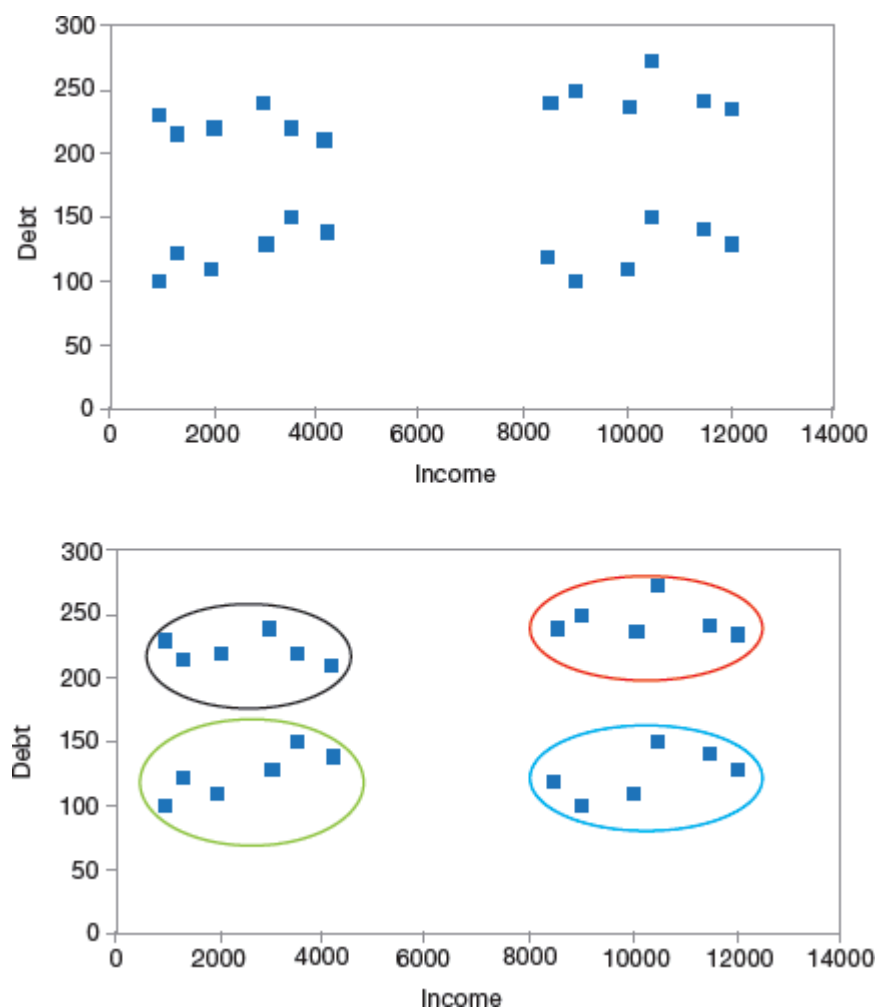


FIGURE 2.20 Cluster of Customers

Income is plotted on the X-axis and the amount of debt is plotted on the Y-axis. A very simple approach is to make four clusters as shown in **Fig. 2.20**. Looking at **Fig. 2.20**, we can conclude the following properties of a cluster.

All the data points in a cluster should be similar to each other. In the bank example, such a cluster allows the bank to perform targeted marketing. **The data points from different clusters should be as different as possible.**

2.5.3 Applications of Clustering in Real-World Scenarios

Clustering is a widely used technique across industry in almost every domain, ranging from banking to recommendation engines, document clustering to image segmentation.

Customer Segmentation

It is to target a specific set of customers or audience in the industry (telecom, e-commerce, sports, advertising, sales, etc.).

Document Clustering

Another very common application of clustering is document clustering in which multiple similar documents are clustered in a similar way as shown in **Fig. 2.21**.



FIGURE 2.21 Document clustering

Image Segmentation

In image segmentation, similar pixels in the image are clubbed together in the same group (refer **Fig. 2.22**).



FIGURE 2.22 Image Segmentation

Recommendation Engines

Clustering when used in recommendation engines can help us to recommend songs / friends/ products to our friends.

The algorithm makes a list of things liked by a person and then finds similar things to recommend them to that person.

2.5.4 Evaluation Metrics for Clustering

The goal of any clustering algorithm is not just to make clusters, but to make good and meaningful ones. For example, consider the two cases given in **Fig. 2.23**.

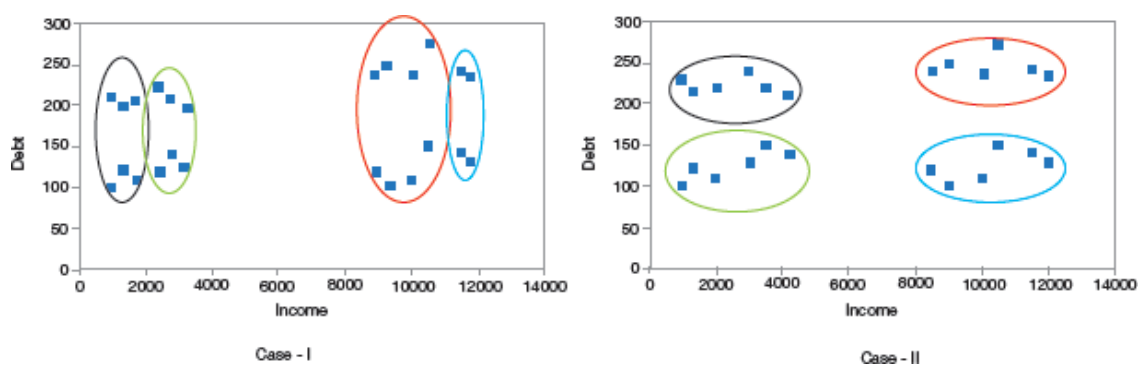


FIGURE 2.23 Different ways of clustering

Since here we have just two features, but in real-world applications, where we have too many features, visualizing all the features together and deciding which clusters are meaningful is just not possible.

Therefore, in such a situation, we use evaluation metrics to evaluate the quality of the clusters.

Inertia

Inertia indicates how far the points within a cluster are. For this, **sum of distances of all the points within a cluster from the centroid of that cluster is calculated. Once inertia is calculated for all the clusters, the final value of inertia is obtained by adding all these values. This final value that gives the distance within the clusters is known as intra-cluster distance.** So, inertia gives us the sum of intra-cluster distances. For best results (or clusters), the inertial value **should be as low as possible.**

Dunn Index

If the distance between the centroid of a cluster and the points in that cluster is small, it means that the points are closer to each other. So, the inertial value ensures that the first property of clusters is satisfied. However, it does not say anything about the second property – that different clusters should be as different from each other as possible. So, for this, we use the Dunn index.



FIGURE 2.24 (a) Intra cluster distance; (b) Inter cluster distance

Along with the distance between the centroid and points, **the Dunn index also considers the distance between two clusters. The distance between the centroids of two different clusters is known as inter-cluster distance.** The expression for Dunn Index can be given as,

$$\text{Dunn index} = \frac{\min(\text{inter cluster distance})}{\max(\text{intra cluster distance})}$$

That is, Dunn index is the ratio of the minimum inter-cluster distances and maximum of intra-cluster distances. Its value must be high for getting better clusters.

We must maximize the value of Dunn index. More the value of the Dunn index, better will be the clusters. For a higher Dunn index value, the numerator should be maximum. So, the distance between even the closest clusters should be more so that every cluster is far away from each other. Correspondingly, denominator should be minimum. So, intra-cluster distances should be more.

2.5.5 How K-Means Algorithm Works?

When implementing the k-means clustering algorithm, the following steps are performed.

First specify the number of clusters (k) that will be generated in the final solution. The algorithm starts by randomly selecting k observations (also known as centroids) to serve as the initial centres for the clusters.

In the next step, every remaining observation is assigned to its closest cluster, where closest is a term calculated using the Euclidean distance between the centroid of the cluster and the observation. This step is called 'cluster assignment step'.

The algorithm then computes the new mean value of each cluster. Therefore, this step is known as 'centroid update'.

Once the centers are re-calculated, every observation is rechecked to determine its closeness with the same and other clusters. All the observations may have to be reassigned a cluster based on the updated cluster means.

The cluster assignment and centroid update steps are repeatedly performed until the cluster assignments remains the same over iterations (technically, until *convergence* is achieved).

2.5.6 Pros and Cons of K-Means Algorithm

Pros

1. K-means clustering algorithm is very simple and fast.
2. The algorithm can efficiently deal with very large data sets.

Cons

1. Number of clusters must be pre-specified.
2. The algorithm is sensitive to outliers
3. Changing the order of data will give different results.
4. Create a random data set and then apply the k-means clustering algorithm.

2.5.7 Density-Based Spatial Clustering of Applications with Noise (Dbscan)

We have seen that clustering is an unsupervised learning technique that forms groups of the data points. All points in the same group have similar properties and those in different groups exhibit a different set of properties.

Clusters or groups in data are formed using a variety of distance measures like K-Means (distance between points), affinity propagation (graph distance), mean-shift (distance between points), DBSCAN (distance between nearest points), Gaussian mixtures (Mahalanobis distance to centres), spectral clustering (graph distance), etc.

Though all clustering analysis techniques use the same approach (calculating similarities and then clustering data points into groups), here in this section, we will explore **Density-based spatial clustering of applications with noise (DBSCAN)** technique.

Why DBSCAN? K-Means clustering, clusters loosely related observations together. Every observation or data point is eventually a part of some cluster even if the observations are scattered far away in the vector space.

The cluster to which an observation will be associated with depends on the mean value of cluster elements. Even a small change in data points *may alter* the clusters formed. However, this problem is greatly reduced in DBSCAN.

Another issue with *k*-means algorithm is that you need to specify the number of clusters (*'k'*) to be formed and we often err to specify the most

optimal value for k . In DBSCAN, the value of k need not be specified. We just need to have an idea about the distance between values and some guidance for what amount of distance is considered 'close'. DBSCAN produce better results than k -means. This is evident from **Fig. 2.25**.

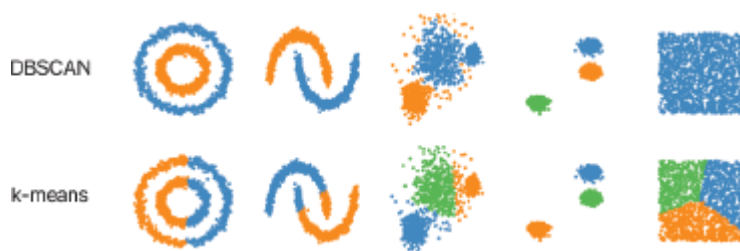


FIGURE 2.25 DBSCAN and k-means Clustering Algorithms

2.5.7.1 Understanding Density-Based Clustering Algorithms

DBSCAN is an unsupervised learning technique that identifies distinct clusters in the data. A cluster is identified as a contiguous region of high point density, separated from other such clusters by contiguous regions of low point density.

The algorithm discovers clusters of different shapes and sizes from a large amount of data that may contain several outlier values.

Two important parameters of the DBSCAN algorithm are discussed here:

minPts which indicates the minimum number of points (a threshold) clustered together for a region to be considered dense.

As a rule of thumb, value of *minPts* can be derived from the number of dimensions D in the data set. Generally, $\text{minPts} \geq D + 1$. Setting $\text{minPts} = 1$ is of no use as then every point will form a cluster of its own. With $\text{minPts} \leq 2$, the result will be similar to hierarchical clustering. Therefore, minimum value of *minPts* should be at least 3. However, for noisy data sets we must set it to a larger value to get more significant clusters. To be more specific, we can choose $\text{minPts} = 2 \cdot \text{dim}$ to choose larger values for dataset that is very large, or is noisy or contains duplicate values.

eps (ϵ) is the distance measure that will be used to locate the points in the neighbourhood of any point.

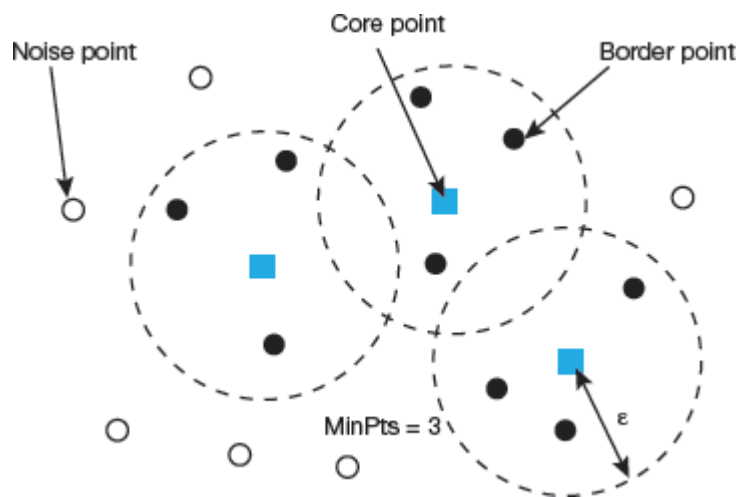


FIGURE 2.26 Terminology used in DBSCAN

The value for ϵ can be chosen by using a k -distance graph, in which the distance is plotted to the k nearest neighbour ordered from the largest to the smallest value. Here, the value of k can be set to $minPts-1$. Optimal values of ϵ can be noted from this plot from the location where an ‘elbow’ is present. If ϵ is too small, a large part of the data will not be clustered; whereas for a large value of ϵ , clusters will merge and most of the points will be in the same cluster. However, small values of ϵ should be preferred to have only a small fraction of points within this distance of each other.

These parameters can be understood by exploring the concepts of density reachability and density connectivity.

Density reachability establishes a point to be reachable from another if it lies within a particular distance (eps) from it.

Connectivity is a transitivity-based chaining-approach that is used to determine points that are located in a particular cluster. For example, a and e points could be connected if $a \rightarrow b \rightarrow c \rightarrow d \rightarrow e$, where $a \rightarrow b$ means b is in the neighbourhood of a.

Once DBSCAN clustering is done, we can recognize three types of points.

Core, it is a point having at least m points within distance n from itself.

Border, is a point having at least one Core point at a distance n .

Noise, is a point that does not fall in the above two categories. This means that noise has less than m points within distance n from itself (refer **Fig. 2.26**).

2.5.7.2 Algorithmic Steps for Dbscan Clustering

Repeat until all points have been visited.

1. Arbitrarily pick up a point in the dataset ().
2. If there are at least 'minPoint' points within a radius of ' ϵ ' to the point then all these points are considered to be a part of the same cluster.
3. Expand the clusters by recursively computing the neighbourhood for each point

2.6 Naïve Bayes Classification

Data science has progressed from simple linear regression models to complex techniques but practitioners still prefer the models that are simple and easy to interpret. In this widely used category of algorithms Naïve Bayes algorithm is one of the prominent names as it is not only simple but so powerful that it outperforms complex algorithms for very large datasets.

Naive Bayes is a probabilistic machine learning algorithm based on the Bayes Theorem. This algorithm is extensively used in a wide range of classification tasks varying from filtering spam, classifying documents to predicting sentiments.

Programming Tip: The term naïve means that features that are used in the model are independent of each other. So when one feature change, the other is not affected.

2.6.1 Understanding Conditional Probability

To understand conditional probability, consider an example of flipping a coin. There are equal chances of getting either heads or tails. So, the probability of getting heads is 50%.

Now, in case we are supposed to find the probability of getting a queen spade from a deck of 52 cards then the denominator is 13 (the eligible population) and not 52. Since there is only one queen in 13 spades, the probability that we get a queen spade is $1/13$.

	Female	Male	Total
Teacher	8	12	20
Student	32	48	80
Total	40	60	100

FIGURE 2.27 Sample Dataset

Therefore, we can say that the conditional probability of A given B, denotes the probability of occurring A given that B has already occurred. Mathematically, it can be expressed as, $P(A | B) = P(A \text{ AND } B) / P(B)$.

Let's apply this concept on the data set given in **Fig. 2.27** and calculate the conditional probability that the teacher is a male.

So, the required conditional probability $P(\text{Teacher} | \text{Male}) = 12 / 60 = 0.2$.

Programming Tip: If Y has two categories, then calculate the probability of each class of Y and select the one with higher value.

$$P(\text{Teacher} | \text{Male}) = \frac{P(\text{Teacher} \cap \text{Male})}{P(\text{Male})} = \frac{12}{60} = 0.2$$

Likewise, the conditional probability of B given A can be computed as,

$$P(A | B) = \frac{P(A \cap B)}{P(B)} \quad (1)$$

$$P(B | A) = \frac{P(A \cap B)}{P(A)} \quad (2)$$

2.6.2 The Bayes Rule

The Bayes Rule uses $P(X|Y)$, known from the training dataset, to find $P(Y|X)$. For observations in test dataset, X would be known while Y is unknown. And for each row of the test dataset, we have to find the probability of Y given that X has already happened (refer **Fig. 2.28**).

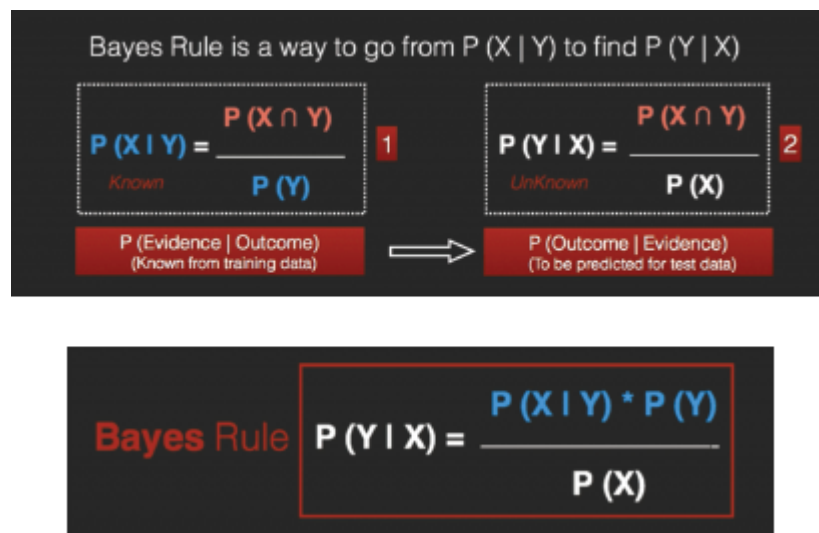


FIGURE 2.28 Bayes Rule for Conditional Probability

2.6.3 Types of Events

If two probabilities, $P(B)$ and $P(B|A)$ are the same, then it means that the occurrence of event A had no effect on event B . Therefore, event A and event B are *independent events*.

If the conditional probability is zero, then it means that the occurrence of event A implies that event B cannot occur. If the reverse is also true, then the events are said to be mutually exclusive events. In such a case, only one event can take place at a time.

All other cases are classified as dependent events where the conditional probability can be either lower or higher than the original.

For example, if we toss a coin twice and we want to calculate the probability of getting a head both times, then we must consider that the second event is independent of the first. The desired probability is $P(\text{Heads in first throw}) * P(\text{Heads in second throw}) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$

Mathematically, we can say that if the events are not independent, we multiply the probability of one event with the probability of the second event.

For expressing dependent events mathematically, we multiply the probability of any one event with the probability of the second event after the first has happened.

$$P(A \text{ and } B) = P(A) * P(B | A) \text{ or } P(A \text{ and } B) = P(B) * P(A | B)$$

For example, while drawing two cards (King and Queen) without replacement, the probability of the first event is dependent of 52 cards whereas the probability of the second event is dependent on 51 cards.

$$\text{This means that } P(\text{King and Queen}) = P(\text{Queen}) * P(\text{King} | \text{Queen})$$

So if $P(\text{Queen})$ is $4/52$ then once the King is drawn, $P(\text{King} | \text{Queen})$ is $4/51$ as now only 51 cards are left. Therefore, the desired probability = $4/52 \times 4/51$

The probability of two non-mutually exclusive events is given as, $P(A \text{ OR } B) = P(A) + P(B) - P(A \text{ AND } B)$.

For example, in a throw of dice, the probability of getting a number that is multiple of 2 or 3 is a scenario of events which are not mutually exclusive since 6 is both a multiple of 2 and 3 and is counted twice. Therefore,

$$P(\text{multiple of 2 or 3}) = P(\text{Multiple of 2}) + P(\text{Multiple of 3}) - P(\text{Multiple of 2 AND 3}) = P(2, 4, 6) + P(3, 6) - P(6) = 3/6 + 2/6 - 1/6 = 4/6 = 2/3$$

2.6.4 Naive Bayes Algorithm

In real-world problems, there are multiple X variables. So, when the features are independent, the Bayes Rule can be extended to what is called Naive Bayes. It is called 'Naive' because of the naive assumption that the Xs are independent of each other.

Naive Bayes classification algorithm can be used with continuous features but performs best for categorical variables. When dealing with numeric features, the algorithm assumes that numerical values are normally distributed.

2.6.5 Laplace Correction

We can better understand the importance of Laplace correction with an example. If we were to classify fruits as mango, orange or banana, then the value of $P(\text{Long} \mid \text{Orange})$ will be zero as there are no 'Long' oranges.

When we use this probability in the model having multiple features, then the entire probability will become zero as anything multiplied by zero is zero. To avoid this, we need to increase the count of the variable with zero to a small value (usually 1) in the numerator, so that the overall probability does not become zero.

This correction is called 'Laplace Correction' and the function for building the Naive Bayes model will accept this correction as a parameter.

2.6.6 Pros and Cons of Naive Bayes Algorithm

Pros

1. It is a simple algorithm that is easy to build and understand.
2. It predicts classes faster than many other classification algorithms.
3. It can be easily trained using a small data set.
4. It can be used to predict values for even large data sets.

Cons

- If a given class and a feature has 0 frequency, then the conditional probability will become 0 resulting in the 'Zero Conditional Probability Problem.' This is a serious issue as it wipes out all other vital information about probabilities. However, to avoid such situations there are several sample correction techniques like the 'Laplacian Correction.'
- Another disadvantage is the very strong assumption of independence class features that it makes. It is near to impossible to find such data

sets in real life.

2.6.7 Applications

Some of the real-world scenarios where Naive Bayes algorithm is used include the following:

1. **Text classification:** Naïve Bayes classification algorithm is one of the most preferred algorithms to classify whether a text document belongs to one or more categories (classes).
2. **Spam filtration:** Many popular email services use the Bayesian spam filtering to distinguish spam email from legitimate email. Many server-side email filters like DSPAM, SpamBayes, SpamAssassin, Bogofilter, and ASSP are all based on Naïve Bayes classification technique.
 1. **Sentiment Analysis:** The Naïve Bayes algorithm is also used to analyse the tweets, comments, and reviews posted on social networking sites to classify as being negative, positive or neutral.
 2. **Recommendation System:** Companies use recommendation systems to predict whether a particular user will like their product and/or buy their product or not. For this, the Naive Bayes algorithm is used in conjunction with collaborative filtering techniques to build hybrid recommendation systems.

2.7 Neural Network

Neural Network (NN) or Artificial Neural Network (ANN) is a machine learning algorithm that is inspired by the biological neuron system and learns by examples. It consists of a large number of highly interconnected processing elements called neurons to solve problems. The algorithm follows a non-linear path and information is processed in parallel throughout the nodes. The best part of neural network is that it can change its internal structure by adjusting weights of inputs.

Neural network algorithms were developed to solve problems which are easy for humans but difficult for machines. These problems include pattern recognition in which we need to identify similar pictures or patterns.

Because of its ability to do pattern recognition, neural networks application ranges from optical character recognition to object detection.

2.7.1 Working of Neural Networks

Do you in human's brain, dendrites of a neuron receive input signals from another neuron and gives output based on those inputs to an axon of some other neuron. The human brain comprises billions of interconnected web of neurons that process information makes the brain to think and take decisions.

Similarly, the neural network algorithm works using a set of connected input/output units. In this structure, each connection has a weight associated with it. In the learning phase, the network learns by adjusting the weights to predict the correct class label of the given inputs.

The learning phase is used along with back propagation error method. Therefore, when error is calculated at the output unit, it is back propagated to all the units because error at each unit contributes to total error at the output unit. The errors at each unit are then used to optimize the weight at each connection.

The output of a neuron may vary from $-\infty$ to $+\infty$. So, we need a mapping function also known as the Activation function that maps inputs to the outputs.

A Simple Neural Network Model

In **Fig. 2.29**, we can see that it is a feedback neural network in which information is passed in both directions (forward as well as backwards). The structure of the neural network can change over time based on the inputs. Though the figure shows that there is one input layer, one hidden layer and one output layer but we can have multiple layers of hidden layers as shown in **Fig. 2.30**.

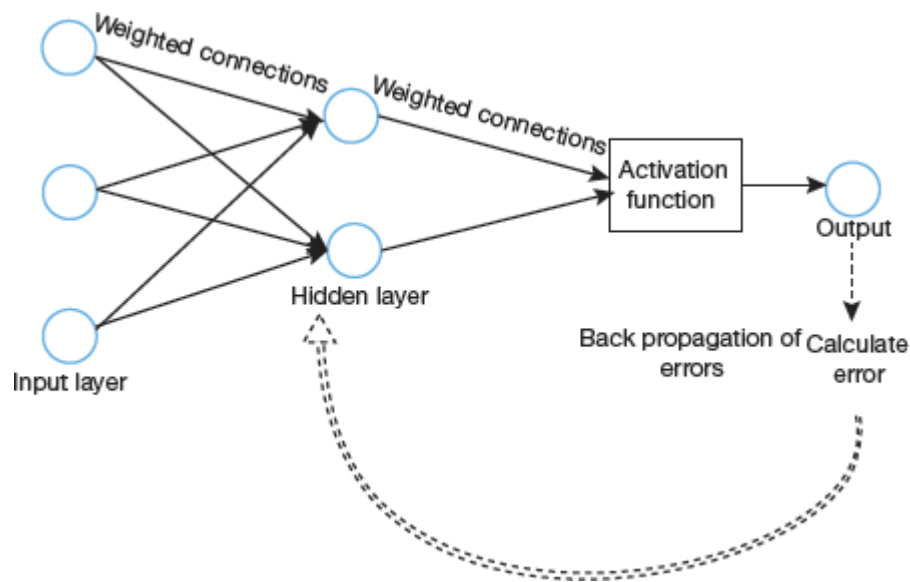


FIGURE 2.29 Feedback neural network

FIGURE 2.30 Neural Network with multiple hidden layers

Here, the input layer is the first layer of the neural network and receives the raw input. It processes the input and passes the processed information to the hidden layers. The hidden layer passes the information to the last layer, which gives the final output.

The output of a neuron can range from $-\infty$ to $+\infty$. The neuron doesn't know the boundary. So, we need a mapping mechanism between the input and output of the neuron. This mechanism of mapping inputs to output is known as Activation Function.

2.7.2 Pros and Cons

Pros

1. Neural network is a flexible algorithm that can be used to solve both regression and classification problems.
2. Neural networks perform well on nonlinear dataset with a large number of inputs such as images.
3. Neural networks can work with any number of inputs and layers.
4. Neural network works very fast as compared to other classification algorithms since it performs calculations in parallel.

Cons

1. Algorithms like Decision Tree and Regression that are simple, fast, easy to train, and provide better performance are seen as a preferred choice for classification and regression problems because
2. Neural networks require more time for development and needs more computation power.
3. Neural Networks needs more data than any other Machine Learning algorithm.
4. Neural networks can be performed only on numerical inputs and non-missing value datasets.

2.7.3 Applications of Neural Networks

Neural network algorithms are extensively used in the following fields.

1. **Pattern recognition** applications like facial recognition, object detection, fingerprint recognition, etc. uses neural network algorithms.
2. **Anomaly detection** applications that are specifically designed to detect unusual patterns that don't fit in the general patterns use neural networks because of their ability to perform well in pattern recognition tasks.
3. **Time series prediction** applications like predicting stock prices, forecasting weather prefer to use neural networks.
4. **Natural language processing** applications including text classification, Named Entity Recognition (NER), Part-of-Speech Tagging, Speech

Recognition, and Spell Checking which are used in a wide range of applications makes use of neural networks algorithms.

2.7.4 How Neural Networks Work?

We have seen that neural networks take several inputs, processes it through multiple neurons from multiple hidden layers, and returns the final result using an output layer. This process is known as '**forward propagation**'.

To make a perfect model, we need to minimize the value or weight of neurons that are contributing more to the error. For this, we need to travel back to the neurons of the neural network and find where the error lies. This process is known as '**backward propagation**'. We can reduce the number of iterations to minimize the error by using the 'gradient descent' algorithm.

A simple strategy of creating input-output relationships is discussed below.

1. **Use a subset of inputs to calculate the output.** Take only those inputs that satisfy a given threshold value. For example, if the threshold value is 0, then if $x_1 + x_2 + x_3 > 0$, the output is 1 and 0 otherwise.
2. **Add weights to the inputs.** Weights give importance to an input. For example, if $w_1=4$, $w_2=5$, and $w_3=6$ then on assigning these weights to x_1 , x_2 and x_3 , we will multiply input with their weights and compare the result with threshold value. That is, $w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 > \text{threshold}$. As per the details provided, more importance has been given to x_3 in comparison to x_1 and x_2 .
3. **Add bias:** Each perceptron has a bias that indicates how flexible the perceptron is. Therefore, linear representation of input will now be, $w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + 1 \cdot \text{bias}$.

Neuron applies non-linear transformations (activation function) to the inputs and biases.

Now, back-propagation (BP) algorithms determine the loss (or error) at the output and then propagate it back into the network. The weights are updated to minimize the error resulting from each neuron. In this process, the first step is to determine the gradient (derivatives) of each node with respect to the final output.

Note that one round of forwarding and backpropagation iteration is known as one training iteration or **Epoch**.

2.7.5 What is an Activation Function?

Activation function takes the sum of weighted input ($w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + 1 \cdot b$) as an argument and returns the output of the neuron.

$$a = f\left(\sum_{i=0}^n w_i x_i\right)$$

The activation function is usually used to make a non-linear transformation that allow us to fit nonlinear hypotheses or to estimate the complex functions. Some commonly used activation functions include ‘Sigmoid’, ‘Tanh’, ReLu, etc.

2.7.6 Gradient Descent

There are two variants of gradient descent algorithm- Full Batch Gradient Descent and Stochastic Gradient Descent (SGD). Both these variants update the weights of the neurons. Although the underlying technique is the same, the only difference lies in the number of training samples used to update the weights and biases.

While Full Batch Gradient Descent Algorithm uses all the training data points to update each of the weights at once, the Stochastic Gradient uses 1 or more data points but never the entire training data for updating at once.

For example, if we have a dataset of 10 data points with two weights **w1** and **w2**, then

In a **Full Batch** algorithm, all 10 data points are used to calculate the change in w_1 (Δw_1) and change in w_2 (Δw_2) and update w_1 and w_2 accordingly.

However, with SGD algorithm, initially the first data point is used to calculate the change in w_1 (Δw_1) and change in w_2 (Δw_2) and update the weights w_1 and w_2 . Then, the second data point is used to update weights

An activation function is what makes a neural network capable of learning complex non-linear functions.

Neural Networks and Deep Learning

Majority of AI systems are supported by breakthroughs in machine learning and deep learning techniques. These techniques together go hand-in-hand to deliver an efficient system that sometimes it is just difficult to understand the difference between artificial intelligence, machine learning and deep learning. However, venture capitalist Frank Chen highlighted this difference by stating that *'Artificial intelligence is a set of algorithms and intelligence to try to mimic human intelligence. Machine learning is one of them, and deep learning is one of those machine learning techniques.'*

Deep learning is inspired from human brain and the neurons in the human brain. Therefore, to understand deep learning, we need to first recapitulate how neurons in our brains work. Have you wondered,

How a child distinguishes between a car and a bike?

How do we perform complex pattern recognition tasks?

How do we differentiate between a male voice and a female voice?

The answer is that our brains have a huge, connected and a complex network of about 10 billion neurons which are, in turn, connected to another 10,000 neurons. This network of neurons, or what we call as, a neural network, lays the foundation of deep learning. So, let us first get an introduction on neural networks.

Example: A school has to select a student for the prestigious title, 'STUDENT OF THE YEAR'. The principal forms a jury of three people for this task. The jury decides that the selection criteria would be- MARKS, EMOTIONAL_QUOTIENT, IQ_ LEVEL, GRADES_EXTRA_CURRICULAR.

The school has a history of fair selection procedure and to continue the same standard, the principal shares with the jury, data of about 10 students previously selected to give the jury an opportunity to practice, which will eventually help them make a fair selection.

Every jury member is given a maximum of 10 points (weight) on which they would rate a student. These 10 points can be distributed across the four criteria. The cut-off average to qualify the selection process is '6'. So, a student must have an average score of ≥ 6 .

SCORE OF STUDENT 1

Jury Member 1: The first jury member gives the entire prominence to marks. Therefore, allots 8 points to Student 1 just on this criterion.

Jury Member 1: The second member distributes the points across all criteria except grades in extra-curricular activities.

Jury Member 3: The third jury member finds emotional quotient and grades in extracurricular activities more important and thus allocated points based on performance in these areas.

Finally, the average score of Student 1 is calculated as- $(8 + 6 + 7) / 3 = 7$.

Since this score is > 6 , Student 1 qualifies round 1. But the principal revealed that this student actually did not get selected in round 1 as per the

original decision.

Now, the jury members were asked to adjust their scoring process to see if they can correctly predict Student 2's selection or not.

JURY MEMBER 1: Member thinks that excess weightage has been assigned to Marks, so now considers IQ_Level also to assign points.

JURY MEMBER 2: Now distributes, points across all four criteria.

JURY MEMBER 3: Now, allots points to all criteria except MARKS.

The average score of Student 2 is $(4 + 4 + 5) / 3 = 4.3$

Since this score is less than 6, Student 2 is not selected. However, this time principal reveals that decision of the jury was indeed correct. Student 2 was rejected in the original selection process.

In this way, the jury evaluates records of other candidates and in the process learns a pattern for assigning right 'weightage' for each criterion. Of course, the weightage that gives the highest number of correct predictions is kept for evaluating the upcoming contest.

This entire process of learning and developing an accuracy is nothing but Artificial Neural Networks (ANN).

Deep Learning

Deep learning (also known as deep neural learning or deep neural network) is an AI technique that imitates the working of human brain in processing data and creating patterns for use in decision making. It is a sub-

set of machine learning that uses unsupervised learning to learn from unstructured or unlabelled data.

In today's scenario we generate massive amount of data (approx..2.6 quintillion bytes) every day. This huge data paves the way for application of deep learning techniques (refer **Fig. 2.31**). Deep learning algorithms flourish when applied on tonnes of data processed using stronger computing power.

When applied on very diverse, unstructured and inter-connected data, deep learning can be used to solve complex problems. The deeper learning algorithms learn, the better they perform.

FIGURE 2.31 How do data science techniques scale with amount of data?

Deep learning is usually used to detect objects, recognize speech, translate languages, recognize similar images, recommend products, make important decisions, detect fraud or money laundering, and explore the possibility of reusing drugs for new ailments among other functions. Its use became unprecedented with massive explosion of data world-wide. The big data that we collect from social media, internet search engines, e-commerce platforms, online cinemas, etc are unstructured. Such data when analysed using deep learning techniques, can unravel wealth of information and can be used with AI systems for automated support. **Figure 2.32** highlights the relation between AI and DL

Deep learning is the key technology behind driverless cars. Such cars are designed to recognize a stop sign, or distinguish a pedestrian from a lamppost. Basically, deep learning model learns to perform classification directly from images, text, or sound. For better accuracy, such models are trained by using a large set of labelled data and neural network architectures containing several layers.

Deep learning uses a hierarchical level of artificial neural networks to carry out the process of machine learning.

How Does Deep Learning Work?

We have studied that neural networks are layers of nodes. It is designed in the same way the human brain is made up of neurons. Nodes within individual layers are connected to adjacent layers. While in the human brain, a single neuron receives thousands of signals from other neurons, neural networks work in the same way. A message or a signal travels between nodes and weights are assigned to every node.

A node with a higher weight will have more weightage on the next layer of nodes. The final layer computes the weighted inputs to produce the final output.

A neural network is said to be a deep neural network depending on the number of layers it has. A deep learning system needs more powerful hardware as it has more layers and perform several complex mathematical calculations on a large amount of data. Large data sets ensure accurate results. For example, a facial recognition program starts with detecting and recognizing edges and lines of faces and then gradually learns to detect other significant parts of the faces. The process continues till the overall representations of faces are identified. Over time, the program trains itself with every iteration and thus the probability of correct answers increases. **Neural networks can be used to perform clustering, classification or regression.** The individual layers of neural networks can be thought of as a filter that increases the likelihood of giving the correct result as output.

FIGURE 2.32 DL is a subset of ML which in turn is just a sub-domain of AI

For example, when creating a neural network that identifies images having a dog, we need to train the network with pictures showing dogs at different angles and with varying amounts of light and shadow. These images in the training data set are converted into data, which moves through different nodes in the network. Prediction, identification or classification result is finally produced by the last layer, that is, the output layer of the neural network. The output produced by the neural network is then matched with the labels provided by any human. If the two values match, the output is confirmed otherwise the neural network notes the error and adjusts the weights provided to each node.

Some popularly used open-source deep learning libraries include Google Tensorflow, Facebook open-source modules for Torch, Amazon DSSTNE on GitHub, and Microsoft CNTK.

Machine Learning Vs Deep Learning

In a machine learning algorithm, relevant features are manually extracted from images and a particular classifier has to be manually implemented. The extracted features are then used to create a model that cate-

gorizes the objects in the image. In contrast, relevant features are automatically extracted from images by the deep learning model. Even the modelling steps are automatic (refer **Figs 2.33** and **2.34**).

FIGURE 2.33 Comparison between ML and DL

While a typical machine learning neural network model may have two to three hidden layers, deep networks on the other hand have as many as 150 hidden layers.

Moreover, deep learning performs ‘end-to-end learning’. This means that a neural network when given raw data and a task (like classification), the model automatically learns how to do this task.

Last but not the least, deep learning algorithms scale well with data. It is free from shallow learning converges problems that a typical machine learning model may face. Shallow learning refers to creation of plateau at a certain level of performance. Even after adding more data for training, the accuracy and performance of the model does not change.

While the biggest advantage of deep learning over machine learning techniques is that it frees users from the worry about trimming down the number of features used, but the main drawback is that a deep learning model is very expensive to train and needs commercial-grade GPUs. Many data scientists, treat deep learning model as a ‘black box’ because the model is very complex and extremely difficult to understand.

FIGURE 2.34 Deep Learning supports more automation using large number of hidden layers)

TABLE 2.3 Differences between machine learning and deep learning

When to Use ML or DL?

If you have a complex problem to solve with huge amounts of data and powerful hardware capabilities, then you must go for implementing a deep learning solution. But in case any of them is missing, choose the ML

model to solve your problem. **Table 2.4** compares the two techniques in more detail.

TABLE 2.4 Comparison between machine learning and deep learning

Note that in case you want to automate feature extraction, especially when a data set contains hundreds and thousands of features (or attributes) then, use deep learning. Since, only few features are actually useful, a deep learning system learns about the relevance of those features to select ones that will help the system to learn useful information. Feature extraction uses PCA, T-SNE or any other dimensionality reduction algorithms.

For example, an image processing application extracts the features in the image (like the eyes, the nose, lips and so on) and feeds it to the classification model. The deep learning model has layers of neural networks. The first layer learns small details from the picture; the next layers will combine the previous knowledge to make more complex information.

Applications of Deep Learning

Deep learning techniques have already outperformed humans in some tasks like classifying objects in images. For example, deep learning uses millions of images and thousands of hours of video to develop driverless cars. For this, it requires substantial computing power. High-performance GPUs (Graphics Processing Units) having a parallel architecture when combined with cloud computing allows development teams to reduce training time for a deep learning network from weeks to hours or even less. Besides, driverless cars, deep learning is also used in the following areas.

In aerospace and defence sector, deep learning is used to ***identify objects from satellites*** that locate areas of interest, and safe or unsafe zones for the military troops.

In medical field, cancer researchers use deep learning to automatically ***detect cancer*** cells.

Industrial automation uses deep learning to improve ***worker safety*** around heavy machinery. It automatically detects when people or objects comes within an unsafe distance of machines.

In electronic gadgets, deep learning is used in ***automated hearing and speech translation***. For example, home assistance devices that recognizes and responds to voice commands and knows user's preferences are trained using deep learning algorithms.

Digital assistants like Siri, Cortana, Alexa, and Google Now use deep learning for natural language processing and ***speech recognition***.

Deep learning assist in ***translation*** between languages. This can be very helpful for travellers, business people and those in government sectors. Skype and Google Translate uses deep neural networks to translate spoken conversations in real-time.

Several email systems including Gmail uses deep learning techniques to identify ***spam messages*** before they even reach the inbox.

PayPal uses deep learning to **prevent fraudulent** payments.

Apps like CamFind implements deep neural networks to allow its users to take a picture of any object and **discover details** of that object using mobile visual search technology.

Google Deepmind's WaveNet can generate speech **mimicking human voice** that sounds more natural than speech systems presently on the market.

Facebook uses deep learning identify and **tag friends** a user uploads a new picture.

Example, How Deep Learning is Used for Fraud Detection

To detect fraud, deep learning uses time, geographic location, IP address, type of retailer, and other features. In such a scenario, the first layer of the neural network processes raw data input (like the amount of the transaction) and passes it as an output to the next layer. The second layer accepts the output of the first layer as its input. This input is then processed by including additional information like the user's IP address and passes its result to the third layer.

The third layer may incorporate geographic location to make the results even better. This process continues across all levels of the neuron network. The final layer generates the output that may signal the analyst to freeze the user's account. Some applications of AI that makes extensive use of Deep Learning are illustrated in **Fig. 2.35**.

Deep learning is used by **chatbots** and service bots for providing **customer service**. Many companies use these techniques to respond in an intelligent and helpful way to text-based questions.

Image colourization activities use deep learning techniques to transform black-and-white images into coloured ones. This work was earlier done manually. But these days, deep learning algorithms generate impressive and accurate results by using the context and objects in the images to colour them to recreate the black-and-white image in colour.

Deep learning algorithms when used in medicine and pharmaceuticals, can be used to diagnose diseases and tumour in the body. They are also used to prescribe personalized medicines created specifically for an individual's genome.

Personalized shopping and entertainment can use deep learning techniques to suggest what users should buy or watch next.

FIGURE 2.35 Applications of Deep Learning

2.8 Support Vector Machine (SVM)

A support vector machine (SVM) is a classification algorithm that classifies data based on its features. An SVM will classify any new element into one of the two classes as shown in **Fig. 2.36**.

FIGURE 2.36 Classification Process using SVM

To make our machine learning model learn, we supply input data to it. The SVM algorithm will automatically extract features from the input data. This knowledge is used to segregate and classify the input data to generate the desired output. Once the model has learnt how to classify, any new data inputted to it can be classified (with a specific accuracy).

FIGURE 2.37 Identifying Best Line on the Plot for Classification

An SVM is therefore a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used to classify data. In the SVM algorithm, each data item is plotted as a point in n-dimensional space (where n is number of features). In this plot, support vectors are simply the co-ordinates of individual observation. For classification, we need to identify the best hyper plane/line or class (refer **Fig. 2.37**). For example, consider the scenarios given below.

Scenario 1: In **Fig. 2.38**, there are three hyper planes A, B and C. However, hyper planes B and C better segregates the two classes.

Scenario 2: When all three hyper-planes (A, B and C) are segregating the classes well, we must choose one that maximizes the distances between nearest data point (either class) and hyper-plane. This distance is called **margin**.

Note: Hyper plane with low margin can result in mis-classification.

In **Fig. 2.38**, we see that the margin for hyper-plane C is high as compared to both A and B. Hence, hyper plane C is chosen as the right decision boundary or hyper plane.

Scenario 3: In this scenario, hyper plane A is selected as it maximizes the margin and reduces classification error.

FIGURE 2.38 Identifying Hyperplanes on the Plot for Classification

Scenario 4: In Fig. 2.38, segregation is difficult as one star lies in the territory of circle class.

Scenario-5: In the scenario, linear hyper-plane between the two classes is not possible. However, SVM can solve this solution in two ways.

Solution 1: Adding a new feature $z=x^2+y^2$ and plotting the data points on axis x and z.

In the plot,

1. all values for z would be positive always because z is the squared sum of both x and y
2. In the original plot, red circles appear close to the origin of x and y axes because of lower value of z. Stars appear away from the origin

due to higher value of z .

Solution 2: Use the kernel trick of the SVM algorithm. The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space, that is, it converts not separable problem to separable problem. It is usually used in non-linear separation problem. In the figure, we see that the hyper-plane in original input space looks like a circle.

FIGURE 2.39 Evaluating Hyperplanes

Example:

Cricket players can be classified as batsmen or bowlers using the runs-to-wicket ratio. A player with more runs is a batsman and the one with more wickets is a bowler.

In this example (refer **Fig. 2.40**), we can create a two-dimensional plot showing a clear separation between bowlers and batsmen.

FIGURE 2.40 2D Plot representing Batsmen and Bowlers Data

Whenever we get data for a new player that is not yet classified, we draw a decision boundary, or a line separating the two classes to help classify the new data points. We can draw multiple decision boundaries as shown in **Fig. 2.41**. Therefore, the aim is to find the line of best fit that clearly separates those two groups. The correct line will help to classify the new data point.

FIGURE 2.41 Identifying Line of Best Fit

The best fit line is computed by evaluating the maximum margin from equidistant support vectors. Support vectors mean two points (one from each class) that are closest together, but that maximize the distance between them or the margin.

Points to Remember

1. **Hyperplane** is a line in 2D, plane in 3D and hyper plane in higher dimensions (above 3 dimensions).
2. **Margin** is the distance between the hyperplane and the closest data point. We need to maximize this margin for better classification.
3. The kernel function is used to handle non-linear separable data. It does this by transforming data into a higher dimensional feature space to make it possible to perform the linear separation. Different kernel functions include the following:
 1. Gaussian RBF kernel
 2. Sigmoid kernel
 3. Polynomial kernel

Any of these functions can be chosen depending on the dimensions and way data has to be transformed.

Tuning Parameters

Tuning parameters can be divided into two groups—linear kernel SVM and non-linear kernel SVM.

For linear kernel SVM, there is only one parameter—cost (C) which implies misclassification cost on training data.

A large C gives you low bias and high variance; low bias, as there is a penalty (cost) for misclassification. A large C means the cost of misclassification is high. This ensures that the algorithm strictly explains the input data stricter (and potentially overfits).

A small C gives higher bias and lower variance. This is because a small value of C lowers the cost of misclassification.

Ideally, a good balance must be maintained between being ‘not too strict’ and ‘not too loose’. For finding an optimal value of C, cross-validation and resampling techniques along with grid search have proved to be excellent.

For non-linear kernel (Radial), Two parameters for fine tuning in radial kernel are cost and gamma. We have already discussed cost (C) in case of linear kernel.

Gamma explains how far the influence of a single training example reaches. A small value of gamma indicates that the model is too constrained and cannot capture the complexity or ‘shape’ of the data.

SVM algorithm is very sensitive to the choice of the kernel parameters.

2.8.1 How Does SVM Work?

Step 1: Select an optimal hyperplane that maximizes margin

Step 2: Apply a penalty or cost ‘c’ for misclassification.

Step 3: Use kernel trick on non-linearly separable data points by transforming data to high dimensional space where it is easier to classify with linear decision surfaces.

Data Standardization

All kernel methods are based on distance. Hence, values of all variables must be scaled. If we do not standardize our variables to comparable ranges, then the variable with the largest range will completely dominate the computation of the kernel function. For example, if we have two variables—X1 and X2 where values of variable X1 lies in the range 0 to 100 while that of X2 lies in range of 100 and 10000, then values of X2 will dominate variable X1. Therefore, we must standardize each variable to the range $[-1,1]$ or $[0,1]$. The z-score and min-max are the two popular methods to standardize variables.

Moreover, standardization avoids numerical problems during computation. Since kernel values depend on the inner products of feature vectors, large attribute values often result in numerical issues.

2.8.2 Advantages of SVM

1. SVM performs well with non-linear separable data using kernel trick.
2. SVM works well in high dimensional space that is, when there are large number of predictor variables.
3. SVM can be efficiently used to classify text as well as images.
4. It is free from multicollinearity problems
5. SVM is a memory efficient algorithm as only a subset of the training points is used to assign a class to a new member. Only these few points need to be stored in memory for making decisions.
6. It is a flexible algorithm as both linear and non-linear data can be classified using this technique.

2.8.3 Disadvantages of SVM

1. It takes more time when working with large sized data sets
2. SVM does not directly return probability estimates.
3. With linear separable data, linear kernel is the same as logistic regression.

2.8.4 Applications of SVM in Real World

In real world, SVM is widely used for classifying data in multiple areas as follows:

Face Detection

SVM algorithm is used to classify parts of an image as a face and non-face. You must have seen that on Facebook, whenever you post a picture, a square boundary around the face is created. This can be easily done using the SVM algorithm by detecting the face area. The algorithm creates a square boundary around faces on the basis of pixel brightness and classifies each image by using the same process.

Credit: bokan Shutterstock

FIGURE 2.42 Classifying Emotion using expressions on the face

We can even use SVM in various life-care systems to classify a face as being normal, happy or sad (**Fig. 2.42**). This technique is also used in filters. For example, whenever we make certain facial expressions, specific filter is applied for that expression.

Text and Hypertext Categorization

SVM algorithms are used to categorize and thus differentiate between text and hypertext. The algorithm also uses training data to classify documents into different categories including news articles, e-mails and web pages. Similarly, news articles can be classified into 'business' and

‘Movies’ and web pages can be categorized as personal home pages and others.

An SVM with an RBF kernel produces a ring-shaped decision boundary instead of a line.

SVMs are also used for classifying emails. An email can be classified as spam or not. Such algorithms are widely used in the email delivery systems provided by platforms like Gmail. The accuracy of such applications is as high as 97 percent accuracy for this application.

Classification of Images

SVM algorithm provides better search accuracy for image classification. It can also be used for recognizing handwriting.

Bioinformatics

SVM algorithms are used for classifying protein, cancer, genes and other biological problems. For example, protein sequences are modelled using different kernel functions to find the similarity between different protein sequences.

If data is very noisy, C should be small. If the data is less noisy, choose C to be large.

Handwriting Recognition

SVM algorithms are used to recognize handwritten characters so that they can be used for data entry applications and validating signatures on documents (refer **Fig. 2.43**).

To deal with noisy data, set C parameter to be a low value (say 1). The SVM classifier choose a large margin decision boundary at the expense of larger number of misclassifications. When C is set to a high value (say 100), the classifier will choose a low margin decision boundary and try to minimize the misclassifications.

FIGURE 2.43 Handwriting Classification using SVM

SVM is also used to differentiate between the handwritings of two different people. The dataset contains images of various alphabets and sentences written in both handwriting. SVM is also used to differentiate between human writing and computer alphabets.

Geo and Environmental Sciences

SVMs are also used for geo (spatial) and spatiotemporal environmental data analysis. Geographical and spatial data is usually uncleaned and noisy.

Generalized Predictive Control

SVM is used to control chaotic dynamics with useful parameters. It provides excellent performance in controlling chaotic systems thereby reducing waiting time and maintaining performance of such systems.

Inverse Geo-Sounding Problem

SVM algorithm is widely used to determine the layered structure of the planet. It is also used in case of electromagnetic data, where in the model is developed using linear programming techniques.

Seismic Liquefaction Potential

SVM is applied in **SPT (Standard Penetration Test)** and **CPT (Cone Penetration Test)** to check the occurrence and non-occurrence of liquefaction. Here, we use field datasets.

Mathematics

A variant of SVM- SSVM or Smooth SVM is used in mathematics for solving various math problems.

Texture Classification

SVM is also used to classify an image as smooth or rough by analysing its texture.

Speech Recognition

SVM when applied on acoustics data can be used for speech recognition applications to design an interface for communication with deaf people.

Steganography Detection in Digital Images

SVM is used to find whether a given image is plain or has some hidden message stored in it. This is usually very helpful in security-based organizations to uncover secret messages. Some people encrypt messages in high-resolution images.

AI Lab Session – Generate New Images

Visit the URL, <https://www.artbreeder.com/> The AI application allows users to keep selecting the most interesting image to discover totally new images. Therefore, Artbreeder turns the simple act of exploration into creativity by helping users make images by mixing any Artbreeder images together. Users can use any shared image for editing and mixing.

Credit: Joel Simon <https://www.artbreeder.com/>

AI Lab Session – Train a Machine Learning Model

Visit the link <https://machinelearningforkids.co.uk/> and create a project that can classify text, images and numbers. You can train the model and test how accurate the results are.

AI Lab Session – Train a Machine Learning Model

Visit the link <https://teachablemachine.withgoogle.com/> and create a project that can classify images, poses and sounds. You can train the model and test how accurate the results are.

Credit: Screenshot of TRAIN A MACHINE LEARNING MODEL, <https://teachablemachine.withgoogle.com/>, is a trademark of Google LLC

The experiment predicts the drawing which the user wants to create based on partial drawing made by the user. The user draws a sketch on the left side of the screen. On the right side, AI is used to predict the complete drawing based on user's inputs.

https://magenta.tensorflow.org/assets/sketch_rnn_demo/multi_predict.html

The experiment takes the drawing made by user as input and uses this input to make similar drawings. Visit the link https://magenta.tensorflow.org/assets/sketch_rnn_demo/index.html to draw interesting sketches. For convenience, users can select a category of

the sketch before drawing. However, do not forget to select a category and then draw entirely different sketch. For example, select cat in the category and draw a butterfly to see how the AI system behaves.

AI Lab Session – Experiments with Google

Google uses ML techniques to support authors create interesting stories with the help of three main tools that can be accessed by clicking on the URL <https://experiments.withgoogle.com/collection/aiwriting>

Machine learning models detect meaningful patterns in huge quantities of complex data. ML models can learn from larger patterns of grammar and semantics and use this learning on completely different contexts. The three tools that extensively assist users in creative writing include:

Between The Lines — The program takes the first and last line of a story and then uses machine learning techniques to interpolate between the lines and generate an interesting story specifying what would happen in the middle.

Once Upon A Lifetime— This is a character life story generator in which the user input keywords describing the kind of life they want to generate. This is especially important when writing biography of a character. The program creates the entire life story using the specified keywords.

Banter Bot— Banter bot is a character chatbot. Users can input information about what a character is like, and then can converse with it through text. As the user converses, the character evolves by learning from it.

Credit: Screenshot of EXPERIMENTS WITH GOOGLE, <https://experiments.withgoogle.com/collection/aiwriting>, is a trademark of Google LLC

AI Lab Session – AI for Music By Google

Visit the link to know about AI applications developed by Google to support music. <https://experiments.withgoogle.com/collection/ai>

Credit: Screenshot of AI FOR MUSIC BY GOOGLE, <https://experiments.withgoogle.com/collection/aiwriting>, is a trademark of Google LLC

FreddieMeter: It is an AI-powered singing challenge that rates how closely your singing matches the voice of Freddie Mercury. This application uses machine learning models to compare user's timbre, pitch, and melody with that of Freddie's. since user's audio is not uploaded to servers for analysis, so the vocal information stays private.

Semi-Conductor: This experiment by Google allows users to conduct their own orchestra in the browser just by moving their arms. Movement of the arms creates variation in the tempo, volume, and instrumentation of a piece of music.

AI Duet: It is another AI application by Google that plays a piano based on the inputs given by the user. Users can play some notes, and the computer will respond to the melody. Users may not even know how to play piano. They just have to press some keys and listen to what comes back.



KEY TERMS

Machine learning: It is a branch of computer science that analyses data and identifies patterns to teach a machine to deduce results and make decisions without any human intervention.

Deep learning: It is an ML technique that teaches a machine to process inputs through layers to more accurately classify, infer and predict the outcome.

Forward propagation: Movement of computations through the hidden layers to the output layer is known as *forward propagation*.

Backward propagation: The process of moving backward to update weights of all nodes is known as *backward propagation*.

Natural language processing: It is a science in which a machine is made to read, understand, interpret and respond to a human language.

Computer Vision: It is a branch of AI that tries to understand an image by breaking it down into several parts and then studying each part of the image.

Supervised machine learning algorithms: These are algorithms that apply learning from past data (or experiences) to new data using labelled examples.

Unsupervised learning: This is an ML technique that trains the machine using information that is neither classified nor labelled.

Clustering: It is an unsupervised ML technique that discovers inherent groupings in the data or discover hidden patterns. It finds similarity and

relationship in the underlying data.

Association analysis: It is an unsupervised ML technique that discover rules describing large portions of data.

Dimensionality reduction: It is a technique that reduces the number of feature variables for the data set. It is used when the data set has a large number (millions).

Outlier detection or anomaly detection: It is an unsupervised ML technique that is used to find out the occurrences of rare events or observations that generally do not occur.

Reinforcement learning: It is a type of dynamic programming that trains an agent over a period of time so that it can interact with a specific environment using a system of reward and punishment.

Regression analysis: It is a statistical method tool that allows users to study the relationship between a dependent (target) and one or more independent (predictor) variables.

Splitting: It is the process of dividing a node into two or more sub-nodes. In a decision tree, splitting done until a user-defined stopping-criteria is reached.

Cluster: It is a group of data objects that are similar to other objects in the same cluster and dissimilar to data objects in other clusters.

Inter-cluster distance: It is the distance between the centroids of two different clusters.

Margin: It is the distance between the hyperplane and the closest data point. We need to maximize this margin for better classification.



RECAPITULATION POINTS

- AI works by combining massive data with fast, iterative processing and intelligent algorithms, allowing the software to learn automatically deduce patterns in the data
- Cognitive computing algorithms try to mimic a human brain by analysing text, speech, images or objects in the same way to give the desired output. **It** is a subfield of AI that is used to provide a natural, human-like interaction with machines.
- A classification algorithm classifies data into a particular group but a regression algorithm predicts a real value.
- When there are multiple predictors in the regression model, then the best combination of predictors needs to be chosen to construct the optimal predictive model.
- When there is a large multivariate data set containing some correlated predictors, then such variables are summarized into few new variables that are a linear combination of the original variables. These techniques based on **principal components** include **principal component regression**.
- Logistic Regression should be used when the dependent variable is binary (dichotomous) in nature (like 0 or 1, true or false, yes or no).
- We can use **forward selection**, **backward elimination** or **bidirectional elimination** for adding/removing variables at each step. As the name suggests, forward selection continuously adds variables to the set and reviews the performance of the model. It stops when no further improvement is needed (or being achieved). Backward elimination removes variables until no extra variable is left to be deleted without considerable loss. Bidirectional elimination is a good combination of both the approaches.
- The k-nearest neighbour algorithm is a supervised learning algorithm in which the output value of data is known but how to get the output is not known
- A decision tree represents choices and their results in form of a tree. The nodes represent an event or choice and the edges of the graph represent the decision rules or conditions.

- Random forests can perform both regression and classification. They give better performance than decision trees as it does everything for reducing number of dimensions (or variables), treating missing values, outlier values and exploring data.
- While bagging chooses all predictors, Random Forest selects only one of the m predictors.
- Density-Based Clustering creates a cluster where there are high densities of data points separated by low-density regions.
- K-means is an unsupervised learning technique as it finds relationships between n observations without being trained by a response variable.
- Inertia indicates how far the points within a cluster are. It gives us the sum of intra-cluster distances. For best results (or clusters), the inertial value **should be as low as possible**
- Naive Bayes classification algorithm can be used with continuous features but performs best for categorical variables. When dealing with numeric features, the algorithm assumes that numerical values are normally distributed.
- If a given class and a feature has 0 frequency, then the conditional probability will become 0 resulting in the 'Zero Conditional Probability Problem.' This is a serious issue as it wipes out all other vital information about probabilities. However, to avoid such situations there are several sample correction techniques like the 'Laplacian Correction.'
- The output of a neuron may vary from $-\infty$ to $+\infty$. So, we need a mapping function also known as the activation function that maps inputs to the outputs.
- While Full Batch Gradient Descent Algorithm uses all the training data points to update each of the weights at once, the Stochastic Gradient uses 1 or more data points but never the entire training data for updating at once.
- Support vector machine is a supervised machine learning algorithm which can be used for both classification or regression challenges.



ANSWER THE QUESTIONS

1. Define the sub-domains of AI.
2. Define reinforcement learning and explain its types.
3. Explain the **k-fold cross-validation technique**.
4. Why is random forest better than decision tree and bagged tree?
5. Explain any three popularly used clustering techniques.
6. Give any two applications of clustering. Also discuss any two metrics that will be used evaluate the clusters.
7. What are the different tuning parameters for SVM?
8. Explain the tasks involved in computer vision.
9. How does computer vision work with deep learning?
10. Write short notes on:
 1. Reinforcement learning
 2. Regression analysis
 3. K-nn
 4. K-means
 5. Gradient descent
 6. SVM
11. Differentiate between:
 1. Forward and backward propagation
 2. Supervised and unsupervised ML
 3. Classification and regression
 4. Clustering and classification
 5. Agglomerative and divisive clustering
 6. DBSCAN and k-means
12. Consider the table given below and compute the probability of a fruit being a banana.
13. Assume that a drug test is 98% accurate and 0.5% of people use that drug. If a person selected at random tests positive for the drug, calcu-

late the probability that the person actually used that drug.

14. Consider the conditional probabilities table. Given a tweet, 'I'm happy today, I'm learning', apply Laplacian smoothing and predict the sentiment of the whole tweet.



1. In a neural network, processing is done in the _____ layer.
2. _____ creates huge neural networks with several layers of processing units.
3. _____ provide heavy computing power required for iterative processing and training neural networks.
4. A _____ algorithm predicts a real value.
5. _____ reduces the number of feature variables for the data set.
6. _____ represents the proportion of variation in the data.
7. _____ Regression should be used when the dependent variable is binary.
8. _____ regression is a mix of ridge and lasso regression.
9. In a decision tree, _____ node performs the first split.
10. In a decision tree, _____ nodes predict the outcome.
11. Density-based clustering uses a _____ parameter that acts as a threshold.
12. _____ is the ratio of the minimum inter-cluster distances and maximum of intra-cluster distances. Its value must be high for getting better clusters.
13. _____ function that maps inputs to the outputs.
14. One round of forwarding and backpropagation iteration is known as one training iteration or _____.
15. _____ algorithm is used to update the weights of the neurons.

16. _____ is the distance between the hyperplane and the closest data point.
17. The _____ function is used to handle non-linear separable data in SVM.



STATE TRUE OR FALSE

1. A neural network learns in just one pass.
2. A neural network has a single hidden layer.
3. ML algorithms learn from experiences rather than instructions.
4. Complex image and speech recognition applications are possible with machine learning.
5. Unsupervised learning should be used when output of training phase is known.
6. Semi-supervised machine learning algorithms use both labelled and unlabelled data for training.
7. Linear regression can be used with a single predictor variable only.
8. Lasso regression analysis is used for variable selection and regularization.
9. The K-nearest neighbour can be used for regression when dependent variable is continuous.
10. Decision trees map only linear relationships.
11. Each terminal (or leaf) node consists of individuals of a single class.
12. Agglomerative clustering uses bottom-up approach.
13. DBSCAN and OPTICS are popular examples of classification algorithms.
14. More the value of the Dunn index, better will be the clusters.
15. You need to specify the values of k in DBSCAN algorithm.
16. Naive Bayes classification algorithm can be used with continuous as well as categorical variables.
17. Support vector machine is an unsupervised machine learning algorithm.
18. We need to minimize margin for better classification.



MULTIPLE CHOICE QUESTIONS

1. A rule like if customer who buys X also buys product Y can be discovered using _____.
 1. Clustering
 2. Classification
 3. Association analysis
 4. Outlier detection
2. Semi-supervised learning algorithms use _____ amount of pre-labelled data and a _____ number of unlabelled data for training.
 1. Small, small
 2. small, large
 3. large, small
 4. large, large
3. Linear regression models work well with _____ predictor variables.
 1. continuous
 2. categorical
 3. both of these
 4. none of these
4. _____ regression may be used when the relationship between the outcome and the predictor variables is not linear.
 1. Linear
 2. Polynomial
 3. Logistic
 4. None of these
5. Ridge regression and Lasso regression are popular examples of _____ regression model.
 1. Linear
 2. Polynomial
 3. Penalized
 4. Spline
6. Identify the incorrect statement.
 1. Lower the RMSE, better is the model.
 2. Higher the adjusted R^2 , better is the model
 3. linear regression suffers from overfitting issues
 4. None of these

7. Identify the incorrect statement.
1. Linear regression is very useful for detecting outliers.
 2. Lasso regression variable gets adds or subtracts a variable at each step.
 3. Polynomial regression is used to analyse curvilinear form of data
 4. ElasticNet regression is usually used in SVM (Support Vector Machine Algorithm)
8. Which regression technique reduces the number of dependent variables if the penalty term is huge?
1. Linear
 2. Ridge
 3. Lasso
 4. Polynomial
9. Decision trees can perform _____.
1. Classification
 2. regression
 3. both of these
 4. none of these
10. Laplacian Correction is used in _____ algorithm.
1. Naïve Bayes
 2. DBSCAN
 3. Decision Tree
 4. k-means
11. Identify the incorrect statement.
1. Decision Tree is a supervised learning predictive model that uses rules to calculate a target value.
 2. Decision tree works for both categorical and continuous input and output variables.
 3. The tree detects local, not global, optima.
 4. Decision tree is affected by outliers.
12. Identify the incorrect statement.
1. Random forests is slow in generating predictions.
 2. Density-based clustering does not require the user to specify the number of clusters.
 3. The inertial value should be high.
 4. The cluster assignment and centroid update steps are repeatedly performed in k-means algorithm.

13. Identify the incorrect statement about k-means algorithm
 1. Number of clusters must be pre-specified.
 2. The algorithm is sensitive to outliers
 3. Changing the order of data will give different results.
 4. None of these
14. Identify the incorrect statement about neural networks.
 1. Neural network is a flexible algorithm that can be used to solve both regression and classification problems.
 2. Neural networks work with linear dataset.
 3. Neural networks can be performed only on numerical inputs
 4. They work only on non-missing value datasets.
15. Choose the algorithm which cannot be used for classification and regression.
 1. SVM
 2. Naïve Bayes
 3. Neural Network
 4. None of these
16. Which AI technique will you use to prepare ‘minutes of the meeting’?
 1. Computer Vision
 2. Cognitive Computing
 3. Chatbot
 4. Natural Language Processing (NLP)



CASE STUDY-BASED ASSIGNMENT

List at least 5 points about DialogFlow- the Google platform to design and integrate a conversational user interface with your applications.

Visit Teachable Machine <https://teachablemachine.withgoogle.com/> to learn how to train a model for speech recognition, classify images, detect snap/ clap, identify facial gestures and tilting of your head. After training the models, what can you say about the following questions?

1. Do you need to train model for a single class or for multiple classes?
2. Why should you upload multiple images or samples of voice for identification?

3. 'Adding more data and variety of data (in the current context) enhances performance of the model.' Justify this statement.
4. Did your model give accurate result when you provided a different image of the same subject? (If the model is trained to identify a dog, then test the model by providing an image of dog that is different from those that were used to train the model).



ENHANCING AUDIO, VISUAL AND COGNITIVE SKILLS

Visit the link given below to understand how an AI tool applying Computer Vision techniques can be used for cleaning the garbage from an ocean.

<https://studio.code.org/s/oceans/lessons/1/levels/2>

Now answer the questions given below.

1. Does 'computer vision' act as the eye of a machine/robot?
2. Why did we supply so many examples of fish before the AI model started recognizing the fish?
3. Can a regular camera see things?
4. What features you would like to add to the camera to make it true smart surveillance camera?

Visit the video at the link https://www.youtube.com/watch?v=JZp7t_DPhWI and answer the following questions.

Can you name any three factors that help us to forecast weather?

What is the role of satellites in the process of weather forecasting?

How does a computer get images of weather conditions in a far-off place?

Can AI help to forecast weather conditions in a particular area?

Visit the video at the following link <https://www.youtube.com/watch?v=B8R148hFxPw> and answer the questions given below.

What should the self-driving car do if it sees an animal coming in from it?

In the age of self-driving cars, do we need zebra crossings?



ANSWERS

Fill Up

1. Hidden
2. Deep learning
3. Graphical processing units
4. Regression
5. *Dimensionality reduction*
6. Adjusted R-square
7. Logistic
8. Elastic
9. Root
10. Terminal
11. distance-based
12. Dunn index
13. Activation
14. Epoch
15. gradient descent
16. Margin
17. kernel

State True or False

1. False
2. False
3. True
4. False
5. False
6. True
7. False
8. True
9. True

10. False
11. True
12. False
13. False
14. True
15. False
16. True
17. False
18. False

Multiple Choice Questions

1. c
2. b
3. c
4. b
5. c
6. d
7. b
8. c
9. c
10. a
11. d
12. c
13. d
14. b
15. d
16. d

Previous chapter

< [1. Introduction to Artificial Intelligence](#)

Next chapter

[3. Artificially Intelligent Machine](#) >

Table of contents expanded

3

Artificially Intelligent Machine

3.1 Defining Intelligence

Howard Gardner, the American developmental psychologist, has categorized intelligence into the following categories.

Linguistic intelligence is the ability to speak, recognize and use mechanisms of phonology (speech sounds), syntax (grammar) and semantics (meaning). It is mainly exhibited by narrators and orators.

Musical intelligence is the ability to create, communicate with and understand different sounds, pitch and rhythm. This intelligence is used by musicians, singers and composers.

Logical-mathematical intelligence helps a person to use and understand complex and abstract ideas. Mathematicians and scientists extensively use this type of intelligence.

Spatial intelligence is the ability to perceive visual or spatial information, change it and re-create visual images. Using this, 3D images can be created and manipulated (translated, transformed and rotated). This intelligence is generally applied by map readers, astronauts and physicists.

Bodily-kinesthetic intelligence defines the ability to use the entire or part of the body to solve problems or manipulate objects. For example, this kind of intelligence is mostly used by players and dancers.

Intrapersonal intelligence is the ability to distinguish between one's feelings, intentions and motivations. A few people like Gautam Buddha has this instinct.

Interpersonal intelligence is the ability to recognize and differentiate between other people's feelings, beliefs and intentions. Mass communicators and interviewers use these skills to the maximum.

A machine or a system is said to be **artificially intelligent** when it exhibits at least one and at most all intelligences.

3.2 Components of Intelligence

Intelligence is an intangible entity composed of the following (**Fig. 3.1**):

- Reasoning
- Learning
- Problem solving
- Perception
- Linguistic intelligence

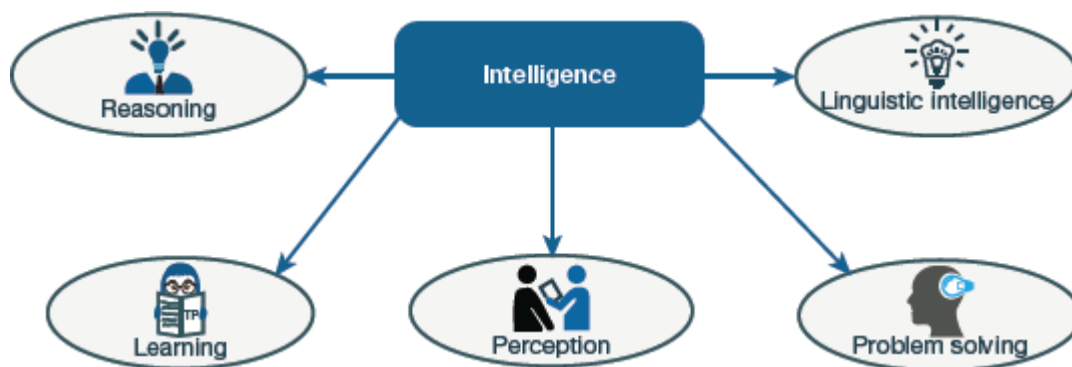


FIGURE 3.1 Components of Intelligence

Reasoning

It is the set of processes that is used in making decisions and prediction. There are two types of reasoning—inductive and deductive. The differences between these two techniques is given in **Table 3.1**.

TABLE 3.1 Inductive vs Deductive Reasoning

Inductive Reasoning	Deductive Reasoning
It uses specific observations to make broad general statements.	It starts with a general statement to obtain a specific, logical conclusion.
Even if all observations are true, there are still chances that the conclusion can be false.	If something is true for a particular class of things in general, it is also true for all members of that class.
Example: ‘Manvi is a topper. Manvi is studious. Therefore, all toppers are studious.’	Example: ‘Students who scored at least 80% are eligible for placement. Manan scored 90%. Therefore, Manan is an eligible candidate.’

Learning

It is a process of gaining knowledge or skill by studying, practicing, being taught or experiencing something. Learning is an ability that is possessed by humans, animals and AI-enabled systems to improve the awareness of the subjects of the study.

Learning can be categorized as follows.

1. **Auditory learning** promotes learning by listening and hearing. For example, students hear recorded lectures to understand a particular concept.
2. **Episodic learning** promotes learning by remembering the sequences of events that one has witnessed or experienced. This is linear and orderly.
3. **Motor learning** helps to learn by precise movement of muscles. For example, picking objects.
4. **Observational learning** triggers learning by watching and imitating others. For example, children often learn by mimicking their parents.
5. **Perceptual learning** is done by recognizing stimuli that one has seen before. For example, identifying and classifying objects and situations.
6. **Relational learning** involves learning to differentiate among various stimuli on the basis of relational properties, rather than absolute properties. For example, if we know that the last time we cooked a dish, it had excess spices, the next time, we would add spices in lesser quantity.
7. **Spatial learning** is done through visual stimuli such as images, colours, maps, etc. For example, we usually create a roadmap in our

minds before actually moving to the road.

8. **Stimulus-response learning** facilitates subjects to perform a particular behaviour when a certain stimulus is received. For example, we usually shout when we touch a hot vessel.

Problem Solving

It is the process in which one tries to find the desired solution in the present situation by following the path which is blocked by known or unknown hurdles. Problem solving uses **decision-making** to select the best suitable alternative to reach the desired goal.

Perception

It is the process of acquiring, interpreting, selecting and organizing sensory information. While humans use sensory organs to perceive their environment, AI systems use data acquired by the sensors to do the same.

Linguistic Intelligence

It is used in interpersonal communication and defines one's ability to use, comprehend, speak and write the verbal and written language.

3.3 Differences Between Human and Machine Intelligence

1. Humans perceive by patterns, whereas machines perceive by analysing data with respect to a given set of rules.
2. Humans store and recall information by patterns but machines use searching algorithms to do the same.
3. Humans have intelligence to deduce any information that is missing or distorted but machines lack this ability to do it with a high level of accuracy.

3.4 Agent and Environment

Artificial intelligence is all about studying rational agents, which may include anything that makes decisions. For example, a person, firm, machine or software that performs action(s) with the best outcome after considering past and current percepts at a given instance are examples of

agents. These agents work in their environment and may interact with other agents in that environment. Thus, we can say that an AI system comprises an **agent and its environment**.

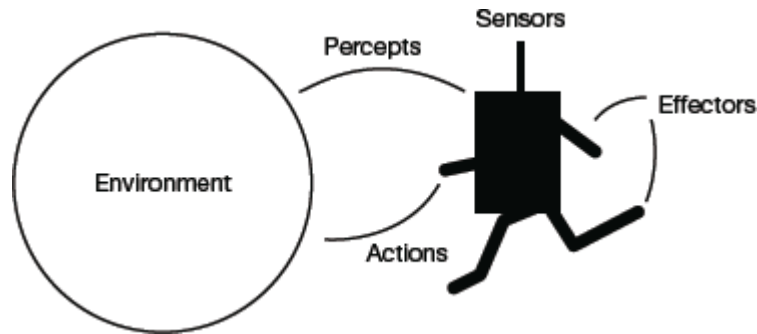


FIGURE 3.2 AI Agent

In an artificially intelligent system, the agents act in their environment which may contain other agents. This **agent** (refer **Fig. 3.2**) can be anything that is capable of perceiving its environment. For this, it may have **sensors**. **Agents also have effectors** that help them to act upon their environment. *The different types of agents in an AI system include the following:*

1. **Human agent:** A human agent has sensory organs (like eyes, ears, nose, tongue and skin) that acts as sensors, and other organs such as hands, legs, mouth as effectors for taking actions.
2. **Robotic agent:** A robotic agent uses cameras and infrared range finders as sensors, and various motors and actuators as effectors.
3. **Software agent:** Such an agent uses bit strings as its programs and actions.

3.4.1 Key Terminology

Performance Measure of Agent

This criteria helps to determine how successful an agent is.

Behaviour of Agent

It is an action performed by an agent after any receiving a certain percept.

Percept

These are perceptual inputs given to an agent at a specific instance.

Percept Sequence

This is a list of all percepts received by an agent till date.

Agent Function

This is a map from the precept sequence to an action.

3.4.2 Rationality

Rationality is a feature that inculcates a sense of responsibility, sensibility and judgment. It empowers the agent to perform expected actions after perceiving. The agent must perform all the actions to maximize its performance measure with respect to the percept sequence and the knowledge base that it has. Therefore, we can say that rationality of an agent depends on the following:

1. **Agent's performance** measure that gives the degree of success.
2. **Agent's percept sequence** received so far.
3. Agent's **prior knowledge** about the environment.
4. Agent's **actions** that can be performed.

A rational agent (refer **Fig. 3.3**) always performs the right action in the given percept sequence to maximize performance. Thus, a problem solved by an agent is characterized by performance measure, environment, actuators, and sensors (PEAS).



FIGURE 3.3 Rational Agent

3.4.3 Structure of Intelligent Agents

The structure of an intelligent agent can be given as,

$$\text{Agent} = \text{Architecture} + \text{Agent Program}$$

where architecture is the machinery on which an agent works, and agent program is an implementation of an agent function.

3.4.4 Types of Agents

In this section, we will discuss different types of agents.

Simple Reflex Agents

Simple reflex agents choose actions based only on the current percept. They are rational only if they make a correct decision based on the current precept. Their environment is completely observable.

They work using **condition-action rule** that maps a state (condition) to an action. If the condition is true, then the action is taken, else not. However, the problem with a simple reflex agent is that they succeed only when the environment is fully observable. If the environment is partially observable, then the agent may get stuck in infinite loops. In such a situation, the agent can escape the loop only if it can randomize its actions. Other issues with such agents are given below.

1. Possess very limited intelligence
2. Has knowledge of any state other than the current state.
3. Any change in the environment will eventually end up updating the collection of rules.

Model-Based Reflex Agents

Model reflex agents (refer **Fig. 3.4**) use a model of the world to choose their actions. For this, they need to maintain an internal state. Here, **model** means knowledge about 'how things happen in the world'.

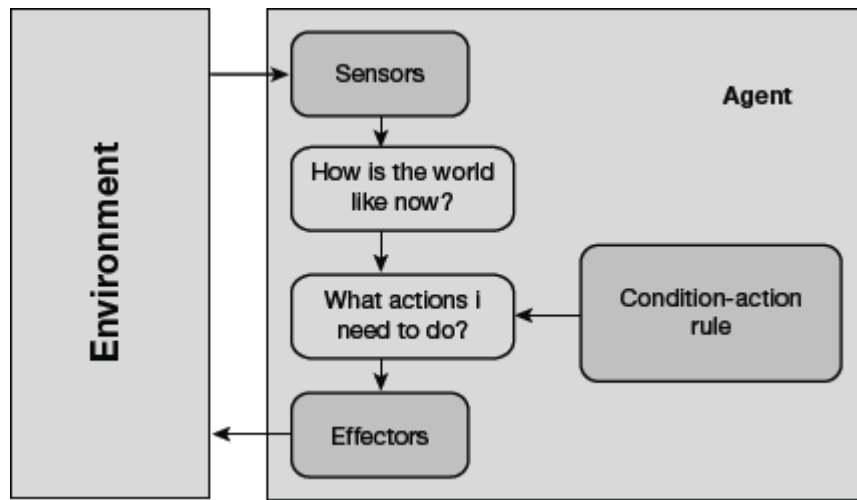


FIGURE 3.4 Model reflex agent

Internal state represents unobserved aspects of the current state depending on percept history.

To update the state, information is required to understand,

- How the world evolves and
- How the agent's actions affect the world.

Goal-Based Agents

Goal means a description of desirable situations. Goal-based agents (refer **Fig. 3.5**) choose their actions to achieve goals. This provides more flexibility than reflex agent as the knowledge supporting a decision is explicitly modeled and permits any sort of modifications.

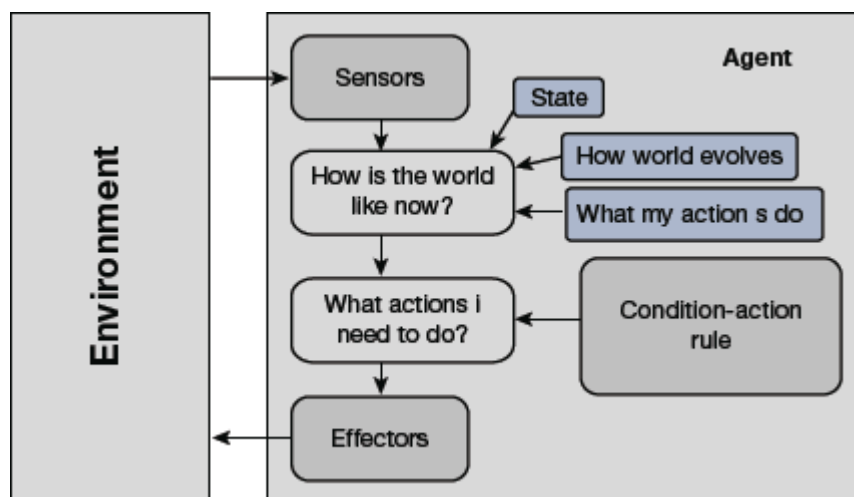


FIGURE 3.5 Goal-based agent

Utility-Based Agents

Many a time, goals are either conflicting or are difficult to achieve. In such situations, utility-based agents (refer **Fig. 3.6**) can be used to achieve goals that are more important than others. These agents choose actions based on a preference (utility) for each state.

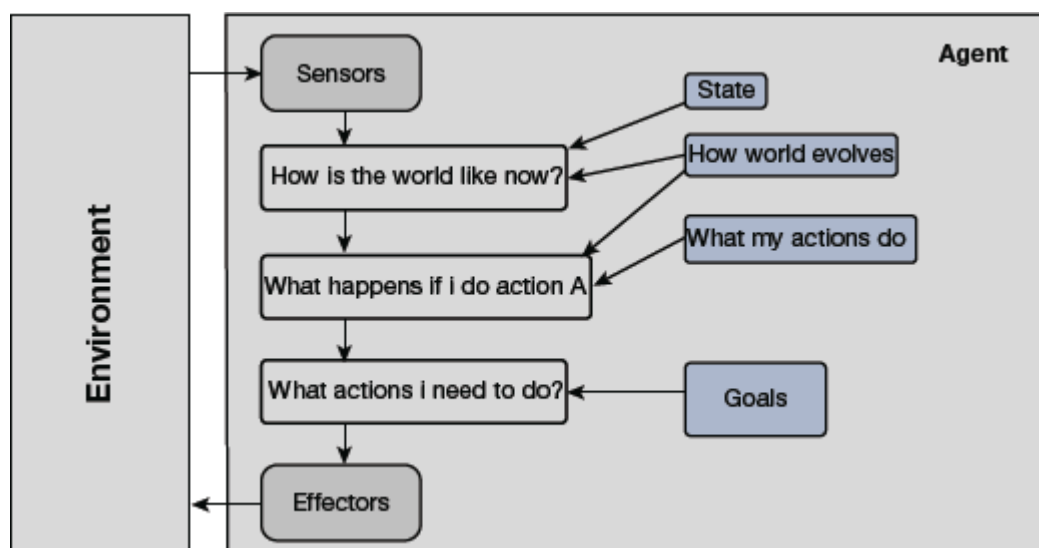


FIGURE 3.6 Utility-based agents

However, at times, achieving the desired goal is not enough and happiness of the agent is also important. In such a scenario, utility describes how **'happy'** the agent is. A utility agent chooses the action that maximizes the expected utility, that is, the associated degree of happiness.

Learning Agent

A learning agent (refer **Fig. 3.7**) learns from its past experiences. It is designed to possess learning capabilities. Initially, it starts working with basic knowledge and then acts and adapts automatically through learning.

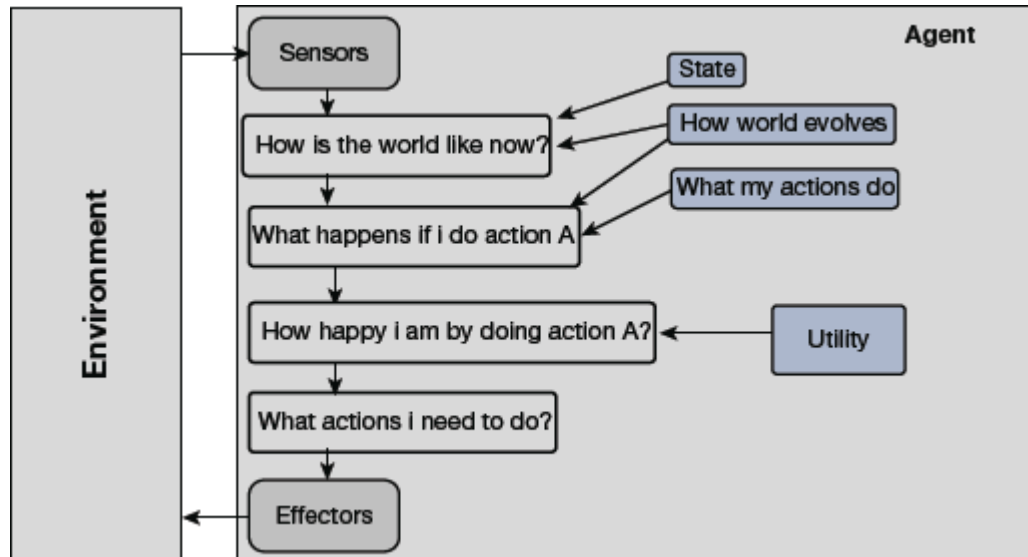


FIGURE 3.7 Learning agent

Such an agent has four main conceptual components, as follows:

Learning element, which is responsible for making improvements by learning from the environment

Critic (refer **Fig. 3.8**) is the element that provides feedback to the learning element, describing how well the agent is doing with respect to a fixed performance standard.

Performance element chooses a particular external action to be taken.

Problem generator suggests actions that result in new and informative experiences.

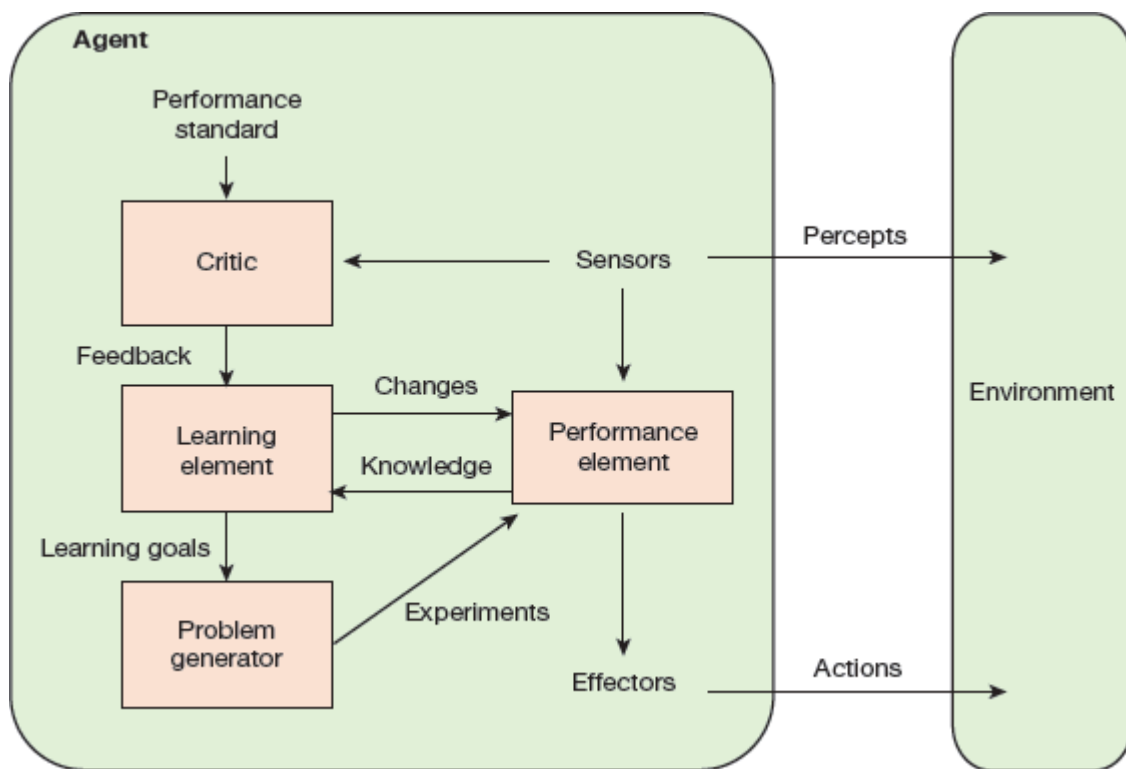


FIGURE 3.8 Critic

3.4.5 The Nature of Environments

When building an artificially intelligent system, some programs would be confined to keyboard input, database, computer file systems and character output on a screen while other software agents (like software robots or softbots) would operate in unlimited domains. The simulator for a softbot has a **very detailed, complex environment**. The software agent must take a specific action by choosing from a long array of actions in real time.

For example, a softbot whose job is to scan customer preferences and display them relevant items works in the **real** as well as an **artificial** environment.

The most famous **artificial environment** is the **Turing Test environment**, in which artificial agents are tested with a real agent (on equal ground). The software agent is trained to perform as well as a human.

Turing Test

Turing test is widely used to determine the success of an intelligent behaviour of a system. Two persons and a machine or intelligent system or software agent to be evaluated participate in the test. One of the two persons becomes the tester. The other person and the software agent are then made to sit in different rooms. The tester does not know who is the machine and who is a human. He asks them questions by typing and sending them to both, and receives typed responses.

The aim of this test is to fool the tester. If the tester fails to determine whether the reply is coming from a machine or a human, then the machine is said to be intelligent.

3.4.6 Types of Environments

As stated earlier, environment is the place where the agent would work. It gives rewards and specifies states and actions of the agents. Although environment in an AI system is quite vast, we can identify a small number of dimensions to categorize them as given below.

Discrete/Continuous

If an environment has a limited number of distinct, clearly defined, states, then it is said to be discrete. For example, a software agent designed to play chess game works in a discrete environment as there are limited moves that a player can make. However, a self-driving car operates in a continuous environment.

While a discrete environment has a finite number of percepts and actions that can be performed within it, a continuous environment has no such constraints.

Known Vs Unknown

Known and unknown are based on an agent's state of knowledge to perform an action and they are not actually a feature of an environment. In a known environment, the agent knows the results for all actions but in an unknown environment, the agent has to learn how it should work to

perform an action. Usually, a known environment is partially observable and an unknown environment is fully observable.

While known environments are good examples of **exploitation**, **unknown environments are best for exploration**. **Reinforcement learning** is extensively applied in an unknown environment.

Observable/Partially Observable

If an agent can determine the complete state of the environment at each time point from the percepts, then it is said to be observable; otherwise, it is only partially observable. Correspondingly, in an unobservable environment, the agent has no sensors. In a fully observable environment, the complete state of the environment *relevant to the choice of action* of the agent can be captured using its sensors.

While a fully observable environment (refer **Fig. 3.9**) does not require to maintain the internal state to keep track history of the world, a not fully observable environment, on the other hand, must maintain an internal state to keep track of the world.

An environment with sensors can be partially observable because of noise, inaccuracy of the sensors, due to the framework of the task itself or because parts of the state are missing from the sensor data.

For example, the classic chess game is fully observable as one agent can perceive the positions and the moves of the other agent. But in the **Kriegspiel** version of chess, the game environment is partially observable as only invalid moves can be observed.

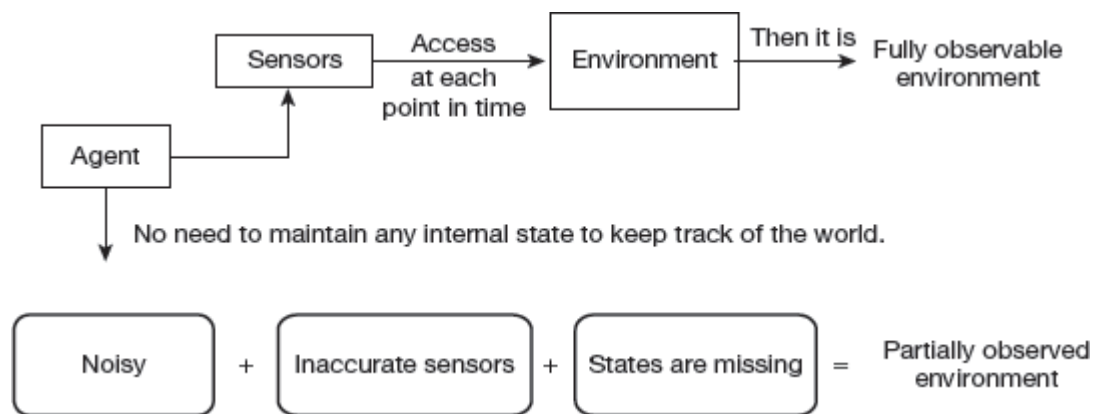


FIGURE 3.9 Observable/Partially Observable Environment

Static/Dynamic

A static environment does not change when an agent is acting, but a dynamic environment does. Agents in static environments, therefore, do not take into account the state of the environment during the action and are thus easy to deal with. However, in a dynamic environment, agents must consider the world while performing each action. For example, self-driving cars work in a dynamic environment, but games like crossword puzzles and chess operate in a static one.

Between these two extremes, we can also have **semi-dynamic environments** which do not change with the passage of time but the agent's performance score does.

Single Agent/Multiple Agents

The environment may contain other agents which may or may not be of the same kind of the agent. For example, vacuum cleaning environment is a single agent but a chess game is a two-agent environment.

In a multi-agent environment, the performance measure of one agent depends on the action performed by another agent. A multi-agent environment (refer **Fig. 3.10**) may be competitive or cooperative. For example, chess is an example of a **competitive** environment and two taxi-driving agents must avoid collisions and work to maximize the performance of both the agents. Thus, it operates in a **cooperative** environment.

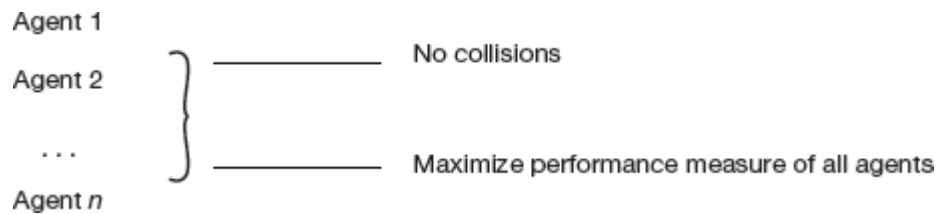


FIGURE 3.10 Multi-agent environment

Accessible/Inaccessible

In an accessible environment, the sensory apparatus of the agent has access to the complete state of the environment. This means that it can obtain complete and accurate information about the state's environment, but this is not so in case of an inaccessible environment.

For example, an empty room whose state can be defined by its temperature is an example of an accessible environment, whereas information about an event on the Earth is an inaccessible environment.

Deterministic/Non-deterministic

In a deterministic environment (refer **Fig. 3.11**), the next state of the environment can be easily determined by the current state as well as the actions of the agent. This is not true for a non-deterministic environment. A **non-deterministic** environment description maximizes an agent's performance for all possible outcome of its actions. Note that in a deterministic, fully observable environment, the agent does worry about uncertainty.

In a **Stochastic** environment, uncertainty about outcomes is quantified in terms of **probabilities**.

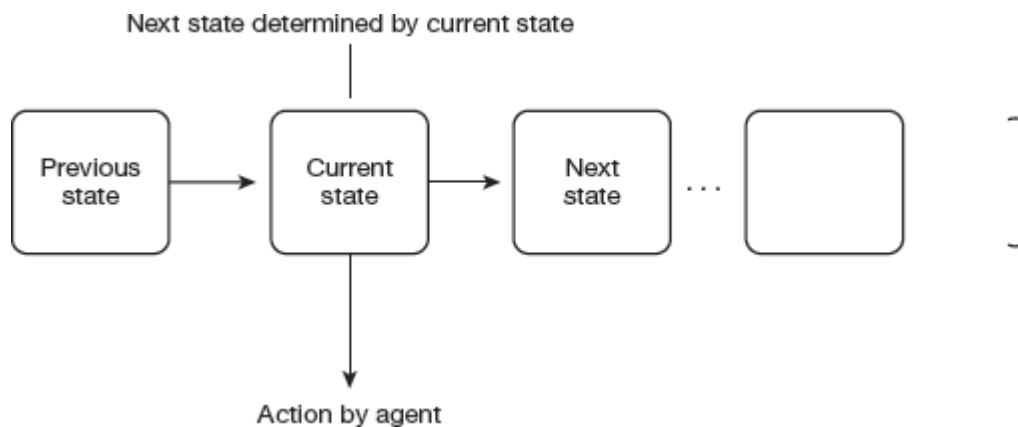


FIGURE 3.11 Deterministic/Non-deterministic Environment

For example, the game of ludo is non-deterministic as the dice will generate a number randomly, creating uncertainty in the environment. On the contrary, chess can be considered a deterministic environment to a certain extent as the state of the game can be determined by estimating the other agent's moves although there can be uncertainty due to the other agent in the game. Most real-life situations are so complex that it is not possible to track of all the unobserved aspects. In such cases, we can consider them as deterministic.

Episodic/Non-episodic

In an episodic environment (refer **Fig. 3.12**), each episode has an agent perceiving and then acting. The quality of its action depends just on the episode itself. Subsequent episodes do not depend on the actions performed in the previous episodes.

Episodic environments are, therefore, simpler as the agent does not need to think ahead. There are however, a series of one-shot actions, and only the current percept is required for the action. On the other hand, in a sequential or non-episodic environment, an agent requires memory to store past actions so that it can determine the next best actions. Therefore, the current decision can affect all future decisions.

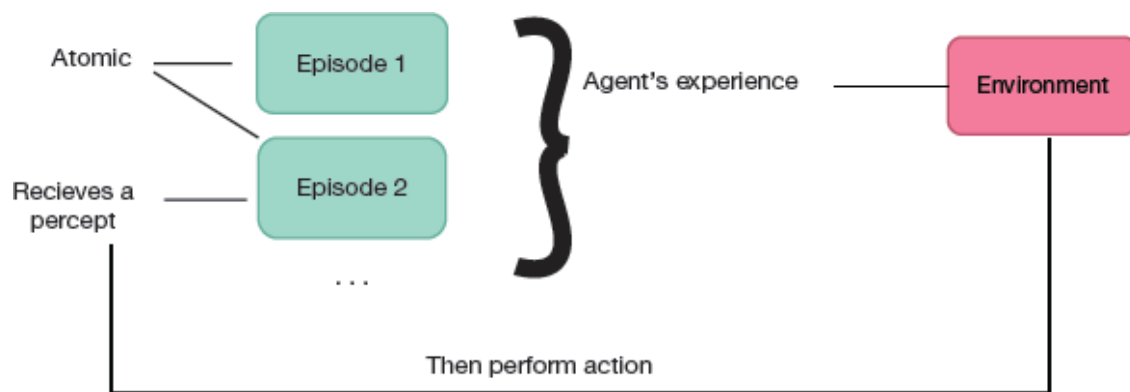


FIGURE 3.12 Episodic/Non-episodic Environment

3.5 Search

Algorithms

We have read that artificial Intelligence is the study of building rational agents. These agents make use of some or the other search algorithms in the background to achieve their tasks. For example, some single-player games like tile games, Sudoku, crossword, etc., use search algorithms to deduce a particular position in the game.

A search problem consists of the following:

1. **State space** is the set of all possible states that can be attained by an agent.
2. **Start state** is the state from where the searching is done.
3. **Goal test** is a function that checks if the current state is the goal state or not.
4. **Solution** to a search problem is a sequence of actions (also known as **plan**) that transforms the start state to the goal state. This plan is realized using search algorithms.

3.5.1 Types of Search Algorithms

There are many search algorithms but, in this section, we will discuss six of the most popularly used ones. For better understanding, we will categorize them under two headings—informed and uninformed as shown in **Fig. 3.13**.

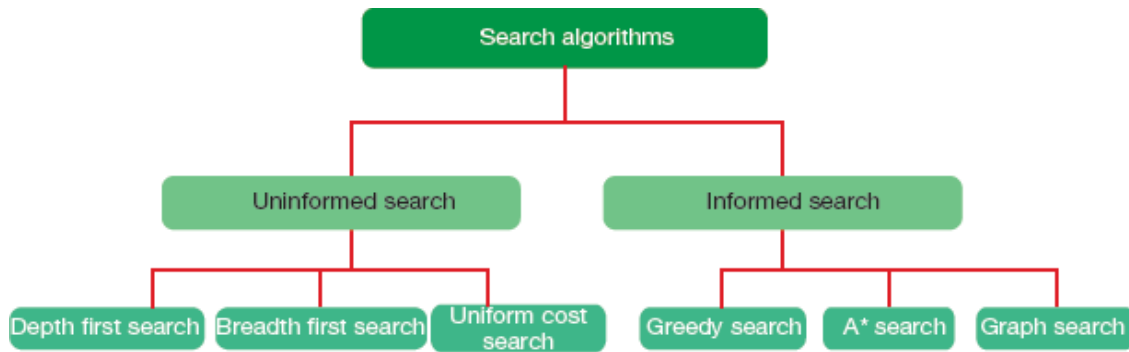


FIGURE 3.13 Different Types of Search Algorithms

3.5.2 Properties of Search Algorithms

1. **Completeness:** A search algorithm is said to be complete if it returns at least one solution for a particular input.
2. **Optimality:** A search algorithm is said to give an optimal (best) solution if it has the lowest path cost.
3. **Time and space complexity:** Time complexity is the time taken by an algorithm to complete a given task, and space complexity is the maximum storage space required to perform the search operation. A good search algorithm takes less time and space to do its work.

3.6 Uninformed Search Algorithms

A uninformed search, also known as a blind search algorithm, has no additional information about the goal state other than the one provided in the problem definition. The only information they have is on how to traverse or visit the nodes in the tree.

In such algorithms, the machine blindly follows the technique irrespective of whether it is right or wrong, efficient or inefficient. There can be multiple plans to reach the goal state from the start state. All these paths differ by the order and/or length of actions.

In this section, we will read about depth first search, breadth first search and uniform cost search algorithms. Each of these algorithms has the following information.

Problem graph, the start node S to the goal node G.

Strategy, stating the path followed by the graph reach G.

Fringe, which is a data structure to store all the possible states (nodes) that can be reached from the current state.

Tree, that depicts path while traversing to the goal node.

Solution plan, that specifies the sequence of nodes from S to G.

Path/Step cost are integers that represent the cost to move from one node to another node.

3.6.1 Depth First Search (DFS)

In this search algorithm, the tree is traversed from the root node. The key will be searched till the leaf node of a particular branch. If the key is not found, then the search process backtracks to the point from where the other branch was left unexplored. This process is repeated for that other branch either until the key is found or the entire tree is completely traversed.

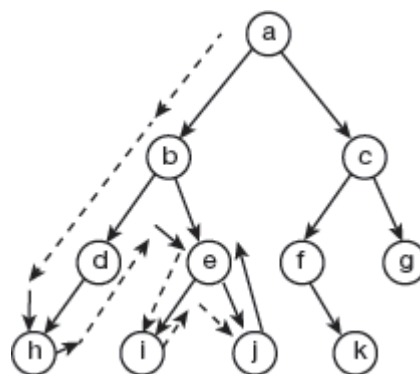


FIGURE 3.14 Depth-first search

In **Fig. 3.14**, we can clearly visualize the working of a DFS algorithm. Here, searching starts from the root node A and then traverses nodes B, D and H. Now, H is the leaf node, so we retrace the path to reach node B to traverse its unexplored branch. Nodes E and I are thus traversed. Now, I is the leaf node, so again, the path is retraced to follow the unexplored branch of node E. As a result, node J is visited and then it is found that all nodes in branches of node B have been traced. So, the algorithm will

move further by exploring the other untraced branch of the root node. Here, node C, followed by nodes F, K and G are explored.

From these observations, we can conclude that DFS algorithm occupies a lot of memory space, and takes a lot of time to execute when the solution is at the bottom or end of the tree. DFS is implemented using LIFO, that is, stack data structure.

Time complexity of the algorithm can be given as, $O(b^m)$. It is *equivalent to the number of nodes traversed in DFS*. Correspondingly, space complexity is *equivalent to how large can the fringe get*. It is given as $O(bm)$, where

b is the maximum branching factor in a tree.

d defines the depth of the least-cost solution.

m specifies the maximum depth of the state space. It maybe infinity.

Note that path = the depth of the search tree = the number of levels of the search tree = number of nodes in level .

DFS algorithm explores as far as possible along each branch before backtracking.

Though the DFS algorithm is complete if the search tree is finite and a solution exists, **it** is not optimal as the number of steps in reaching the solution, or the cost spent in reaching it is high.

Advantages

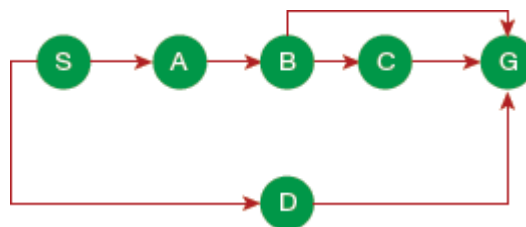
1. It uses less memory as it stores only the nodes on the path from the root node to the current node.
2. It takes less time to reach the goal node/state as compared to the BFS algorithm.

Disadvantages

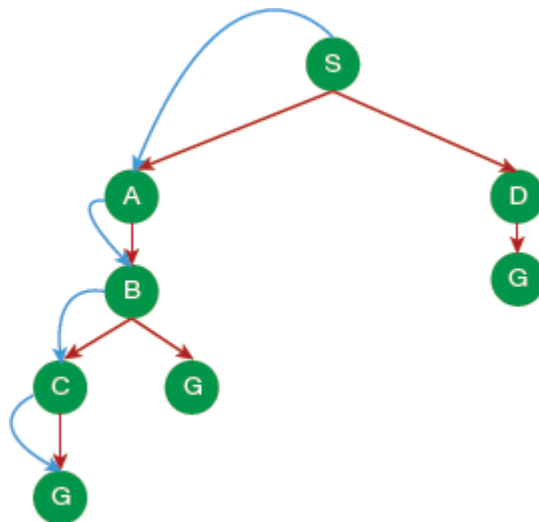
1. Sometimes, many states keep recurring. In such a scenario, there is no guarantee of finding the solution.
2. In certain cases, the algorithm may get stuck in an infinite loop while going deep searching for nodes in the tree. A possible solution to this problem is to choose an appropriate value of cut-off depth. If the ideal cut-off is d , and the chosen cut-off is less than d , then algorithm fails. But, if the chosen cut-off is more than d , the execution time increases.
3. Complexity of the algorithm depends on the number of paths.
4. DFS cannot check duplicate nodes.

Example: Explore the path that will be explored using the DFS algorithm to reach node G from S.

DFS creates the same set of nodes as breadth-first method, different order.

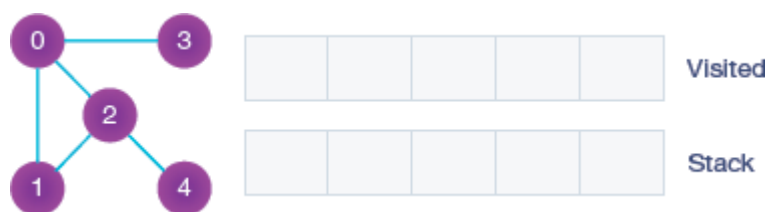


Solution: To explore the path, a search tree for the graph will be created. Since DFS traverses the tree using the ‘deepest node first’ technique, it would always pick the deeper branch until it reaches the solution or until all nodes of the current branch have been traversed. The traversal is shown in blue arrows.



The DFS path can be given as, S -> A -> B -> C -> G

Example: Consider the graph given below and state its DFS traversal.



Solution: We start with node 0. Exploring its branch to node 1, we move to node 2 and node 4. From node 4, we backtrack to node 3. Hence, the DFS path can be given as, 0 -> 1 -> 2 -> 4 -> 3.

3.6.2 Depth-Limited Search Algorithm (DLS)

DLS is similar to the DFS algorithm with an addition of a predetermined limit. This limit helps to overcome the limitations of the infinite path in the DFS. In DLS, the node at the depth limit will be treated as it is a leaf node (or a node with no successor nodes).

The DLS algorithm is terminated in any of the three cases:

1. First, if the problem does not have any solution. This is known as standard error failure.
2. Second, if the problem does not have any solution within a given depth limit. This is known as cut-off failure.
3. Third, when the solution is found.

For example, in the tree given in **Fig. 3.15**, if level is limited to 2, then the search technique will not go to level 3. Therefore, nodes E, F, G and H will not be traversed.

Thus, the DLS search algorithm is complete if the solution is present within the specified depth limit. While time complexity of DLS algorithm is $O(b^l)$, space complexity is $O(b \times l)$, where l is the depth limit and d is the depth of the tree.

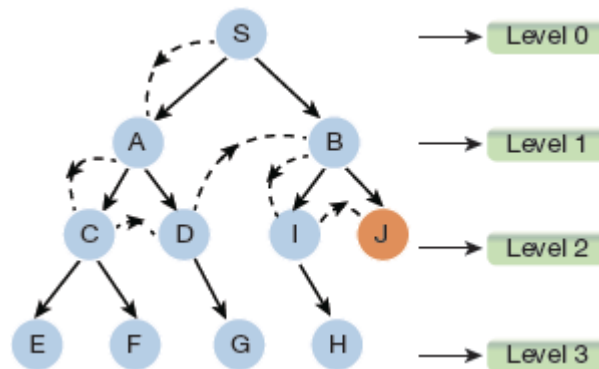


FIGURE 3.15 Depth limited search

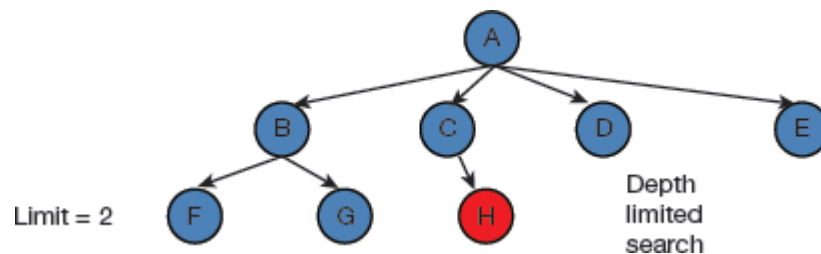
Advantages

1. It is a memory efficient algorithm as it consumes less space.
2. The algorithm takes less time to execute.
3. It terminates in finite time.

Disadvantages

1. It is an incomplete algorithm as we may not be able to get the solution every time we do the search (even if the solution exists, due to limit constraint).
2. If multiple solutions of a problem exist, then DLS may not be able to find the optimal solution. In other words, the algorithm is not optimal even if $\ell > d$.

Example: Traverse the given tree to search node H using DLS with predefined limit as 2.



Solution: We start with Node A at level 0.

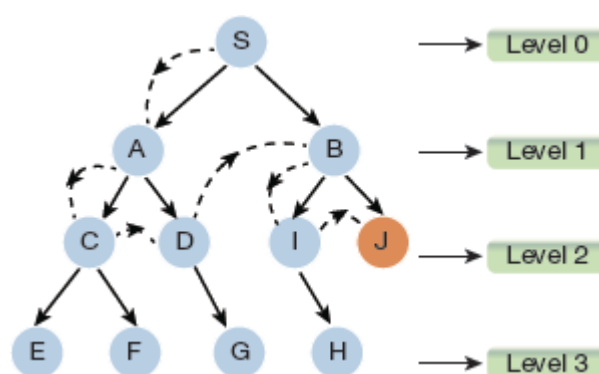
The search process then continues to explore nodes B, C, D, and E at level 1.

Now, we go to level 2 and explore the child nodes of node B. When H is not found there, a backtrack is done to level 1.

Child nodes of node C are searched to find node H. Finally, node H is found at level 2 as a child node of C and the algorithm terminates.

Example: Traverse the given tree to search node H using DLS with predefined limit as 2.

Solution: In the tree, node H is not present till level 2, so the algorithm terminates returning a cutoff failure. The path traversed for searching is marked with a dashed line.



This image explains the DLS implementation and could be referred to for better understanding.

3.6.3 Breadth First Search (BFS)

In this algorithm, nodes in the tree are traversed breadthwise to reach the goal node. Searching begins from the root node and expands the suc-

cessor node breadthwise before traversing depth-wise.

As evident from **Fig. 3.16**, a BFS algorithm starts from the root node A and then traverses node B followed by node C. This means that children of the root node are traversed before exploring other nodes on any branch. After traversing all nodes at a particular level, control then passes to the next level. Therefore, after node C, all nodes from D to G are traversed. This is followed by traversal of nodes H to K. Thus, the BFS algorithm explores all neighbour nodes at the present depth before moving the nodes at the next depth level.

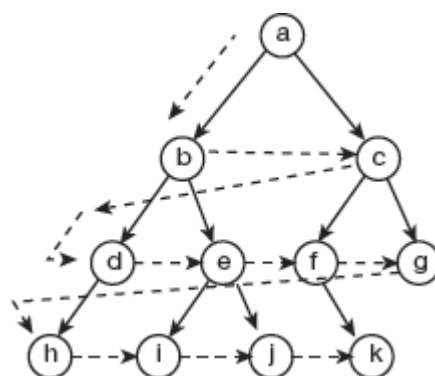


FIGURE 3.16 Breadth-first search

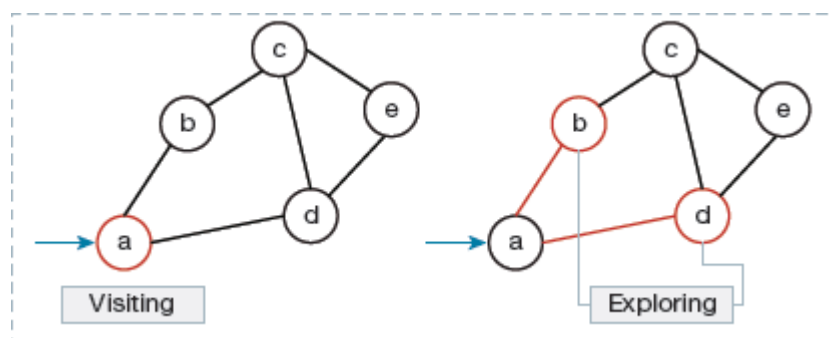


FIGURE 3.17 BFS Tree starting from a random node

In **Fig. 3.17**, BFS algorithm selects a random initial node (also known as source node or root node) and traverses the graph layer-wise in such a way that all the nodes and their respective children nodes are visited and explored. Here,

Visiting a node means to visit or select a node. And,

Exploring a node means exploring the adjacent nodes (child nodes) of the selected node.

Advantages

1. BFS will find a solution if it exists.
2. If a given problem has multiple solutions, then BFS will give the minimal solution requiring least number of steps.
3. The architecture of the BFS algorithm is simple and robust.
4. It constructs the shortest path of traversing through the nodes of the graph so that the graph can be traversed in the smallest number of iterations.
5. The algorithm does not get stuck in an infinite loop.

Disadvantages

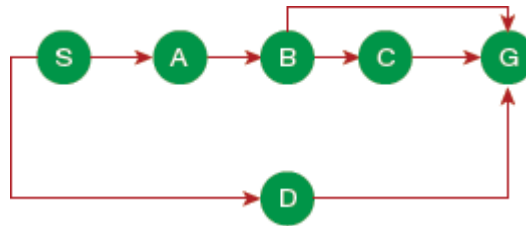
1. Since each level of node is saved for creating the next one, it consumes a lot of memory space. Space requirement to store nodes is exponential.
2. It takes less time if the solution is far away from the root node—at the bottom or at the end.
3. Its complexity depends on the number of nodes.

The BFS algorithm is implemented using a FIFO queue. It is a complete algorithm, that is, it finds a solution if it exists. The algorithm is optimal, if step cost = 1 (that is, if either there is no cost or if all step costs are same). Moreover, it has both time and space complexity given by, $O(b^d)$. Here, b is the branching factor and d is the depth of the tree.

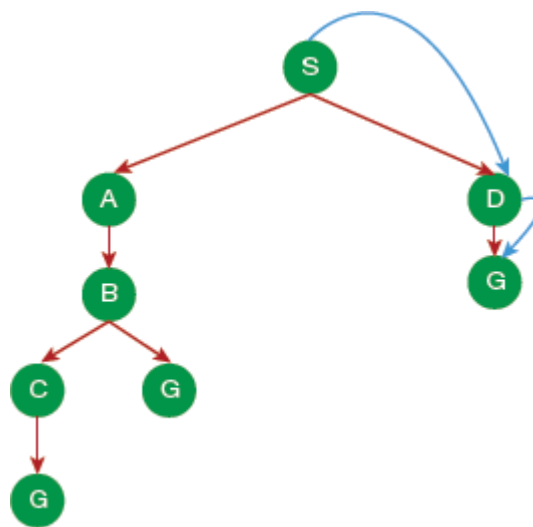
Note that total number of nodes created in the worst case is $b + b^2 + b^3 + \dots + b^d$.

While time complexity is equivalent to the number of nodes traversed in BFS space complexity, on the other hand, it is equivalent to how large can the fringe get. Due to high precision and robust implementation, BFS is used in multiple real-life solutions like P2P networks, Web Crawlers and Network Broadcasting.

Example: Find the BFS traversal from node S to node G.



Solution: The BFS creates a tree and traverses it using the principle ‘shallowest node first’. So, at node S, node D will be traversed followed by node G. Thus, **Path is S -> D -> G**



*In a BFS algorithm, nodes in the path = the depth of the shallowest solution
= number of nodes in level .*

Example: Consider the graph. Using node A as the source node, traverse the graph and trace the working of the algorithm using a queue.

According to the pseudocode, s is the root node of the graph, G . initially, s is inserted in the queue. Then, all child nodes of ' s ' are marked. These child nodes are visited after removing ' s ' from the queue. At each step of the algorithm, child nodes, w are inserted into the queue to further visit its child nodes. The process is repeated under the queue is empty.

Applications of Breadth-First Search Algorithm

BFS is a simple graph traversal method that is widely used in the following areas.

Crawlers in Search Engines

BFS algorithm is used for indexing web pages. In this application, each web page is considered as a node in a graph. The algorithm starts traversing from the source page and follows all the links associated with that page.

GPS Navigation Systems

GPS systems use BFS algorithm to find neighbouring locations.

Find the Shortest Path for an Unweighted Graph

For an unweighted graph, the shortest path can be easily calculated by choosing a path with the least number of edges. BFS algorithm can do this by traversing a minimum number of nodes starting from the source node. Similarly, either breadth-first search or depth-first traversal methods can be used to find a spanning tree.

Broadcasting

In computer networks, data is broken down into small packets before transmission across the communication media. These packets use an algorithm to compute a path to the destination. For messages that are broadcasted across all the nodes in a network, BFS is a preferred choice.

Peer-to-Peer Networking

BFS is used in peer-to-peer network as a traversal method to find all the neighbouring nodes. For example, BitTorrent uses BFS for peer-to-peer communication.

3.6.4 Uniform Cost Search (UCS)

The UCS algorithm is used to find an optimal solution to the goal state when the step costs are not the same. In such a scenario, the algorithm computes the cumulative cost to expand each node from the root node to the goal node. It neither traverses depth nor breadth, but searches for the next node with the lowest cost. Sorting is done in increasing cost of the path to a node so that the algorithm can explore paths in the increasing order of cost. UCS always expands the node with the least cost. It is identical to BFS if each transition has the same cost. Remember that,

Cost of a node is defined as:

cost(node) = cumulative cost of all nodes from root

$\text{cost}(\text{root}) = 0$

In **Fig. 3.18**, if S is the start node and G is the goal state, then node A is explored as it has the lowest cost (the other node G has a higher cost). Now, node A has two child nodes—B and C. C having the lowest step cost is traversed and from there, node D, followed by G is reached. Though UFS performs well for this algorithm, it may not work with all cases.

Thus, we see that uniform-cost search (UCS) expands nodes based on their path costs from the root node. It can be used to traverse any graph/tree where the optimal cost is the traversal criteria. The uniform-cost search algorithm is implemented by the priority queue, giving maximum priority to the lowest cumulative cost. If you observe carefully, you will find that UCS is equivalent to BFS algorithm if the path cost of all edges is the same.

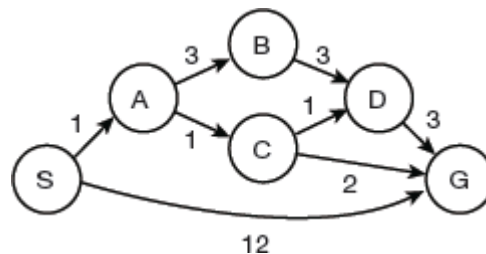


FIGURE 3.18 Uniform cost search

The UCS algorithm is a complete and optimal algorithm with time and space complexity $O(b^{(c/\epsilon)})$, where ϵ is the lowest cost and c is the optimal cost.

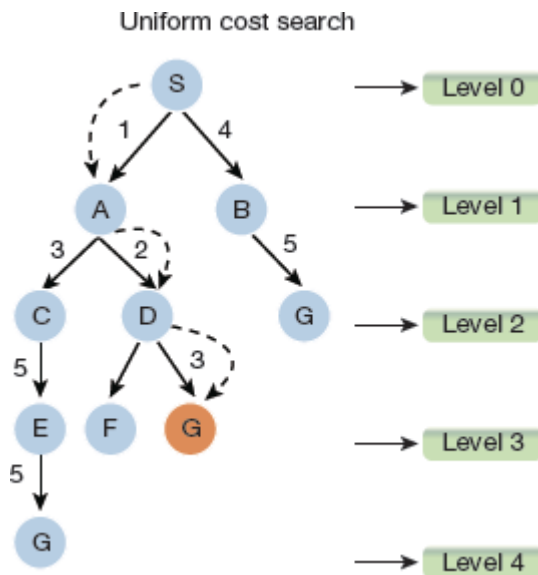
Advantages

1. It finds an optimal solution by considering the least cost at every state.
2. UCS is complete only if states are finite and if there is no loop with zero weight.
3. UCS is optimal only if there is no negative cost.

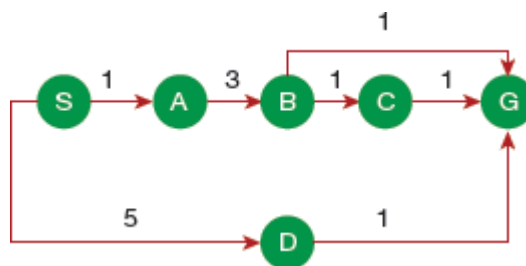
Disadvantages

1. The algorithm may get stuck in an infinite loop as it considers only cost and not the number of steps taken to reach the goal state.
2. It explores nodes in every 'direction'.
3. It does not have any information about the location of the goal state.
4. It requires more space for storing information about nodes.
5. The UCS must explore all paths including the long ones.

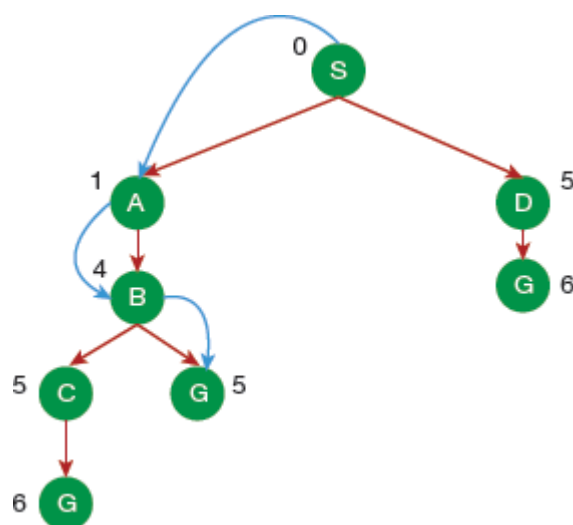
Example: Consider the tree given below and its UCS traversal to reach node G from S.



Example: Find the path and cost to move from node S to node G in the graph given below.



Solution: For a better clarity, we can draw an equivalent search tree for the graph. Using UCS, the path with the least cumulative cost is chosen.



Path: S -> A -> B -> G

Cost: 5

3.6.5 Iterative Deepening Depth-First Search (IDDFS)

The iterative deepening algorithm combines the features of DFS and BFS algorithms to find out the best depth limit that can be used to implement the DLS algorithm. IDDFS does so by gradually increasing the limit until the goal is found.

The algorithm performs depth-first search to level 1, starts over, executes a complete depth-first search to level 2 and continues until the solution is found. It never creates a node until all lower nodes are generated. It only saves a stack of nodes.

IDDFS gets benefitted by fast search technique of BFS algorithm and memory efficiency of the depth-first search algorithm. This algorithm is widely used when search space is large, and depth of the goal node is not known.

Advantages

1. It combines the benefits of BFS and DFS search algorithms—fast search and memory efficiency.
2. The algorithm is complete is if the branching factor is finite.
3. IDDFS is an optimal algorithm if path cost is a non-decreasing function of the depth of the node.

Disadvantages

1. It repeats all the work of the previous phase.
2. It takes more time (exponential) to reach the goal node.
3. The algorithm fails when the BFS fails.

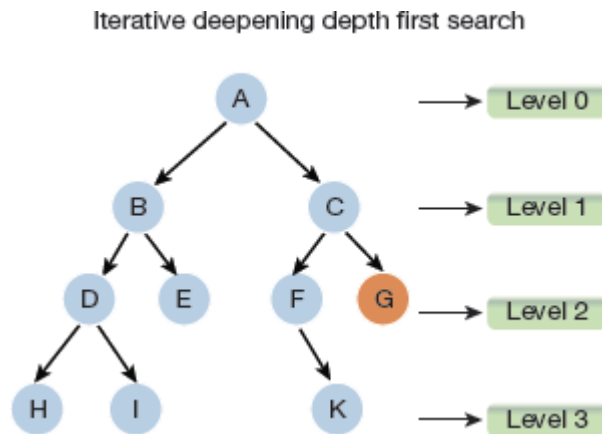
Example: Traverse the given tree using the iterative deepening depth-first search algorithm.

Solution: In the first iteration, node A at level 0 is explored.

In the second iteration, nodes B and C are traversed at level 1.

In the third iteration, nodes D, E, F and G are traversed.

In the fourth iteration, node H is reached.

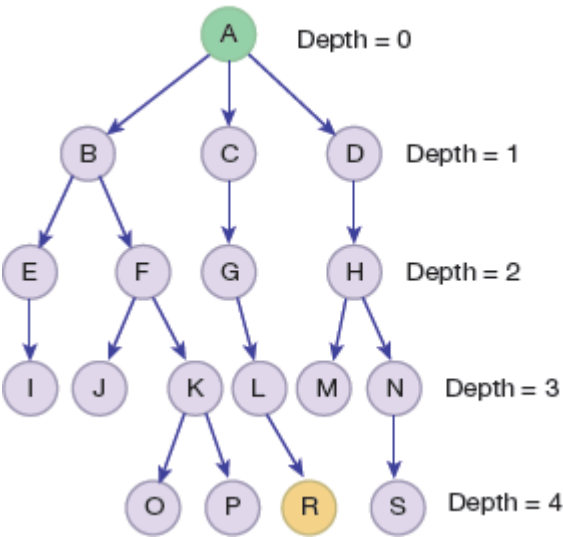


The algorithm ends when it finds a solution at depth d . The number of nodes created at depth d is b^d and at depth $d-1$ is b^{d-1} .

Time complexity of IDDFS algorithm is $O(b^d)$ and its space complexity is $O(bd)$, where b is the branching factor and depth of the tree is d .

In IDDFS, we perform DFS up to a certain 'limited depth,' and keep increasing this 'limited depth' after every iteration.

Example: Consider the tree given below and demonstrate the application of IDDFS.



The tree can be visited as: A B E F C G D H
Depth = {0, 1, 2, 3, 4}

Depth limits	IDDFS
0	A
1	A B C D
2	A B E F C G D H
3	A B E I F J K C G L D H M N
4	A B E I F J K O P C G L R D H M N S

3.6.6 Bidirectional Search

In a bidirectional search, searching is done **from both the directions simultaneously**. It searches forward from initial state and backward from goal state till both meet to identify a common state.

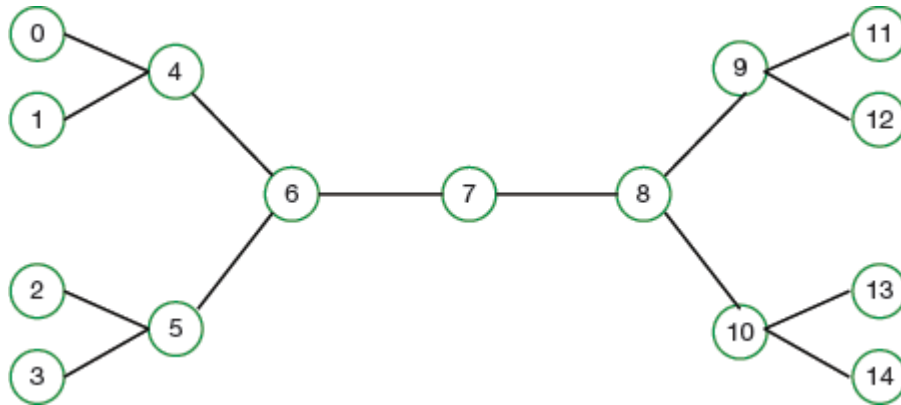
The path from initial state is concatenated with the inverse path from the goal state. Each search is done only up to half of the total path.

The algorithm finds the smallest path from the source node to the goal node. Thus, bidirectional search replaces single search graph with two smaller sub graphs—one starting from initial or start node to the goal node (termed as forward search) and the other starting from goal node towards the source node (known as backward search).

The search process terminates when the two graphs intersect.

Bidirectional search can be guided by a heuristic estimate of remaining distance from source to goal and vice versa. Heuristic means finding the shortest path from the current node in the graph to the goal node.

Example: Consider the graph given below and apply bidirectional search on it to reach goal node 14 from source node 0.



Solution: Two searches are executed simultaneously-, one from node 0 and the other from node 14. Both these searches intersect at node 7. At this point, we have found a path from node 0 to 7 and from node 14 to 7. The search process terminates successfully, thereby avoiding unnecessary exploration.

Advantages

1. It is faster than other algorithms
2. **It avoids unnecessary** exploration of nodes
3. Time and space complexity of searching is $O(b^{d/2} + b^{d/2})$, which is far less than $O(b^d)$. Here, **b** is the branching factor of the tree and d is the distance of goal node from the source node.
4. The algorithm drastically reduces searching time by supporting simultaneous searches.
5. It takes less memory capacity to store all the searches.
6. Bidirectional search is complete if BFS is used in both searches.
7. The algorithm is optimal if BFS is used for search and paths have uniform cost.

Disadvantages

1. The algorithm can be used only when the goal state is clearly known.
2. It is difficult to implement.
3. The algorithm must be robust enough to correctly deduce the intersection point where it can terminate. Otherwise, the algorithm may get stuck in an infinite loop.
4. It is very difficult to search backwards through all states.

When to Use Bidirectional Approach?

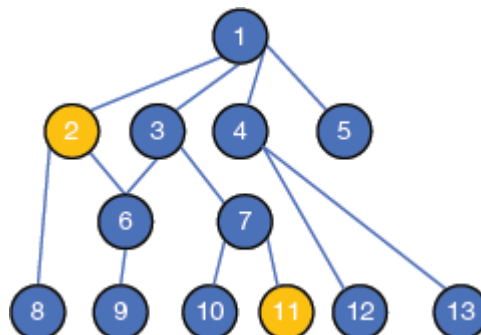
Bi-directional search can be best used in the following situations:

- When both the starting node and the goal node are unique and completely defined.
- When the branching factor is exactly the same in both directions.

Example: Consider the graph given below and demonstrate the application of bidirectional search from Start forward search from node 2 and backward search from node 11.

Do BFS from both directions.

In the forward search process, nodes 1, 6 and 8 are explored. In the backward search process, mode 7 is explored. But no intersection node is encountered.



Next explore node 3 while doing a forward search and node 3 while doing a backward search. Here, we find intersecting node as node 3, so the algorithm terminates and the path is- 2 -> 1 -> 3 -> 7 -> 11.

3.7 Informed Search Algorithms

The uninformed search algorithms that we have learnt so far had no knowledge about search space. But informed search algorithm, as the name suggests, contains some information about how far we are from the goal, cost of the path, how to reach the goal, etc. This knowledge helps agents to explore less to the search space and reach the goal node more efficiently.

The informed search algorithm which is extensively used in a large space is also known as heuristic search as it uses a heuristic function. The heuristic function takes the current state of the agent as its input and estimates how close the agent is from the goal. Though the heuristic method may not always give the best solution, it is, however, guaranteed to find a good solution in reasonable time.

A heuristic function $h(n)$, calculates the cost of an optimal path between the pair of states. The value of the heuristic function is always positive. We can write,

$$h(n) \leq h^*(n)$$

where $h(n)$ is heuristic cost, and $h^*(n)$ is the estimated cost. This means that the heuristic cost should be less than or equal to the estimated cost.

3.7.1 Pure Heuristic Search

To solve large problems with a large number of possible states, problem-specific knowledge needs to be added to increase the efficiency of search algorithms. In such a situation, heuristic search algorithm performs well.

Heuristic search is the simplest form of heuristic search algorithms that expands nodes based on their heuristic value $h(n)$. The algorithm maintains two lists—OPEN and CLOSED. All nodes that have already been expanded are placed in the CLOSED list, while others that have not been expanded are in the OPEN list.

On each iteration, nodes with the lowest heuristic value are expanded. Then, the heuristic function is applied to the child nodes and they are placed in the open list according to their heuristic value. The shorter paths are saved and the longer ones are disposed. The process continues until a goal state is found.

In this section, we will discuss two important informed search algorithms—**best first search algorithm (greedy search) and A* search algorithm**.

3.7.2 Best-First Search Algorithm (Greedy Search)

Greedy best-first search algorithm combines depth-first search and breadth-first search algorithms and always selects the path which appears best at that moment. It uses the heuristic function and best-first search at each step to choose the most promising node. The most promising node is the one that is closest to the goal node. The greedy best first algorithm is implemented by the priority queue.

Heuristic evaluation functions calculate the cost of optimal path between two states.

Best first search algorithm steps:

Step 1: Insert the starting node in the OPEN list.

Step 2: Stop and return failure if the OPEN list is empty,

Step 3: From the OPEN list, remove the node having the lowest value of $h(n)$, and insert it in the CLOSED list.

Step 4: Expand the node (removed in Step 3) and generate its successors.

Step 5: Check each successor to find if it is the goal node or not. If any successor node is found to be the goal node, then return success and terminate the search, else go to Step 6.

Step 6: Check if each successor node is in either OPEN or CLOSED list. If the node is not present in any of the lists, then add it to the OPEN list.

Step 7: Go to Step 2.

Advantages

1. It takes advantage of both BFS and DFS algorithms.
2. Best first search algorithm is more efficient than BFS and DFS algorithms.

Disadvantages

1. In some worst-case scenarios, best first search can behave as an unguided depth-first search.
2. Like DFS, best first search algorithm can get stuck in a loop.
3. Best first search algorithm is not an optimal algorithm.

Example: Consider the tree given below and traverse it using greedy best-first search algorithm.

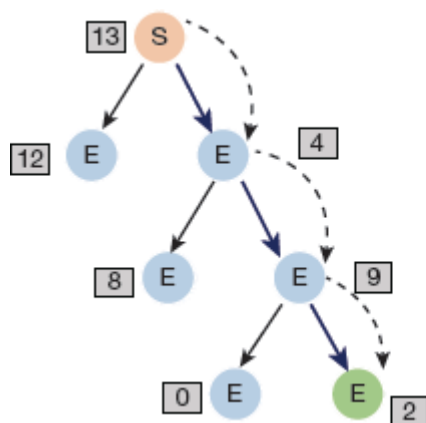


Table shows the heuristic estimates:

Node	$h(n)$	Node	$h(n)$	Node	$h(n)$
A	11	E	4	I, J	3
B	5	F	2	S	15
C, D	9	H	7	G	0

Solution: Expand the node S and put it in the CLOSED list. Generate its successors and place them in the OPEN list.

Open [A, B], Closed [S]

Iteration 1: Since $h(n)$ for node B is less than that of node A, expand B and place it in CLOSED list. Now only node A is in the OPEN list. Open [A], Closed [S, B]

Iteration 2: Generate successors of node B and place them in the OPEN list. Open [E, F, A], Closed [S, B].

Since node F has the lowest heuristic value among all nodes in the OPEN list, we will expand node F. Now, Open [E, A], Closed [S, B, F]

Iteration 3: Generate successors of node F and place them in the OPEN list. So, Open [I, G, E, A], Closed [S, B, F]

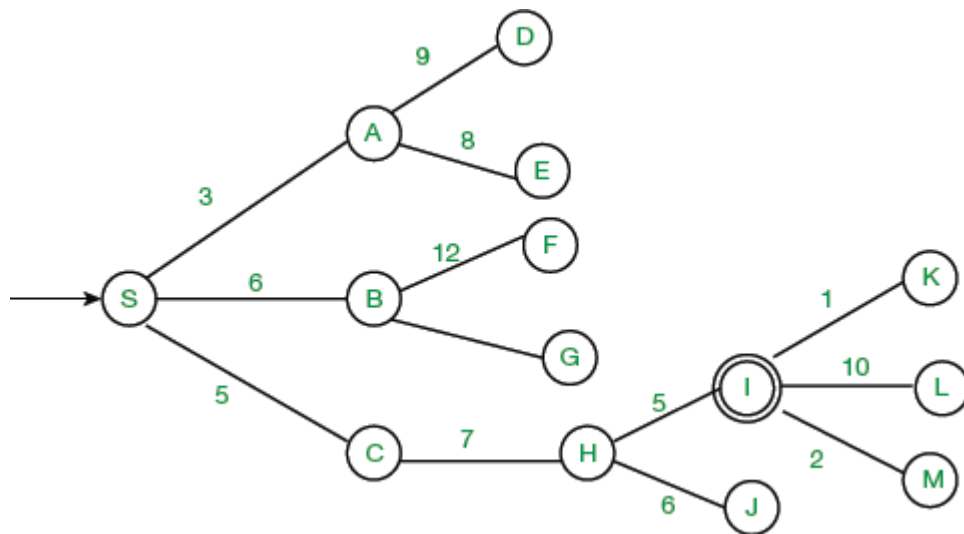
Now, generate succors of node F and place them in OPEN list. Since one of the successors of node F is the goal node, the algorithm returns success and terminates. The path is given by the nodes present in the CLOSED list. Open [I, E, A], Closed [S, B, F, G]

Hence, the final solution path will be: **S**----> **B**---->**F**----> **G**

The worst-case time complexity and space complexity of greedy best first search is $O(b^m)$ where, m is the maximum depth of the search space.

Although the algorithm is complete, it can, at times, behave like an incomplete algorithm even if the given state space is finite. This makes the Greedy best first search algorithm not an optimal one.

Example: Apply greedy best first search algorithm on the graph given below to reach node I from node S.



Solution: Add node S in the CLOSED list and place its successors in the OPEN list. Open [A, B, C], Closed [S]

Remove A from the OPEN list as it has minimum $h(n)$, place it in CLOSED list and put its successors in the OPEN list.

Open [B, C, E, D], Closed [S, A]

Remove C from the OPEN list as it has minimum $h(n)$, place it in CLOSED list and put its successors in the OPEN list.

Open [B, E, D, H], Closed [S, A, C]

Remove B from the OPEN list as it has minimum $h(n)$, place it in CLOSED list and put its successors in the OPEN list.

Open [E, D, H, F, G], Closed [S, A, C, B]

Remove H from the OPEN list as it has minimum $h(n)$, place it in CLOSED list and put its successors in the OPEN list. Since I is the successor of node H, the algorithm returns success.

Open [E, D, F, G, I, J], Closed [S, A, C, B, H]

3.7.3 A★ Search Algorithm

The A★ search algorithm, best-known form of best first search, uses heuristic function $h(n)$, and $g(n)$ which specifies the cost to reach the node n from the start state (refer **Fig. 3.19**). It solves the searching problem efficiently by expanding less search tree to find shortest path and give an optimal result faster. That is, it avoids expanding paths that are already expensive, but expands most promising paths first.

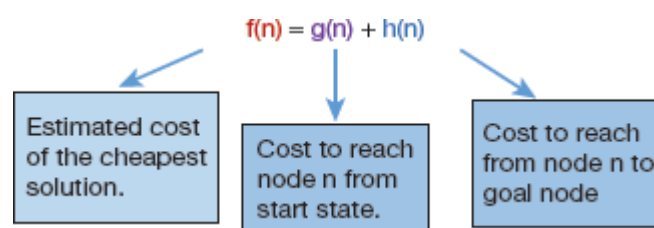


FIGURE 3.19 Calculating Cost of A★ Algorithm

The A* search algorithm, uses search heuristic as well as the cost to reach the node. Sum of both the costs gives a number known as a **fitness number**.

At each point in the search space, node having the minimum $f(n)$ is expanded. The algorithm terminates when the goal node is found. The steps of this algorithm can therefore be summarized as given below.

Step1: Put the starting node in the OPEN list.

Step 2: If the OPEN list is empty then return failure and stop.

Step 3: From the OPEN list, choose the node having minimum value of $f(n)$. If this node is the goal node, then return success and stop.

Step 4: Expand the node selected in Step 3 and generate its successors. Put the node in the closed list. For each successor, check if it is present in the OPEN or CLOSED list. If it is not present in either then compute its $f(n)$ and place it into Open list.

Step 5: Else if the successor node is already in OPEN or CLOSED list, then attach it to the back pointer reflecting the lowest $g(n')$ value.

Step 6: Return to **Step 2**.

Points to Remember

1. **Complete:** A* algorithm is complete as long as:
 1. Branching factor is finite.
 2. Cost at every action is fixed.
2. **Optimal:** A* search algorithm is optimal if it follows the two conditions:
 1. **Admissible:** The first condition requires for optimality in that $h(n)$ should be an admissible heuristic for A* tree search. An admissible heuristic is optimistic in nature.

2. **Consistency:** The second required condition is consistency for only A* graph-search.

If the heuristic function is admissible, then A* tree search will always find the least cost path.

3. **Time Complexity:** The time complexity of A* search algorithm depends on heuristic function, and the number of nodes expanded is exponential to the depth of solution d . So, the time complexity is $O(b^d)$, where b is the branching factor.

4. **Space Complexity:** The space complexity of A* search algorithm is $O(b^d)$.

Advantages

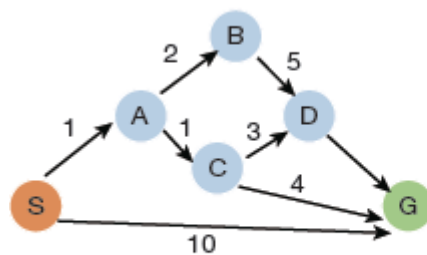
1. It is better than all other search algorithms.
2. A* search algorithm is optimal and complete.
3. The algorithm can be used to solve very complex problems.
4. It is used by many games and web-based maps to find the shortest path efficiently.

Disadvantages

1. It may not always produce the shortest path since it is based on heuristics and approximation.
2. A* search algorithm is more complex than other search algorithms
3. It uses more memory as it keeps all generated nodes in the memory. This makes it difficult to use for large-scale problems.
4. At times, A* algorithm returns the path which occurred first, and ignores searching all other paths.
5. The efficiency of A* algorithm depends on the quality of heuristic.

Example: Consider the graph given below and traverse it using the A* algorithm.

Note that $f(n)$ will be calculated by adding $h(n)$ and the cost of reaching that node.



State	$h(n)$
S	5
A	3
B	4
C	2
D	6
G	0

Solution:

Initially, we start with node S having $h(n) = 5$. Therefore, $\{(S, 5)\}$

Iteration1: From node S, we can reach node A and node G.

$h(n)$ of node A is 3 and the cost to reach node A from node S is 1.

Therefore, $f(n)$ of node A = 4. Similarly, node G has $h(n) = 0$ and the cost to reach node G from node S is 10. Therefore, $f(n)$ of node G = 10.

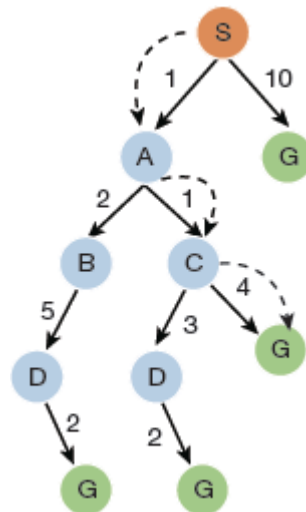
$\{(S \rightarrow A, 4), (S \rightarrow G, 10)\}$

Iteration2: Since node A has less $f(n)$ as compared to node G, we will expand node A.

$h(n)$ of node B is 4 and the cost to reach node B from node S is 3.

Therefore, $f(n)$ of node B = 7. Similarly, node C has $h(n) = 2$ and the cost to reach node C from node S is 2. Therefore, $f(n)$ of node C = 2.

$\{(S \rightarrow A \rightarrow C, 4), (S \rightarrow A \rightarrow B, 7), (S \rightarrow G, 10)\}$



Iteration3: Since node C has less $f(n)$ as compared to other nodes, we will expand node C.

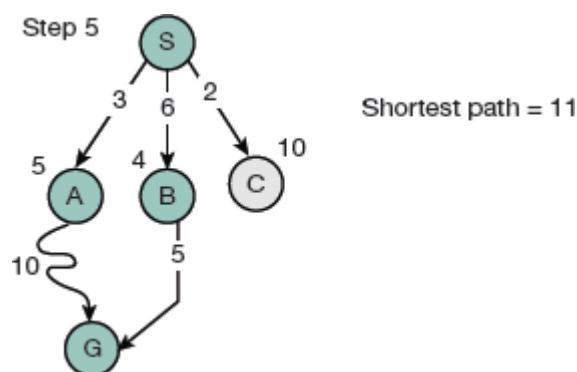
$h(n)$ of node D is 6 and the cost to reach node D from node S is 5.

Therefore, $f(n)$ of node D = 1. Similarly, node G has $h(n) = 0$ and the cost to reach node G from node S is 6. Therefore, $f(n)$ of node G = 2.

$\{(S \rightarrow A \rightarrow C \rightarrow G, 6), (S \rightarrow A \rightarrow C \rightarrow D, 11), (S \rightarrow A \rightarrow B, 7), (S \rightarrow G, 10)\}$

Iteration 4: Since G is the goal node, the algorithm will return success and terminate giving the final path as, $S \rightarrow A \rightarrow C \rightarrow G$ with cost 6.

Example: Consider the graph given below and traverse it from node S to G using A^* algorithm.



In the graph, the cost to reach node A from node S is $3 + 5 = 8$, the cost to reach node B from S is $6 + 4 = 10$ and the cost to reach C is $2 + 10 = 12$.

Undoubtedly, node A is processed further. Now,

$\{(S \rightarrow A, 8), (S \rightarrow B, 10), (S \rightarrow C, 12)\}$

In the next iteration, node is found to be reached from A with a total cost of $3 + 10 = 13$.

$\{(S \rightarrow A \rightarrow G, 13), (S \rightarrow B, 10), (S \rightarrow C, 12)\}$

Now, traversing the $S \rightarrow B$ path to reach G, we get total cost = 11 ($6 + 5$). Therefore,

$\{(S \rightarrow A \rightarrow G, 13), (S \rightarrow B \rightarrow G, 11), (S \rightarrow C, 12)\}$

The final path to G from S is thus, $S \rightarrow B \rightarrow G$ with a cost of 11.

3.7.4 AO* Algorithm

AO* algorithm is based on problem decomposition in which a problem is broken down problem into smaller pieces.

Therefore, AO* algorithm can be applied only on those algorithms that can be divided into a set of sub problems. Each sub problem can be solved separately. The final result is obtained by combining all the solutions. Usually, AND-OR graphs /trees are used for representing the solution. The decomposition of the problem generates AND arcs.

Figure 3.20 shows an AND-OR graph. It shows that if you want to have breakfast then you can eat bread or have fruits and milk together. The arc denotes AND condition.

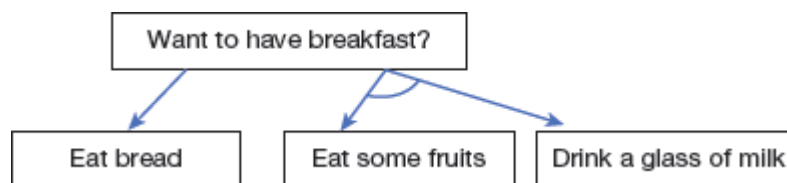


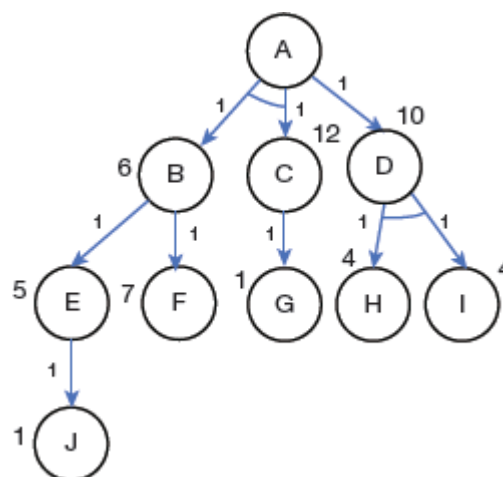
FIGURE 3.20 AND-OR graph

3.7.4.1 A* Vs AO*

Both A* and AO* algorithms are informed search techniques and use heuristic values to solve the problem.

Although both the algorithms guarantee solution, only A* guarantees an optimal solution. By optimal, we mean the shortest path with the lowest cost. The AO* algorithm does not always provide an optimal solution. This is because AO* does not explore all the solution paths once it finds a solution.

Example: Consider the tree given below and apply the AO* algorithm on it.



Remember that the AO* algorithm always moves towards a lower cost value. The cost function, $F(n) = G(n) + H(n)$, where, **H(n) is the heuristic or the estimated value of the node** and **G(n) is the actual cost or edge value**.

Step 1: From node A, we can either choose D or B-C together. The cost of both the paths are calculated as,

$$F(A \text{ to } D) = 1 + 10 = 11 \text{ and}$$

$$F(A \text{ to } B \text{ and } C) = 1 + 1 + 6 + 12 = 20$$

We will choose node D as the next node to traverse as it has a lower cost than path B-C.

Step 2: From D, we are left with only one choice H-I. Now, the cost of H-I from node D is calculated as,

$$F(D \text{ to } H-I) = 1+1+ 4 +4 =10$$

Correspondingly, the cost of H-I from node A is 11 as,

$$F(A \text{ to } D \text{ to } F-E) = 1+10 =11$$

If the goal state is found on the path from A to H-I, then the other path remains unexplored.

Let us explore the other path from node A to B-C. We had calculated,

$$F(A \text{ to } B \text{ and } C) = 1 + 1 + 6 +12 = 20$$

Step 1: From B, there are two nodes can be reached- E and F.

$$F(B-E)= 5+1 = 6 \text{ and } F(B-F) = 7 + 1 = 8$$

Step 2: Since cost from **F(B-F)** is more than that of the path B-E, the algorithm moves to node J.

$F(E-J) = 1 +1 = 2$, which is less than heuristic value 5. So, the new heuristic value from E is 2.

Step 3: We need to update the cost from E to B, which is now,

$$F(E-B)= 1+2 =3$$

Therefore, the new heuristic value of B is 3.

Step 4: We need to update the value of A as A is associated with both B and C.

But to update the heuristic value of A, we must know the value of node C.

From C, node G can be reached with a cost of 1. So, the heuristic value of C is now $1 + 1 = 2$.

Since cost from C to G = $F(C-G) = 2$ which is less than its heuristic value, the new heuristic value of C is 2.

Step 5: Now the updated from A- BC that is $F(A-BC) = 1+1+2+3 = 7$ which is less than $F(A-D) = 11$.

This shows choosing path A-BC is more cost effective and thus better path A-D.

However, this is possible if the algorithm explores this path as well. But the algorithm will not explore this path. We have just shown this path to demonstrate the weakness of the AO* algorithm.

The AO* does not give optimal performance in all the cases.

3.7.4.2 Algorithm

The AO* algorithm maintains two lists like the A* algorithm.

The **OPEN list** contains the nodes that have been traversed have not been marked as solvable or unsolvable.

The **CLOSE** list contains nodes that have already been processed.

Steps in the Algorithm:

Step 1: Put the initial/ starting node in the OPEN list.

Step 2: Find the most promising solution tree say T0.

Step 3: Select a node n that is both a member of T0 as well as present in the OPEN list. Remove n from OPEN and place it in CLOSE list.

Step 4: If n is the leaf node as well as the goal node, then label n and all its ancestors as solved. If the starting node is marked as solved then re-

turn success and exit.

Else if n is not a solvable node, then mark n as unsolvable. If starting node is marked as unsolvable, then return failure and exit.

Step 5: If n is not a leaf node, then expand n . Enumerate all its successors, compute their $h(n)$ values and push them into OPEN.

Step 6: Return to Step 2.

Step 7: Exit.

Advantages

1. It gives optimal solution.
2. It can be used for both OR and AND graph when nodes are traversed in a particular order.

Disadvantages

1. It does not find the optimal solution in every case.
2. Its complexity is more than other algorithms.

3.8 Hill Climbing Algorithm in Artificial Intelligence

Hill climbing is a heuristic search technique that works well on a large set of inputs to find a sufficiently good solution to the problem. This solution may not be the global optimal maximum. The algorithm can be applied to problems that need to maximize or minimize a given real function by choosing values from the given inputs. For example, in the travelling salesman problem, hill climbing algorithm is used to minimize the distance traveled by the salesman. We know that a **heuristic function** uses all the available information to rank all the possible alternatives at any branching step in the search algorithm and select the best route out of possible routes.

3.8.1 Features of Hill Climbing

The **hill climbing algorithm** is a variant of **generate and test** algorithm which works in the following manner:

Step 1: Generating all possible solutions.

Step 2: Testing them to see if it is the expected solution.

Step 3: If the solution is found, the algorithm terminates else goes to step 1.

Heuristic search algorithm may not find the optimal solution to the problem but gives a good solution in a **reasonable time**.

Thus, the hill climbing algorithm takes feedback from the test procedure and uses it to generate the next move in the search space.

At any point in the state space, the hill climbing algorithm uses the greedy approach to move the search in the direction that optimizes the cost of function so that an optimal solution can be obtained at the end. To take such decisions, the algorithm uses a heuristic value which indicates how close the current state is to the goal state.

It does not support backtracking in the search space, as it does not remember the previous states.

Remember the following:

1. **Hill climbing = generate-and-test + heuristics**
2. **A node of hill climbing algorithm has two components—state and a value.**
3. **Hill climbing algorithm is mostly used when a good heuristic is available.**
4. **The algorithm does not maintain and handle the search tree or graph. It only keeps a single current state.**

5. Hill climbing is the simplest implementation of a **genetic algorithm**. Although this algorithm is less thorough than the traditional ones, it executes faster when compared with other genetic algorithms.

3.8.2 Types of Hill Climbing

In this section, we will discuss different variants of hill climbing algorithm.

3.8.2.1 Simple Hill Climbing

The algorithm examines the neighbouring nodes and selects the first neighbouring node which optimizes the current cost as the next node. The steps performed in this algorithm can be given as follows.

Step 1: Evaluate the initial state. If it is the goal state, then return success and terminate the process. Else, make initial state as the current state.

Step 2: Repeat Step 2 until the solution state is found or no new operators are present which can be applied to the current state.

1. Select a state that has not been set as the current state. Use this state to produce a new state.
2. Evaluate the new state.
 1. If the current state is the goal state, then return success and stop
 2. If the new state is better than the current state, then set it as the current state and processed further.
 3. If the new state is not better than the current state, then continue in the loop until a solution is found.

Advantage

It takes less time.

Disadvantages

1. Solution is less optimal.
2. It does not guarantee any solution

3.8.2.2 Steepest-Ascent Hill Climbing

The algorithm examines all the neighbouring nodes. It then selects the node closest to the solution state as the next node. Steps to perform steepest-ascent hill climbing can be given as follows:

Step 1: Evaluate the initial state. If it is the goal state, then return success and terminate the process. Else, make initial state as the current state.

Step 2: Repeat Step 2 until either the solution state is found or the current state does not change

1. Select a state that has not been set as the current state.
2. Initialize a new 'best state' equal to current state. Use this state to produce a new state.
3. Evaluate the new state.
 1. If the current state is the goal state, then return success and stop.
 2. If the new state is better than the best state, then make it the best state; else continue with the loop to generate and test another new state.
 3. If the new state is not better than the best state, then make best state as the current state. Continue in the loop until a solution is found.

3.8.2.3 Stochastic Hill Climbing

Unlike the previous two variants, the scholastic hill climbing algorithm does not examine all the neighbouring nodes before deciding which node to select next. Rather, it randomly selects a neighbouring node and determines whether to move to that neighbour or to examine another depending on the amount of improvement in that neighbour. Steps to be performed in scholastic hill climbing are listed below.

Step 1: Evaluate the initial state. If it is the goal state, then return success and terminate the process. Else, make initial state as the current state.

Step 2: Repeat Step 2 until either the solution state is found or the current state does not change.

1. Select a state that has not been set as the current state.
2. Apply successor function to the current state and generate all the neighbour states.
Randomly select (or use a probabilistic function) a state from all the neighbouring states generated in 2 (b) and are better than the current state.
3. If the state selected in 2 (c) is the goal state, then return success, else make it the current state and continue with the loop.

3.8.3 State Space Diagram for Hill Climbing

The state-space diagram (**Fig. 3.21**) is a graphical representation of the set of states that can be attained in the search algorithm with respect to the value of the objective function (the function that has to be maximized). While the state space is represented on the X-axis, the values of objective function corresponding to a particular state are marked on the Y-axis.

The best solution is the state space where the objective function has a maximum value (global maximum). The goal of the hill climbing algorithm is to find the global maximum and local maximum.

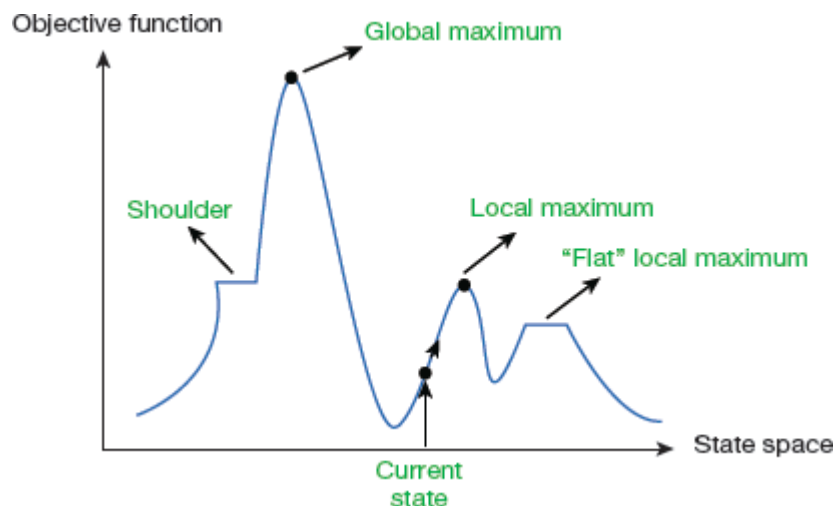


FIGURE 3.21 State Space Diagram for Hill Climbing

3.8.3.1 Understanding Different Regions in the State Space Diagram

Any hill climbing algorithm cannot reach the optimal or the best state (also known as the global maximum) if it enters any of the following regions in the state space diagrams.

Local maximum is a state which is certainly better than its neighbouring state. At this state, the value of the objective function is higher than its neighbours.

However, there does exist a state which is better than it. This better state is known as the global maximum.

Global maximum is the best possible state in the state space diagram and has the highest value for the objective function.

Plateau/Flat local maximum is a flat region in the state space where neighbouring states have the same value.

Ridge is a special type of local maximum that is higher than its neighbours but itself has a slope.

Current state is the region in the state space diagram where the search is being done currently.

Shoulder is identified as a flat plateau that has an uphill edge.

3.8.3.2 Resolving Problems Posed By Different Regions in Hill Climbing

Local Maximum

We know that at the local maximum, all neighbouring states have a value less than that of the current state. Since hill climbing uses a greedy approach, it will not move to the worse state and terminate itself once local maximum is attained. Therefore, the problem here is that the algorithm will terminate without exploring a better solution that may exist.

To overcome this problem, backtracking technique can be used. A list of visited states is maintained so that if the search reaches an undesirable state, then it can backtrack to a previous state and explore a new path.

Plateau

Plateau is a flat land indicating that all neighbours have the same value. In such a scenario, it is difficult to find the best direction.

To overcome plateaus, a big jump is required to come out of this region. For this, a state far away from the current state is selected randomly to increase the probability of landing in a non-plateau region.

Ridge

Ridge is any point in the space that looks like a peak as movement in all possible directions leads to downward value. In such a situation, the algorithm terminates when it reaches a ridge. It is a special form of local maximum.

To overcome issues related to Ridge, use two or more rules before testing to move in several directions at once. For this, bidirectional search can be used or the algorithm can be moved in different directions.

3.8.4 Simulated Annealing

We have seen that a hill-climbing algorithm may get stuck on a local maximum or may not find an optimal solution while doing a random walk.

Simulated annealing is an algorithm which is both efficient and complete.

Usually, a hill climbing algorithm does not support downward moves but simulated annealing allows such moves to escape local maxima and guarantee both efficiency and completeness.

In simulated annealing, the algorithm picks a random move rather than the best move. If the random move improves the state, then it follows the same path. Otherwise, the algorithm moves downhill and chooses another path.

3.8.5 Applications

The hill climbing algorithm can be applied in the following areas:

Marketing

The hill-climbing algorithm is used by marketing people who need to travel to several cities multiple times in a year to develop the best marketing plans. This problem is popularly known as travelling-salesman problem. The algorithm can help them to optimize the distance covered and reduce their travel times by establishing the local minima efficiently.

Robotics

In robotics, hill climbing algorithm is used to enhance the coordination among different systems and components in robots. It is also used in inductive learning methods for coordinating with multiple robots in a team.

Job Scheduling

Job scheduling is a process in which system resources are allocated to different tasks within a computer system. It is achieved by migrating jobs from one node to another (possibly a neighbouring node). In such a scenario, the hill climbing algorithm can be used to establish the right migration route.

3.8.6 Pros and Cons of Hill Climbing Algorithms

Advantages

1. Hill climbing is useful in routing the related problems. For example, in portfolio management, chip designing, job scheduling, vehicle routing, automatic programming, circuit designing, etc.
2. It can be used in optimization problems, especially when there is limited computational power.
3. The algorithm is more efficient than other search algorithms.
4. It can easily work in cases where the goal is to point out the best suitable state.
5. All in all, hill climbing algorithm is popularly used for optimizing the given problems. It provides outstanding solutions to computationally challenging situations.

Disadvantages

However, the disadvantage of this algorithm is that it may terminate at a Local Minima, Ridge or a Plateau.

3.9 Adversarial Search

Adversarial search is a searching algorithm that has more than one entity. Each of these entities has conflicting goals and objectives. These entities participate in a game-like situation. The strategy or game approach of each player depends on the opponent's move. For example, the game situation and intent of the opponent determines whether a player would take a maximizer role or a minimizer role. While Maximizer, maximizes the gain, Minimizer, on the other hand, makes best efforts to limit the damage and minimize the loss.

While normal search has a single entity, adversarial search on the other hand has more than one entity.

Mathematically, adversarial search is based on the concept of '**game theory**' *which states that a game is played between two players. To complete the game, one has to win the game and the other loses automatically.*

Adversarial search can be used for all types of games with predictable as well as unpredictable outcomes by following transparent or blind rules.

Each player considers the movements of other players, then dynamically formulates a strategy to reach the end goals. While playing, players may take unforeseen turns to upset the opponent and win the game. So, a quick reaction is required to counter strategize and win the game. To choose the best optimal strategy in a limited time, techniques like pruning and heuristic evaluation are performed.



FIGURE 3.22 Game Theory

In pruning, the unwanted portions of a search tree that makes no difference in its final result are ignored.

Heuristic evaluation function approximates the cost at each level of the search tree, before reaching the goal node.

3.9.1 Game Scenarios

Adversarial search examines the problem which arises when an agent plans ahead of other agents that are planning against it. It is applied in a multi-agent environment, like in games that are modelled as a search problem and use the heuristic evaluation function.

There are different types of game scenarios which are discussed below.

Games with Perfect Information

Such games are open and transparent as every player has complete information. The players are well aware about the status of the game and the move of the opponent. Examples of such games include chess, checkers, ethello and go.

Games with Imperfect Information

In these games, players do not have any information on the status of the game. They take blind moves by making certain assumptions and predicting the outcomes. Examples of the games include, tic-tac-toe, battleship, blind and bridge.

Deterministic Games

These games are played in a transparent manner. All players can see the moves of the opponents and follow pre-defined rules and pattern. There is no randomness associated with a deterministic game as the rules and patterns are followed very strictly. For example, chess, checkers, tic-tac-toe, go are deterministic games.

Non-Deterministic Games

As the name suggests, non-deterministic games are highly unpredictable. Players take random moves and win by sheer luck or chance. For example, games like poker, monopoly, backgammon fall in this category.

Zero-Sum Games

These are purely competitive games in which one player’s gain is compensated by another player’s loss. The net value of gain and loss is zero, thus, the name.

Here also, each player has different conflicting strategies. So, each player will either try to maximize the gain or minimize the loss, depending on the game situation, strategy likely to be adopted by the opponent and the player’s game plan. This scenario is popularly known as **embedded thinking** in problem resolution in AI. A player can easily backtrack and change his strategy depending on the direction in which the game progresses. Hence, it is also known as backward reasoning in AI. Chess, tic-tac-toe are games that are played in this group.

TABLE 3.2

	Deterministic	Chance Moves
Perfect information	Chess, Checkers, go, Othello	Backgammon, monopoly
Imperfect information	Battleships, blind, tic-tac-toe	Bridge, poker, scrabble, nuclear war

3.9.2 Key Terminology

In AI, a game can be treated as a search algorithm. In this section, let us get familiar to commonly used terms in the field of game playing.

1. **Start of the game:** This is the initial state of the game and configurations.
2. **Players:** They are the participants in the game space.
3. **Action:** Actions are the moves made by the players in the game space.
4. **Result:** This is the outcome of the moves of the players.
5. **Terminal Test:** It is a check that determines whether the game is over or not.
6. **Utility:** It is the final outcome of the game in terms of loss or win. The utility function is also known as **objective function** or **payoff function**. The winner may get a price of -1, + 1 or 0.
 - (-1):** If the PLAYER loses.
 - (+1):** If the PLAYER wins.
 - (0):** If there is a draw between the PLAYERS

3.9.3 Game Tree

Like any other tree, the game tree has a root node at the top and multiple nodes below the root. Nodes of this tree are the game states and edges are the moves made by players. Each node represents certain moves of the players and thus the status of the game. Each layer in the tree is either a maximizer or a minimizer. As stated earlier, the maximizer maximizes the gain and the minimizer minimizes the loss. At each level of the tree, the maximizer at a previous layer will be followed by a minimizer layer. Similarly, a minimizer at a previous layer will be followed by a maximizer layer.

Thus, a player takes a maximizer role or minimizer role depending on the game environment and the opponent's strategy. Let us understand how game playing works.

Example: Tic-Tac-Toe game tree

In the game, there are two players MAX and MIN. Both the players take alternate turns and the game starts with MAX. Each player thinks of an optimal strategy to prevent another one from winning. This implies that MIN is acting against Max in the game. Let's enumerate the process stepwise.

Step 1: Each game has an initial state and more than one player will be playing the game.

Step 2: The root node takes the role of maximizer and fills 'X' in any one of the 3×3 vacant cells.

Step 3: The minimizer assesses the move made by its opponent (maximizer) and accordingly place 'O' in a vacant cell.

Step 4: The maximizer observes the move made by the minimizer in Step 3 and places a 'X' in any of the vacant cell.

Step 5: Repeat Steps 3 and 4 by alternating moves by minimizer and maximizer, respectively, from their standpoint, either till one of the rows or columns gets filled with 'X' or 'O'.

Once the terminal condition is reached, the game is declared to be over. At this point, the measure of utility determines the final outcome—maximizer's win or minimizer's win or draw.

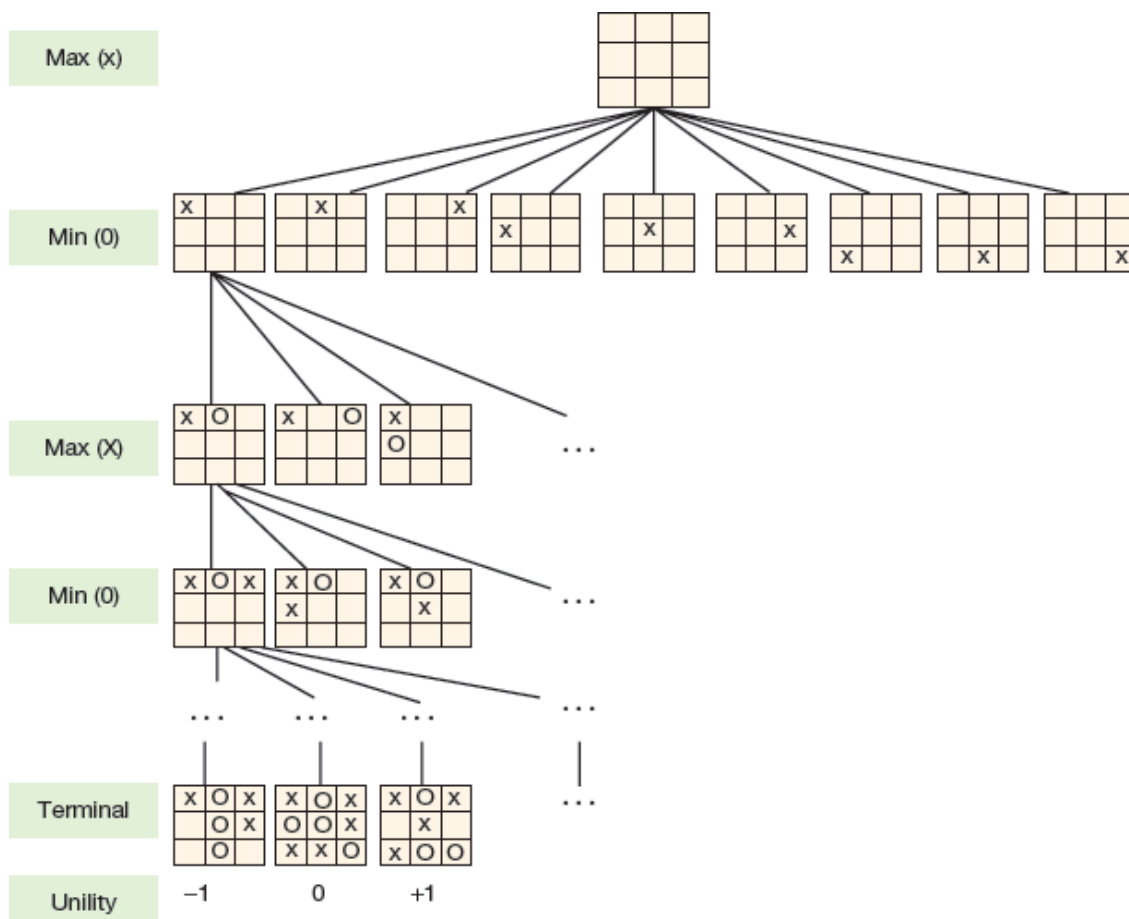


FIGURE 3.23 Game Tree for Tic-Tac-Toe

3.9.4 Uses of Adversarial Search

Adversarial search is widely used in different fields to find optimal solutions in a competitive environment. Some examples are as follows.

1. In the legal system, two advocates argue their stand representing their party's cases. In such a scenario, judges can use adversarial search to analyse and deliver judgment.
2. Business organizations use this search technique while sourcing suppliers and buyers for products/services in a competitive manner.
3. Moreover, this technique can be used in any situation where hard negotiations are required. It helps users to maximize the gain by guiding them to negotiate intelligently and prudently.
4. Adversarial search also forms the base for further development of new techniques like Alpha Beta Pruning.

Adversarial search techniques optimize the time, effort and cost required to take complex decisions.

3.9.5 Mini-Max Algorithm

Mini-max algorithm is a recursive or backtracking algorithm which is used in decision-making and game theory. It takes the optimal move to search through the game-tree. Games like chess, checkers, tic-tac-toe, go and other two-player games use the min-max algorithm to compute the best decision for the current state.

The two players named MIN and MAX compete to let the opponent get the minimum benefit and themselves earn the maximum benefit. So, MAX will always choose the maximum value and MIN will select the minimized value.

The algorithm performs a depth-first search to explore the complete game tree. It proceeds all the way down to the terminal node of the tree, then backtracks the tree.

3.9.5.1 Pseudo-Code for MinMax Algorithm

Pseudocode

```

function minimax(node, depth, isMaximizingPlayer, alpha, beta):
    if node is a leaf node :
        return value of the node

    if isMaximizingPlayer :
        bestVal = -INFINITY
        for each child node :
            value = minimax(node, depth+1, false, alpha, beta)
            bestVal = max( bestVal, value)
            alpha = max( alpha, bestVal)
            if beta <= alpha:
                break
        return bestVal
    else :
        bestVal = +INFINITY
        for each child node :
            value = minimax(node, depth+1, true, alpha, beta)
            bestVal = min( bestVal, value)
            beta = min( beta, bestVal)
            if beta <= alpha:
                break
        return bestVal
// Calling the function for the first time.
minimax(0, 0, true, -INFINITY, +INFINITY)

```

3.9.5.2 Working of Min-Max Algorithm

In the minimax algorithm, DFS is used to reach the terminal or the leaf nodes where terminal values are specified. Once we reach those nodes, the algorithm backtracks the tree until the initial state occurs. Given below are the steps involved in solving the two-player game tree.

Step 1: The algorithm generates the game tree and applies the utility function to get the utility values for the terminal states.

In the tree given in **Fig. 3.24**, node A is the root node that is also the initial state of the tree. Assuming that maximizer takes the first turn, then set the worst-case initial value as -infinity. For minimizer, take the worst-case initial value equal to +infinity.

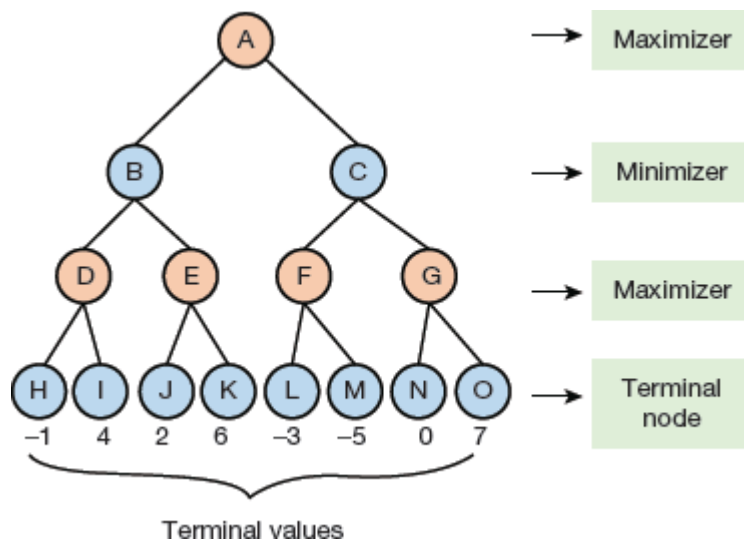


FIGURE 3.24 Min-Max Algorithm Step 1

Step 2: Starting from the layer above the terminal nodes, we will find the utilities value for the maximizer.

Note that the initial value of maximizer is $-\infty$, so each value in terminal state is compared with the initial value of Maximizer. The higher value is then set as the new value of the maximizer as shown in the figure.

For node D, 4 is greater than both -1 as well as $-\infty$. Therefore, the utility of D is 4.

For Node E, 6 is greater than both 2 as well as $-\infty$. Therefore, the utility of E is 6.

For Node F, -3 is greater than both -5 as well as $-\infty$. Therefore, the utility of F is -3.

For Node G, 7 is greater than both 0 as well as $-\infty$. Therefore, the utility of F is 7.

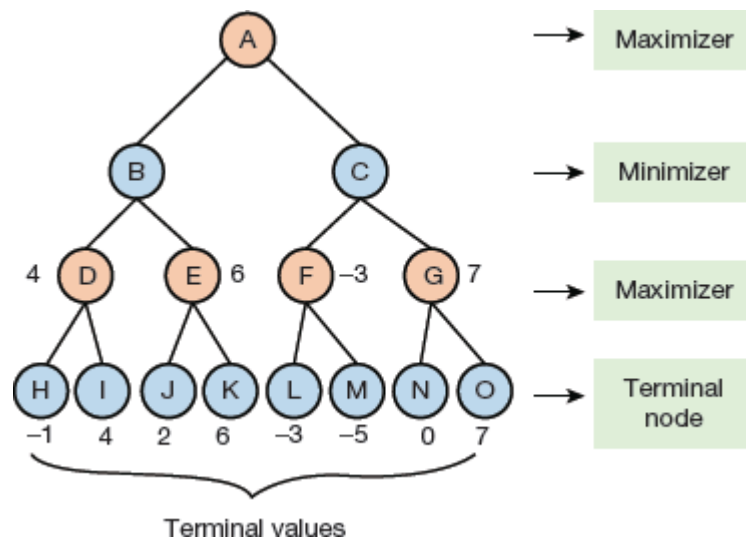


FIGURE 3.25 Min-Max Algorithm Step 2

Step 3: In the next step, minimizer finds the utility value of nodes B and C by comparing values of the nodes with $+\infty$.

For Node B, 4 is less than both 6 as well as ∞ . Therefore, the utility of B is 4.

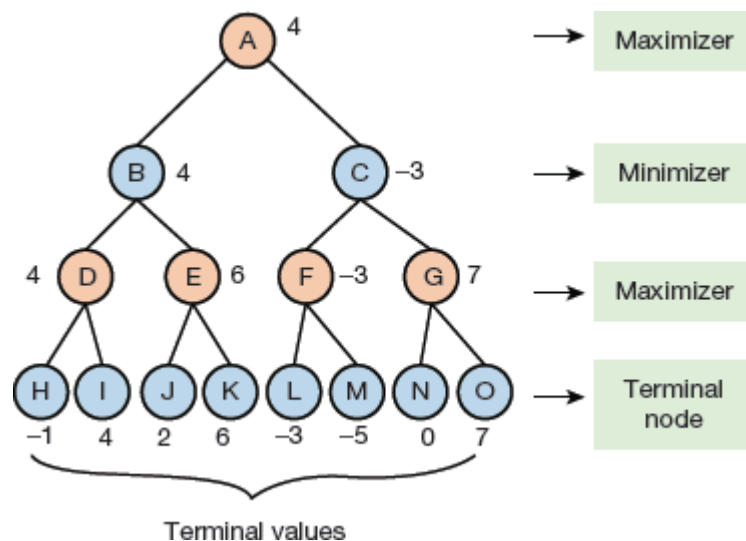


FIGURE 3.26 Min-Max Algorithm Step 3

For Node C, -3 is less than both 7 as well as ∞ . Therefore, the utility of C is -3.

FIGURE 3.27 Min-Max Algorithm Step 4

Step 4: During maximizer's turn, maximum of all nodes value is selected as the maximum value for the root node. Therefore,

For node A, the utility value is 4 because $\max(4, -3) = 4$

3.9.5.3 Properties of Mini-Max Algorithm

1. **Complete:** The Min-Max algorithm is complete as it always finds a solution if one exists in the finite search tree.
2. **Optimal:** The algorithm gives an optimal solution if both the opponents are playing optimally.
3. **Time complexity:** Since the algorithm uses DFS technique, the time complexity is $O(bm)$, where b is branching factor (number of child nodes of a node) of the game-tree, and m is the maximum depth of the tree.
4. **Space complexity:** Space complexity of Mini-max algorithm is same as that of DFS which is $O(bm)$.

Advantages

1. The search space is thoroughly assessed to find an optimal solution.
2. It supports decision-making.
3. New and smart machines are developed with this algorithm.

Disadvantages

1. The minimax algorithm gets very slow for complex games such as chess, go, etc. as these games have a high branching factor, and there are lots of choices at each node.
2. If there are time and space constraints, then applying this algorithm would be difficult as more time and space is required to explore the entire tree.

However, this limitation can be improved by using **alpha-beta pruning**.

3.10 Alpha-Beta Pruning

Alpha-beta pruning is the modified as well as optimized version of the minimax algorithm. We have seen that the minimax search algorithm has a large number of game states to examine. The alpha-beta pruning technique cuts the number of nodes to be explored to half. This is done by using the pruning process in which the correct minimax decision is taken without checking each node of the game tree. In the alpha beta pruning algorithm, we take two threshold parameters—alpha and beta.

Alpha is the best or the highest-value so far at any point along the path of maximizer. Its initial value is $-\infty$. **Beta; on the other hand, is the** lowest-value found so far at any point along the path of minimizer. Its initial value of beta is $+\infty$.

Alpha-beta pruning can be applied at any depth of a tree. The algorithm removes all the nodes which do not affect the final decision. Therefore, removing such nodes speeds up processing of the algorithm. Before executing the alpha-beta pruning algorithm, remember the following points.

1. The main condition that must be met for alpha-beta pruning is: $\alpha \geq \beta$.
2. The Max player will update the value of alpha.
3. The Min player will update the value of beta.
4. While backtracking the tree, the value of the node (and not α, β) is passed to upper nodes.
5. The values of α and β are passed to the child nodes.

3.10.1 Pseudo-Code for Alpha-Beta Pruning

Example: Consider the game play tree given in **Fig. 3.28** and apply alpha-beta pruning algorithm on it.

Step 1: For node A, set $\alpha = -\infty$ and $\beta = +\infty$. Pass down these values to node B and node D. Now, all three nodes have $\alpha = -\infty$ and $\beta = +\infty$.

FIGURE 3.28 Alpha-beta pruning algorithm step 1

Step 2: Node D will play the role of maximizer and will therefore choose a value of α . The value of α at node D = 3 since $\max(2, 3) = 3$.

Step 3: Algorithm backtracks to node B which plays the role of a minimizer. So, node B will set the value of β . Recall that $\beta = +\infty$ at B. After comparing this value with the available subsequent node's value, we set $\beta = 3$ as $\min(\infty, 3) = 3$. The algorithm also passes values $\alpha = -\infty$, and $\beta = 3$ to its successor node E.

FIGURE 3.29 Alpha-beta pruning algorithm after step 3

Step 4: At node E, maximizer will update the value of alpha. The current value of alpha will be compared with 5. Since $\max(-\infty, 5) = 5$, $\alpha = 5$ and $\beta = 3$, where $\alpha \geq \beta$, so the right successor of E will be pruned, and algorithm will not traverse it, and the value at node E will be 5.

FIGURE 3.30 Alpha-beta pruning algorithm step 4

Step 5: The algorithm backtracks the tree to node A from node B. At this node, the value of alpha is updated to 3 as $\max(-\infty, 3) = 3$. Now α and β ($+\infty$) values are passed to the right successor of A which is Node C. At node C, $\alpha=3$ and $\beta=+\infty$, and these values are passed to node F.

Step 6: At node F, the value of α is compared with the value of its left child which is 0. Since, $\max(3,0) = 3$, and then with right child which is 1, and $\max(3,1) = 3$. The node value of F becomes 1.

FIGURE 3.31 Alpha-beta pruning algorithm after step 5

Step 7: The value of node F is returned to node C. At C, $\alpha=3$ and $\beta=+\infty$. Since node B plays the role of minimizer, value of beta is set to 1 as, $\min$

$(\infty, 1) = 1$. Now, the condition $\alpha \geq \beta$ is met. Therefore, the right child of C which is G is pruned. Thus, the algorithm will not compute the entire subtree G.

FIGURE 3.32 Alpha-beta pruning algorithm after step 7

Step 8: Node C returns the value of 1 to A. Since, A plays a maximizer, it selects value 3 as $\max(3, 1) = 3$.

FIGURE 3.33 The Final Tree

3.10.2 Worst-Case and Best-Case in Alpha-Beta Pruning

The efficiency of alpha-beta pruning algorithm depends on the order in which each node in the game tree is examined. This order can be of two types.

Worst Ordering

This occurs when the algorithm does not prune any of the leaf nodes. In this case, alpha beta pruning works like the minimax algorithm and consumes more time. The best move occurs on the right side of the tree and the time complexity for such an order is $O(b^m)$.

Best Ordering

The best or the ideal ordering occurs when many branches in the tree are pruned. The best moves occur at the left side of the tree. Time complexity in ideal ordering is $O(b^{m/2})$.

Thus, we see that the alpha-beta pruning is not a new algorithm, it is just an optimization of minimax algorithm. It reduces the computation time while going deep in the tree at a faster speed by cutting off branches in the game tree which need not be searched as a better move has already been found.

Example: Consider the tree given in **Fig. 3.34** and apply Alpha Beta pruning algorithms on it.

FIGURE 3.34 Tree for Alpha Beta Pruning

Step 1: Start from node A. Set $\alpha = -\infty$ and $\beta = +\infty$. Pass down these values to subsequent nodes in the tree.

Recall that node A will must choose max of B and C, so we need to first find the value of B. B is a minimizer, so have to choose min of D and E. Hence, we need to first know the value of D.

Step 2: D is a maximizer node, so its value is 5 as, $\max(-\infty, 3) = 3$ and $\max(3, 5) = 5$.

Step 3: The algorithm returns the value of D to B as now $\alpha \geq \beta$. The value of B is then set to 5 because $\min(\beta, 5) = 5$ and $\min(5, 6) = 5$.

Here, right child of node E node is not evaluated. This is because node B is a minimizer and it already knows a value that is smaller than node E, so it will not explore all the branches of E.

FIGURE 3.35 Alpha Beta Tree after Step 2

FIGURE 3.36 Alpha Beta Tree after Step 3

FIGURE 3.37 The final Alpha Beta Tree after Left Sub Tree is Traversed

Step 4: The value of B is returned to node A. At **A**, $\alpha = \max(-\infty, 5)$ which is 5. The algorithm will then find the value of node C before finalizing the α and β values for node A.

The values α, β are passed down to C, F and G. At **C**, $\alpha = 5$ and $\beta = +\infty$. To decide value of C, we need to know the value of node F.

Step 5: At **F**, $\alpha = 5$ and $\beta = +\infty$. **F** being a maximizer, sets its $\alpha = 5$ as $\max(1, 5) = 5$ and $\max(2, 5) = 5$. And the best value of F is 2. The value of F is returned to node C.

Step 6: At **C**, $\beta = \min(+\infty, 2) = 2$. Now, condition $\alpha \geq \beta$ is met and the algorithm breaks without compute the sub-tree of **G**. node C is a minimizer node and has got a value that is less than what it already has. **C** returns a value of 2 to **A**.

Step 7: A being a maximizer node sets its value as 5 because $\max(5, 2)$ which is a 5.

FIGURE 3.38 Alpha Beta Tree after Step 4

FIGURE 3.39 Alpha Beta Tree after Step 5

FIGURE 3.40 Alpha Beta Tree after Step 6

3.11 Local Search Algorithms

Local search algorithms start from a prospective solution and then move to a neighbouring solution. They can return a valid solution even if it is interrupted at any time before they end.

3.11.1 Hill-Climbing Search

It is an iterative algorithm that starts with an arbitrary solution to a problem and attempts to find a better solution by changing a single element of the solution incrementally. If the change produces a better solution, an incremental change is taken as a new solution. This process is repeated until there are no further improvements.

function Hill-Climbing (problem), returns a state that is a local maximum.

Disadvantage

This algorithm is neither complete, nor optimal.

3.11.2 Local Beam Search

In this algorithm, there are k number of states at any given time. At the start, these states are generated randomly. The successors of these k states are computed with the help of objective function. If any of these successors is the maximum value of the objective function, then the algorithm stops.

Otherwise, the (initial k states and k number of successors of the states = $2k$) states are placed in a pool. The pool is then sorted numerically. The highest k states are selected as new initial states. This process continues until a maximum value is reached.

function BeamSearch(problem, k), returns a solution state.

3.11.3 Simulated Annealing

Annealing is the process of heating and cooling a metal to change its internal structure for modifying its physical properties. When the metal cools, its new structure is seized, and the metal retains its newly obtained properties. In simulated annealing process, the temperature is kept variable.

We initially set the temperature high and then allow it to 'cool' slowly as the algorithm proceeds. When the temperature is high, the algorithm is allowed to accept worse solutions with high frequency.

Start

- Initialize $k = 0$; L = integer number of variables;
- From $i \rightarrow j$, search the performance difference Δ .
- If $\Delta \leq 0$ then accept else if $\exp(-\Delta/T(k)) > \text{random}(0,1)$ then accept;
- Repeat steps 1 and 2 for $L(k)$ steps.
- $k = k + 1$;

Repeat steps 1 through 4 till the criteria is met.

End

3.11.4 Travelling Salesman Problem

In this algorithm, the objective is to find a low-cost tour that starts from a city, visits all cities en-route exactly once and ends at the same starting city.

Start

Find out all $(n-1)!$ Possible solutions, where n is the total number of cities.

Determine the minimum cost by finding out the cost of each of these $(n-1)!$ solutions.

Finally, keep the one with the minimum cost.

end

FIGURE 3.41 Solving Travelling Salesman Problem

3.12 Single Agent Path-Finding Problems

Games such as 3X3 eight-tile, 4X4 fifteen-tile and 5X5 twenty-four tile puzzles are single-agent-path-finding challenges. They consist of a matrix of tiles with a blank tile. The player is required to arrange the tiles by sliding a tile either vertically or horizontally into a blank space with the aim of accomplishing some objective.

Other examples of single agent pathfinding problems are Travelling Salesman Problem, Rubik's Cube and Theorem Proving.

3.12.1 Search Terminology

1. **Problem space:** It is the environment in which the search takes place.
(A set of states and set of operators to change those states)
2. **Problem instance:** It is Initial state + Goal state.
3. **Problem space graph:** It represents problem state. States are shown by nodes and operators are shown by edges.
4. **Depth of a problem:** Length of a shortest path or shortest sequence of operators from Initial State to goal state.
5. **Space complexity:** The maximum number of nodes that are stored in memory.
6. **Time complexity:** The maximum number of nodes that are created.
7. **Admissibility:** A property of an algorithm to always find an optimal solution.
8. **Branching factor:** The average number of child nodes in the problem space graph.

9. **Depth:** Length of the shortest path from initial state to goal state.

3.12.2 Brute-Force Search Strategies

They are most simple, as they do not need any domain-specific knowledge. They work fine with small number of possible states.

Requirements:

- State description
- A set of valid operators
- Initial state
- Goal state description

Comparison of Various Algorithms Complexities

Let us see the performance of algorithms based on various criteria (**Table 3.3**).

TABLE 3.3



Write a Python code to implement Depth First Search Technique

OUTPUT

Write a Python code to implement Breadth First Search Technique

OUTPUT

Write a program to implement Best First Search Technique

OUTPUT

0 1 3 2 8 9

Write a program to implement minmax algorithm

OUTPUT

The optimal value is : 12

Write a Python code to implement Local Beam Search algorithm

OUTPUT



KEY TERMS

Reasoning: It is the set of processes that is used making decisions, and prediction.

Learning: It is the process of gaining knowledge or skill by studying, practising, being taught or experiencing something.

Problem solving: It is the process in which one tries to find the desired solution in the present situation by following the path which is blocked by known or unknown hurdles. Problem solving uses decision-making to select the best suitable alternative to reach the desired goal.

Perception: It is the process of acquiring, interpreting, selecting and organizing sensory information.

Rationality: It is a feature that inculcates a sense of responsibility, sensibility and judgment.

State space: It is the set of all possible states that can be attained by an agent.

Start state: It is the state from where the searching is done.

Goal test: It is a function that checks if the current state is the goal state or not.

Solution to a search problem: It is a sequence of actions (also known as plan) that transforms the start state to the goal state. This plan is realized using search algorithms.

Fringe: It is a data structure to store all the possible states (nodes) that can be reached from the current state.

State-space diagram: It is a graphical representation of the set of states that can be attained in the search algorithm with respect to the value of the objective function.

Plateau/Flat local maximum: It is a flat region in the state space where neighbouring states have the same value.



RECAPITULATION POINTS

1. Simple reflex agents choose actions based only on the current percept.
2. A learning agent learns from its past experiences. It is designed to possess learning capabilities. Initially, it starts working with basic knowledge and then acts and adapts automatically through learning.
3. Turing test is widely used to determine the success of an intelligent behaviour of a system.
4. Environment is the place where the agent would work. It gives rewards and specifies states and actions of the agents.
5. A uninformed or blind search algorithm has no additional information about the goal state other than the one provided in the problem definition. The only information they have is on how to traverse or visit the nodes in the tree.
6. Depth-limited-search (DFS) creates the same set of nodes as Breadth-First method, only in the different order.
7. In Depth-First-search (DLS), the node at the depth limit will be treated as it is a leaf node (or a node with no successor nodes).
8. In BFS algorithm, nodes in the tree are traversed breadthwise to reach the goal node. Searching begins from the root node and expands the successor node breadthwise before traversing depth-wise.
9. The uniform cost search (UCS) algorithm is used to find an optimal solution to the goal state when the step costs are not the same. In such a scenario, the algorithm computes the cumulative cost to expand each node from the root node to the goal node. It neither traverses depth nor breadth, but searches for the next node with the lowest cost.
10. The iterative deepening algorithm combines the features of DFS and BFS algorithms to find out the best depth limit that can be used to implement the DLS algorithm. IDDFS does it by gradually increasing the limit until the goal is found.
11. In bi-directional search, searching is done from both the directions simultaneously. It searches forward from initial state and backward from goal state till both meet to identify a common state.
12. The heuristic function takes the current state of the agent as its input and estimates how close the agent is from the goal. Though the heuris-

tic method may not always give the best solution, it is, however, guaranteed to find a good solution in reasonable time.

13. A* algorithm solves the searching problem efficiently by expanding less search tree to find shortest path and give an optimal result faster. It avoids expanding paths that are already expensive, but expands most promising paths first.
14. AO* Algorithm is based on problem decomposition in which a problem is broken down problem into smaller pieces.
15. Hill climbing is a heuristic search technique that works well on a large set of inputs to find a sufficiently good solution to the problem.
16. In simulated annealing, the algorithm picks a random move rather than the best move. If the random move improves the state, then it follows the same path. Otherwise, the algorithm moves downhill and chooses another path.
17. Mini-max algorithm is a recursive or backtracking algorithm which is used in decision-making and game theory.
18. Local search algorithms start from a prospective solution and then move to a neighbouring solution. They can return a valid solution even if it is interrupted at any time before they end.



ANSWER THE QUESTIONS

1. When is a machine said to be artificially intelligent?
2. Describe the components of intelligence.
3. Define learning and give different types of learning.
4. What are the different types of agents?
5. What is Turing test?
6. Explain the term 'environment' in the context of artificial intelligence.
7. Give the properties of a search algorithm.
8. When can DFS algorithm get stuck in an infinite loop? How can it be overcome?
9. Under what circumstances does the DLS algorithm terminate?
10. How does UCS work? When is it the same as BFS?
11. List the merits and demerits of UCS algorithm.

12. Give the pros and cons of A* algorithm.
13. Explain the different regions in the State Space Diagram.
14. Discuss the different types of game scenarios.
15. Explain the mini max algorithm.
16. Write short notes on:
 1. Bi-directional search
 2. Heuristic search
 3. Hill climbing algorithm
 4. Local search algorithms
17. Differentiate between the following:
 1. Inductive and deductive reasoning.
 2. Human and machine intelligence
 3. Competitive and cooperative multi-agent environment
 4. Episodic and non-episodic environment
18. Apply the specified algorithms on the graphs given below.
 1. Breadth First Search (BFS)

2. Uniform Cost Search (UCS)

3. Uniform Cost Search (UCS)

4.

5. Best First Search

Heuristic Values: A=2; B=3; C=1; D=4; S=10; G=0

6.

7. Bi-directional search

8. Breadth First Search

19. Consider the graph given below and find the cost of reaching G from S using Uniform Cost Search.

20. Compute the route to Romania, using the straight-line distance as the heuristic value. If the goal is Bucharest, and initial position is Arad, then compute the straight-line distances to Bucharest, in the graph

given below.

21. Consider the graph given below. The numbers written on the edges represent the distance between the nodes and the numbers written on nodes represent the heuristic value. Using A* Algorithm, find the most cost-effective path to reach from start state A to final state J.

22. Consider the graph given below. Find a path from A to D using A* algorithm.

23. Give the traversal of the graph given below using DLS algorithm with limit = 1

24. Identify the terms.

25. Consider the game tree given below and compute the utility value of root node using the minimax algorithm.

26. Consider the graph given below and apply AO* algorithm on it.



1. _____ intelligence is mostly used by players and dancers.
2. Two types of reasoning are _____ and _____.
3. If something is true for a particular class of things in general, it is also true for all members of that class. This happens in case of _____ reasoning.
4. _____ is a process of gaining knowledge or skill by studying, practicing, being taught or experiencing something.
5. _____ learning promotes learning by remembering sequences of events that one has witnessed or experienced.
6. A _____ agent uses bit strings as its programs and actions.
7. _____ is a feature that inculcates a sense of responsibility, sensibility and judgment.
8. _____ means a description of desirable situations.
9. _____ is widely used to determine the success of an intelligent behaviour of a system.
10. _____ is the place where the agent would work.
11. In a _____ environment, the next state of the environment can be easily determined by the current state as well as the actions of the agent.
12. In a stochastic environment, uncertainty about outcomes is quantified in terms of _____.
13. Complexity of DFS algorithm depends on the _____.
14. UCS is equivalent to _____ algorithm if the path cost of all edges is the same.
15. The _____ algorithm combines the features of DFS and BFS algorithms.
16. _____ means finding the shortest path from the current node in the graph to the goal node.
17. A _____ function uses all the available information to rank all the possible alternatives at any branching step in the search algorithm and select the best route out of possible routes.
18. The _____ diagram is a graphical representation of the set of states that can be attained in the search algorithm with respect to the value

of the objective function.

19. _____ is identified as a flat plateau that has an uphill edge.
20. In _____, the unwanted portions of a search tree that makes no difference in its final result are ignored.
21. Players take random moves and win by sheer luck or chance when playing _____ games.
22. _____ is the modified as well as optimized version of the minimax algorithm.
23. In _____ algorithm, there are k number of states at any given time. At the start, these states are generated randomly.
24. _____ is the average number of child nodes in the problem space graph.



STATE TRUE OR FALSE

1. *Interpersonal intelligence* is the ability to distinguish between one's feelings, intentions and motivations.
2. Deductive reasoning uses specific observations to make broad general statements.
3. Motor learning helps to learn by precise movement of muscles.
4. Problem solving is the process of acquiring, interpreting, selecting and organizing sensory information.
5. Performance element provides feedback to the learning element.
6. A known environment is partially observable and an unknown environment is fully observable.
7. A fully observable environment must maintain the internal state to keep track history of the world.
8. Deterministic environment descriptions maximize an agent's performance for all possible outcome of its actions.
9. Game of ludo is non-deterministic.
10. In a non-episodic environment, the current decision can affect all future decisions.
11. DFS cannot check duplicate nodes.
12. Visiting a node means exploring the adjacent nodes of the selected node.

13. Bi-directional search takes less memory capacity to store all the searches.
14. Heuristic function always finds a good solution in reasonable time.
15. All nodes that have already been expanded are placed in the OPEN list.
16. On each iteration, nodes with the highest heuristic value are expanded.
17. An admissible heuristic is optimistic in nature.
18. The AO* does not give optimal performance in all the cases.
19. Hill climbing does not support backtracking in the search space.
20. The value of the objective function should be minimized.
21. Plateau is a special type of local maximum that is higher than its neighbours but itself has a slope.
22. Tic-Tac-Toe is an example of a game with perfect information.
23. Zero-sum games are purely competitive games in which one player's gain is compensated by another player's loss.
24. The alpha-beta pruning technique cuts the number of nodes to be explored to one-fourth.
25. The max player will update the value of beta.
26. The values of α and β are passed to the upper nodes in alpha-beta pruning algorithm.
27. In alpha-beta pruning algorithm, best ordering occurs when many branches in the tree are pruned.
28. The depth of a problem is the length of the longest path from Initial State to goal state.



MULTIPLE CHOICE QUESTIONS

1. _____ *intelligence* is the ability to perceive visual or spatial information, change it and re-create visual images
 1. Bodily-kinesthetic
 2. Spatial
 3. Logical-mathematical
 4. Musical

2. _____ is the set of processes used in making decisions and prediction.
 1. Reasoning
 2. Learning
 3. Perception
 4. Problem solving
3. _____ learning triggers learning by watching and imitating others.
 1. Perceptual
 2. Relational
 3. Observational
 4. Stimulus-response
4. Identify the incorrect statement.
 1. Agent can be anything that is capable of perceiving its environment.
 2. Agents acts upon their environment using sensors.
 3. An agent must perform all the actions to maximize its performance measure with respect to the percept sequence and the knowledge base that it has.
 4. Rationality empowers the agent to perform expected actions after perceiving.
5. _____ agents work using Condition-Action Rule that maps a state to an action.
 1. Simple reflex
 2. Model-based reflex
 3. Goal-based
 4. Utility-based
6. _____ element of the learning agent provides feedback.
 1. Performance
 2. Learning
 3. Critic
 4. Problem generator
7. A _____ environment has a finite number of percepts.
 1. Continuous
 2. known
 3. observable
 4. discrete
8. _____ learning is extensively applied in an unknown environment.

1. Reinforcement
 2. Machine
 3. Deep
 4. Transfer
9. An environment with sensors can be partially observable because of ____.
1. noise
 2. inaccuracy of the sensors
 3. states are missing from the sensor data
 4. All of these
10. Crossword puzzle is an example of ____ environment.
1. Static
 2. dynamic
 3. Semi-dynamic
 4. None of these
11. A/An ____ environment has a series of one-shot actions, and only the current percept is required for the action.
1. episodic
 2. non-episodic
 3. deterministic
 4. non-deterministic
12. ____ is a data structure to store all the possible states (nodes) that can be reached from the current state.
1. tree
 2. strategy
 3. fringe
 4. path
13. Identify the incorrect statement about DFS.
1. It occupies a lot of memory space.
 2. DFS is implemented using queue data structure.
 3. Time Complexity of DFS is $O(b^m)$.
 4. Space complexity of DFS is $O(bm)$.
14. In DLS, the node at the depth limit will be treated as a ____ node.
1. parent
 2. root
 3. child
 4. leaf

15. _____ search algorithm is used in multiple real-life solutions like P2P networks, Web Crawlers and network broadcasting.
1. Breadth first
 2. Depth first
 3. Depth limited
 4. Uniform cost
16. The uniform-cost search algorithm is implemented using a _____.
1. Stack
 2. Queue
 3. Priority queue
 4. Linear array
17. Identify the incorrect statement about UCS.
1. UCS is complete only if there is no loop with zero weight.
 2. UCS is optimal only if there is no negative cost.
 3. UCS does not have any information about the location of the goal state.
 4. UCS requires minimal space for storing information about nodes.
18. _____ algorithm is used when search space is large, and depth of the goal node is not known.
1. BFS
 2. DFS
 3. UFS
 4. IDDFS
19. The value of the heuristic function is always _____.
1. zero
 2. negative
 3. positive
 4. one
20. The A* search algorithm, is the best-known form of _____ search algorithm.
1. Breadth first
 2. Best first
 3. Depth first
 4. IDDFS
21. The space complexity of A* search algorithm is _____.
1. $O(b^d)$

2. $O(bd)$
 3. $O(d^b)$
 4. $O(b+d)$
22. _____ algorithm uses AND-OR graphs /trees used for representing the solution.
1. A*
 2. Best First Search
 3. AO*
 4. IDDFS
23. _____ is a flat region in the state space where neighbouring states have the same value.
1. Plateau
 2. Ridge
 3. Shoulder
 4. Current state
24. _____ search can be used to overcome ridge problems.
1. Best first
 2. Bidirectional
 3. Depth first
 4. Breadth first
25. _____ search examines the problem which arises when an agent plans ahead of other agents that are planning against it.
1. Simulated annealing
 2. Adversarial
 3. Hill climbing
 4. Bi-directional
26. Chess is an example of _____.
1. Games with perfect information and deterministic games
 2. Games with imperfect information and deterministic games
 3. Games with perfect information and non-deterministic games
 4. Games with imperfect information and non-deterministic games
27. The mini-max algorithm performs a _____ search to explore the complete game tree.
1. Depth first
 2. Breadth first
 3. Best first

4. Adversarial

28. Values of Alpha and Beta are _____.

1. $-\infty, +\infty$
2. $-\infty, -\infty$
3. $+\infty, -\infty$
4. Not defined



Fill Up

1. Bodily-kinesthetic
2. inductive and deductive.
3. deductive
4. Learning
5. Episodic
6. Software
7. Rationality
8. Goal
9. Turing test
10. Environment
11. Deterministic
12. Probabilities
13. number of paths
14. BFS
15. iterative deepening
16. Heuristic
17. heuristic
18. state-space
19. Shoulder
20. pruning
21. Non-deterministic.
22. Alpha-beta pruning
23. Local beam search
24. Branching factor

State True or False

1. False
2. False
3. True
4. False
5. False
6. True
7. False
8. False
9. True
10. True
11. True
12. False
13. True
14. True
15. False
16. False
17. True
18. True
19. True
20. False
21. False
22. False
23. True
24. False
25. False
26. False
27. True
28. False

Multiple Choice Questions

1. b
2. a
3. c
4. b
5. a

6. c
7. d
8. a
9. d
10. b
11. a
12. c
13. b
14. d
15. a
16. c
17. d
18. d
19. c
20. b
21. a
22. c
23. a
24. b
25. b
26. a
27. a
28. a

Previous chapter

< [2. Artificial Intelligence Technologies](#)

Next chapter

[4. Knowledge Representation](#) >

4

Knowledge Representation

4.1 Introduction

Artificial intelligence as technology has always fascinated human beings. Multiple science fiction novels and movies have demonstrated the use of AI-powered systems (like robots) that can think, act, understand complex information and make smart decisions based on it.

Although this all looks like a fantastic development, we agree that it is not easy to make machines that behave exactly like humans. This is because humans have conscience and this conscience develops gradually in us with our knowledge, experiences and memories.

To build AI systems that have conscience, we need to inculcate knowledge in them. This chapter is, therefore, dedicated towards representing knowledge in machines.

Knowledge and Intelligence

Knowledge of real world plays a crucial role in intelligence and creating AI agents that demonstrate intelligent behaviour. Such an agent acts by sensing its environment and using knowledge to act intelligently. However, for this, the knowledge or experience about the input is mandatory. The relationship between knowledge and intelligence can be clearly understood from **Fig. 4.1**. You can see that if we remove the knowledge component, then the decision maker will not be able to sense the environment accurately and take appropriate decisions.

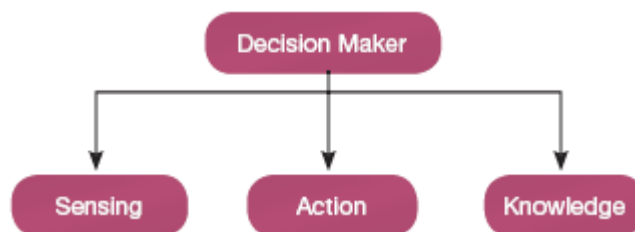


FIGURE 4.1 Relationship between Knowledge and Intelligence

4.2 Knowledge Representation

Humans are best at understanding, reasoning and interpreting things they know (knowledge). Humans use their knowledge to perform various actions in the real world. ***But how machines perform actions, comes under the domain of knowledge representation and reasoning.***

Knowledge representation and reasoning (KR, KRR) is a part of artificial intelligence which focuses on how AI agents think and how their thinking contributes to intelligent behaviour. A knowledge-based system represents information about the real world in such a way that a computer can easily understand and utilize this knowledge to solve complex real-world problems like diagnosing a medical condition or communicating with humans in natural language.

Knowledge representation represents knowledge available in the form of beliefs, intentions and judgments that an intelligent agent must possess for automated reasoning. One of the primary purposes of knowledge representation includes modelling intelligent behaviour for an agent.

Knowledge representation is not just storing data in a database. It goes beyond this aspect to facilitate an intelligent machine to learn from that knowledge and experiences so that it can behave intelligently like a human.

4.2.1 What Knowledge Needs to be Represented?

Humans have intuition, intentions, prejudices, beliefs, judgments, common sense, etc. apart from knowledge about certain facts. We need to incorporate all this information in a machine-understandable format and

make the AI system truly intelligent. For this, we need to represent the following knowledge in AI systems:

1. **Object:** Information and facts about all objects (relevant in context). For example, in a self-driving car, vehicles and roads are objects.
2. **Events:** Information and facts about actions which occur in the real world. For example, in a self-driving car, an event can be applying breaks when an object comes in front of it.
3. **Performance:** The manner in which actions are performed. It describes the behaviour related to how to do things.
4. **Meta-knowledge:** It is the knowledge about knowledge (what we know).
5. **Facts:** Facts are the truths about the real world that needs to be represented for an intelligent agent.
6. **Knowledge base:** It is the most important component of a knowledge-based agent that stores a group of sentences (technical sentences, not simple English language ones).

4.2.2 What is Knowledge?

We know that knowledge is the basic element of our brain as it helps us to know and understand the things logically and is gained by experiences of facts, data and situations. A knowledgeable person performs all the actions in a better way. Everyone has five types of knowledge illustrated in **Fig. 4.2** and enumerated as follows.

1. **Meta knowledge** is the knowledge about knowledge.
2. **Heuristic knowledge** is the knowledge about a specific topic. For example, it can be knowledge of some experts in a field or subject. Heuristic knowledge is treated as the rule of thumb as it is based on previous experiences and awareness of approaches, which are good to

work but not guaranteed.

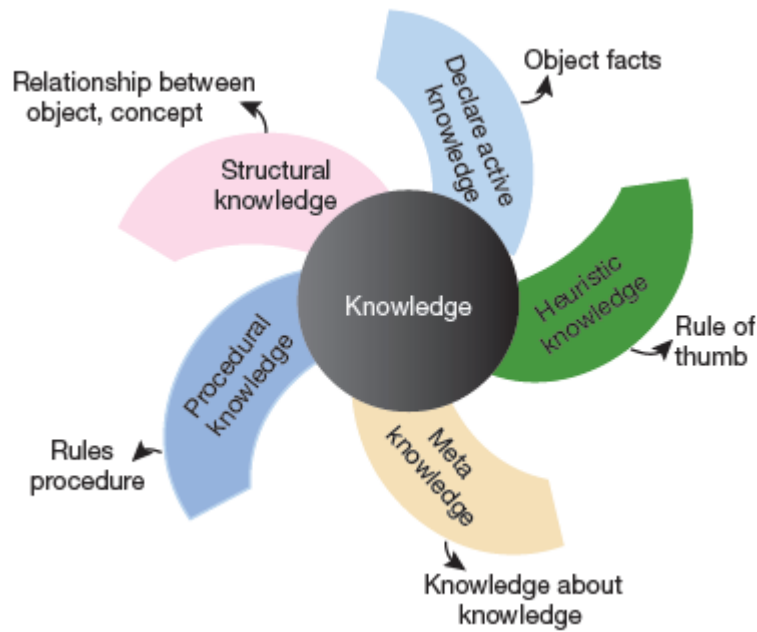


FIGURE 4.2 Types of knowledge

3. **Procedural knowledge**, also known as imperative knowledge, gives information about how to achieve or do something. This knowledge includes rules, strategies, procedures, agendas, etc. that can be directly applied to perform any task.
4. **Declarative knowledge** is the information that we have about an object. This knowledge helps us to describe a particular concept, fact, object and its attributes. Declarative knowledge is simpler than procedural language and is also called descriptive knowledge as it is usually represented using declarative sentences.
5. **Structural knowledge** is the basic knowledge to solve complex problems. It describes relationships between various concepts or objects such as kind of, part of, and grouping of something.

4.2.3 What is Logic?

Logic is the main component of any knowledge as it facilitates a person to draw conclusion by filtering required information from large volumes of sentences.

In AI, knowledge is represented using logic. Every logic has three main elements.

Syntax

Every language has its own syntax to specify the sequence of constructs that makes a complete sentence in that language. It is therefore not wrong to say that syntax is the representation of a language.

Semantics

Semantics is used to check if a syntactically correct sentence is logically correct and meaningful or not. Therefore, semantics defines the sense of the sentence which relates to the real world. For example, consider the statement:

Ram is riding a bike.

This sentence is syntactically as well as semantically correct. However, the sentence,

Bike is riding Ram.

is syntactically correct but semantically incorrect.

Logical Inference

Inference means to deduce conclusions in the context of some fact or problem. Correspondingly, logical inference uses inference algorithms to think all the possible reasons that can give a proper result.

4.2.4 Cycle of Knowledge Representation in AI

An AI system has several components (**Fig. 4.3**) that helps it to exhibit an intelligent behaviour. These components include the following:

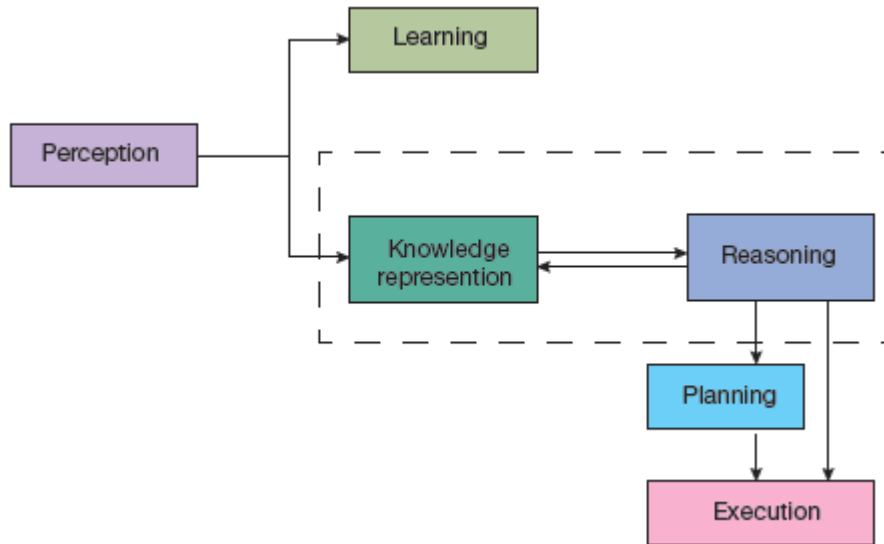


FIGURE 4.3 Components of an AI system

Perception

The perception component retrieves data from the environment, discovers the source(s) of noise, checks if the AI was damaged by anything and defines response to be given when any sense has been detected. Data from the environment is gathered using different types of sensors in the form of video, audio, text, time, temperature or any other sensor-based input.

Learning

The learning component learns from the data that is captured by the perception component. The aim here is to develop a system that can be taught instead of being programmed. With learning, the system focuses on self-improvement through knowledge acquisition, inference, acquisition of heuristics, faster searches, etc.

Knowledge Representation and Reasoning

This is the main component which shows human-like intelligence in the machines. Knowledge representation techniques (refer **Fig. 4.4**) are designed to understand using a top-down approach focussing in detail, what an agent needs to know to behave intelligently. The KRR component also defines how automated reasoning procedures can make this knowledge available as needed.



FIGURE 4.4 Forms of Knowledge Representation

Planning and Execution

This component analyses knowledge representation and reasoning. The planning component selects an initial state, enumerates preconditions and effects, and a sequence of actions to achieve the goal state. Once planning is done, actions are executed to get the desired results.

4.2.5 Knowledge Representation Requirements

A good knowledge representation system must have the following properties.

1. Representational accuracy, to ensure that the system represents all kinds of required knowledge.
2. Inferential adequacy to manipulate the representational structures to produce new knowledge corresponding to the existing structure.
3. Inferential efficiency to direct the inferential knowledge mechanism to generate appropriate results.
4. Acquisitional efficiency that helps the knowledge representation system to easily acquire new knowledge using automatic methods.

4.3 Knowledge-Based Agent

In AI, the agents that mimic a human being's knowledge are known as knowledge-based intelligent agents. Such an agent uses its knowledge and reasoning to act efficiently by making appropriate decisions. A knowledge-based agent, therefore, does the following:

1. It maintains an internal state of knowledge to represent states, actions, etc.
2. It deduces reasoning with that knowledge.
3. It updates their knowledge after observations by incorporating new precepts.
4. It takes actions accordingly.

Knowledge-based agents represent knowledge that can be reasoned to act intelligently according to the requirements. Usually, these agents store knowledge about its surroundings in the form of sentences.

4.3.1 The Architecture of Knowledge-Based Agent

Knowledge-based agents represent the world with some formal representation to act intelligently. For this, they contain a knowledge-base and an inference system.

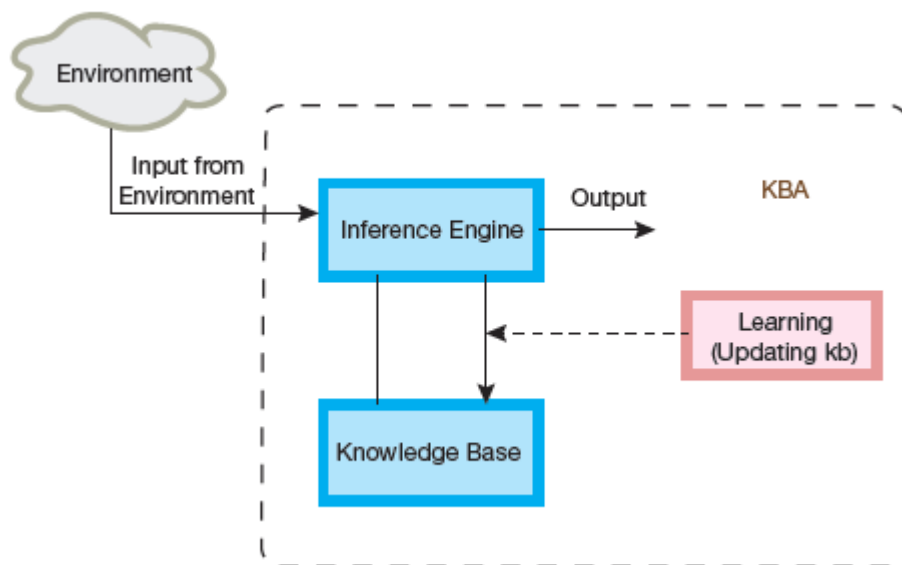


FIGURE 4.5 Architecture of Knowledge-Based Agent

Knowledge base (KB) is the most important component of a knowledge-based agent. It stores real facts of world using sentences which are explained in knowledge representation language.

Inference engine (IE) is the knowledge-based system engine that infers new knowledge in the system.

Figure 4.5 shows a generalized architecture for a knowledge-based agent (KBA). According to **Fig. 4.5**,

Step 1: The KBA perceives the environment to take input from it.

Step 2: This input is taken by the inference engine of KBA.

Step 3: The inference engine interacts with the knowledge base to make decisions based on the knowledge stored in it.

Note that the learning element of the KBA regularly updates the knowledge base by learning new knowledge. This knowledge base is nothing but a collection of sentences (technical information not same as a sentence in English) that are expressed in a language known as knowledge representation language and stores fact about the world.

In AI, a sentence is not an English language sentence, but a fact represented in a language known as knowledge.

The inference system derives new sentences from old ones and also facilitates addition of new sentence(s) to the knowledge base. As stated earlier, a sentence is a proposition about the world. New information is deduced from the knowledge base by applying logical rules (either forward chaining or backward chaining).

4.3.2 Operations Performed By KBA

Whenever the KBA is called to add/update knowledge in the KB or to answer queries made to the KB, it acts intelligently by performing a set of important functions that are given below.

TELL operation tells the knowledge base what knowledge it already has about its environment and what additional knowledge it needs to know. TELL is also used to inform the knowledge base which action was selected and performed by the agent.

ASK operation asks the knowledge base what action it should perform. The agent gets its answer from the knowledge base.

PERFORM operation performs the selected action.

4.3.3 A Generic Knowledge-Based Agent

Given below is the structure outline of a generic knowledge-based agent program (**Fig. 4.6**).

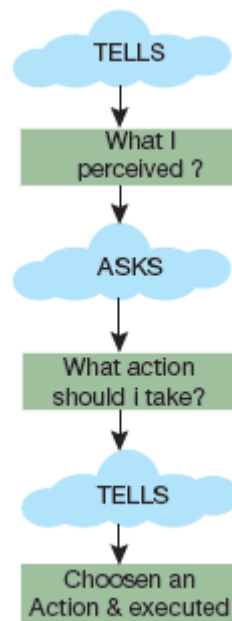


FIGURE 4.6 Mechanism of an agent program

```

function KB-AGENT(percept):
  persistent: KB, a knowledge base
             t : time ( counter initially 0)
  TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
  Action = ASK(KB, MAKE-ACTION-QUERY(t))
  TELL(KB, MAKE-ACTION-SENTENCE(action, t))
  t = t + 1
  return action
  
```

The knowledge-based agent accepts the information it perceives from the environment as input and returns an action as the output. The agent maintains the knowledge base (KB). It has some background knowledge of the real world and maintains a counter, t with initial value zero to indicate the time for the whole process.

Each time the function is called, the KBA performs operations as given below.

TELLS the KB what it perceives. For this, the MAKE-PERCEPT-SENTENCE function is used to generate a sentence that informs about what information the agent had perceived at the given time.

ASKS KB what action it should take. The MAKE-ACTION-QUERY function is executed to generate a sentence asking which action should be performed at the current time.

TELLS the KB about the action that was chosen. The MAKE-ACTION-SENTENCE function generates a sentence informing about the chosen action that was executed.

4.3.4 Various Levels of Knowledge-Based Agent

A knowledge-based agent can be viewed at different levels as follows.

Knowledge Level

This is the first level of a KBA. In this level, we need to specify what the agent knows, and what its goals would be. This information is then used to fix the behaviour of the agent. For example, if an agent has to go from point A to point B and it knows the optimum way to reach the destination, then this information is added at the knowledge level (using sentences).

Logical Level

At this level, the knowledge representation of the knowledge stored in the knowledge base is understood. For this, sentences are encoded into different logics. Therefore, knowledge is encoded into logical sentences. Continuing with the above example, agent deduces (understands) logic to reach destination, that is, point B.

Implementation Level

This is the physical representation of logic and knowledge. At this level, the agent performs actions according to the knowledge obtained from the

logical and knowledge level. For example, the agent that wants to move from point A to B, implement the knowledge and logic to reach the destination.

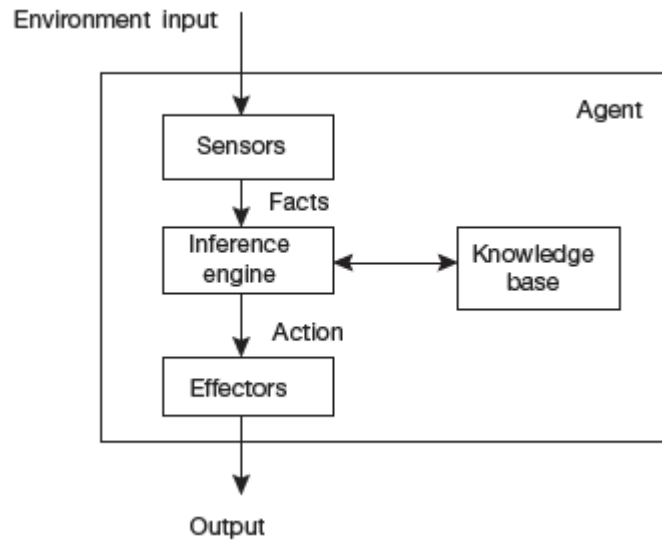


FIGURE 4.7 Adjusting AC temperature

Example: In an automated air conditioner, the inbuilt knowledge stored in the system states that ‘It would adjust its temperature according to the weather.’ Now, when to adjust and how to adjust is logic. The actual working of adjusting the temperature is at the implementation level of the knowledge-based agent.

4.3.5 Approaches to Designing a Knowledge-Based Agent

There are mainly two approaches to build a knowledge-based agent.

Declarative Approach

In this approach, the knowledge-based agent is created by initializing it with an empty knowledge base and then telling the agent all the sentences with which we want to start with.

These sentences are told to the empty system one by one until the system becomes knowledgeable enough to deal with the environment.

Procedural Approach

In the procedural approach, the desired behaviour is directly encoded in the agent as a program code. Therefore, in this approach, we are actually specifying the desired behaviour of the system through coding in programming languages like LISP, Prolog.

Practically, a **hybrid approach** is used that combines features of both declarative as well as procedural approaches. In the hybrid approach, declarative knowledge is compiled into more efficient procedural code.

4.4 Types of Knowledge

We can express different types of knowledge in a knowledge representation system.

Name	Age	Course
Nrip	19	BCA
Krish	18	BBA
Riya	17	BTech

FIGURE 4.8 A simple relation/ table of students

Simple Relational Knowledge

In this technique, facts about a set of objects are systematically stored using relations (or tables) in database systems (refer **Fig. 4.8**). These relations depict relationship between different entities. However, the main shortcoming of this approach is that there is little opportunity for inference.

Inheritable knowledge Inheritable knowledge is one that stores data using a hierarchy of classes. We start representing using generalized classes and move down to specialized classes in the hierarchy. For example, consider **Fig. 4.9** which shows such a relationship. Note that Zinya and Vidhi both are instances of under-graduate and post-graduate students, respectively. Inheritable knowledge shows the relation between instance and

class, and is therefore known as instance relation of IS-A relation. In this diagram, objects and values are represented in boxed nodes.

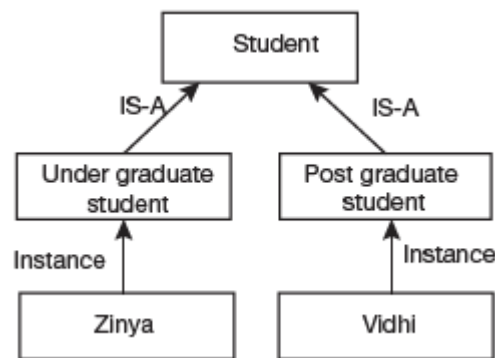


FIGURE 4.9 Inferential Knowledge Representation

Inferential Knowledge

Inferential Knowledge Represents Knowledge in the Form of Formal Logic and Can be Used to Derive More Facts Accurately. For example,

Statement 1: Diya is a student.

Statement 2: All students are bright.

Then, it can be represented as;

Student(Diya)

$\forall x = \text{Student}(x) \longrightarrow \text{Bright}(x)$

4.5 Techniques of Knowledge Representation in AI

Different techniques of representing knowledge can be given as follows.

4.5.1 Logical Representation

Logical representation is a language with some definite rules. These rules state the proposition without any sort of ambiguity. Rules specify the conclusion based on different conditions. The syntax and semantics of the

language guide the process of converting sentences into logics. While syntax of a language is used to decide how legal sentences in logic can be constructed, semantics, on the other hand, are the rules which helps to interpret (deduce the meaning of) sentences in the logic.

Logical representation can be of two types.

Propositional logic, also known as propositional calculus or statement logic, works in a Boolean, that is, true or false method.

First-order logic (FoL) is an advanced version of propositional logic that represents the objects in quantifiers and predicates.

Language semantics are used to represent this information. The downside of this method is that it is tough to work with as it is highly logical, not very natural and less efficient at times. **Table 4.1** summarizes the advantages and disadvantages of Logic representation techniques

TABLE 4.1 Pros and Cons of Logic Representation

Advantages	Disadvantages
Logic representation is the foundation of any programming language. Logical representation helps to perform logical reasoning.	Logical representations have some restrictions and are challenging to work with. This technique may not be very natural, and inference may not be very efficient.

4.5.2 Semantic Network Representation

Semantic network is used as an alternative of logic representation. It represents knowledge through graphical networks. In a graphical network (refer **Fig. 4.10**), nodes represent objects and arcs describe the relationship between those objects. These objects can be categorized in different forms and can be linked to other objects through two types of relations—IS-A relation (inheritance) and kind-of-relation (instance). **Table 4.2** highlights the advantages and disadvantages of Semantic Network Representation.

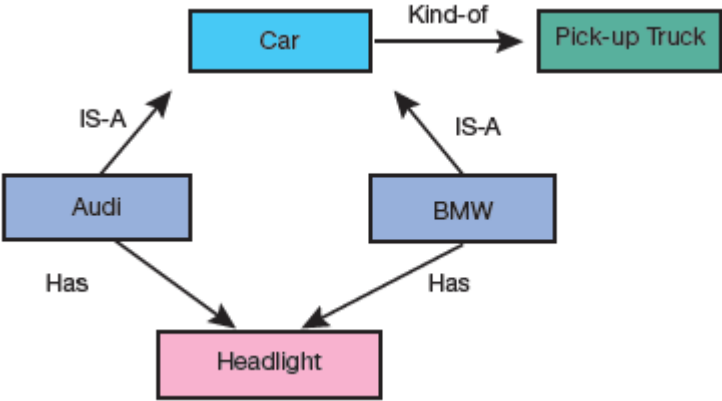


FIGURE 4.10

TABLE 4.2 Pros and Cons of Semantic Networks

Advantages	Disadvantages
• Semantic networks are a natural representation of knowledge. • They convey meaning in a transparent manner. • These networks are simple and easy to understand.	• Semantic networks take more computational time at runtime. • They are inadequate as they do not have any equivalent quantifiers. • These networks are not intelligent and may vary depending on its developer. • Semantic networks do not have any standard definition for the link names. • It is impossible to represent all facts from human’s memory into a semantic network.

4.5.3 Frame Representation

A frame is a record-like structure that describes a real world using a collection of attributes and values. A frame also known as slot-filter knowledge representation is derived from semantic networks. These frames at a later stage can be designed as classes and objects using any object-oriented programming language. A single frame is not of much use, so, a collection of inter-connected frames is used. These frames store knowledge about an object or event.

Slot	Filter
Title	Artificial Intelligence
Area	Computer Science
Author	Reema Thareja
Publisher	Pearson
Edition	First
Publication Year	2023
Price	Rs 500
Page Count	700

Frame representation is widely used in various applications including natural language processing and machine visions. Basically, a frame contains slots that, in turn, can store values of any type and size. Every slot has a name and value which we call a facet.

Facets are features of frames which are used to put constraints on the frames. For example, IF-NEEDED facts are called when data of any particular slot is needed. A frame can have any number of slots, a slot may include any number of facets and facets may have multiple values. **Table 4.3** lists the advantages and disadvantages of Frame Representation techniques

Example: Design a frame for a book

TABLE 4.3 Pros and Cons of Frame Representation

Advantages	Disadvantages
• It is easy to program as it groups together the related data. • It is simple to understand and visualize. • The programmer can easily add slots for new attributes and relations. • It is easy to add default data and search for missing values.	• In frame system inference, data processing is not a straight forward task • The inference mechanism finds it difficult to smoothly process data available in frame representation. • It has a very generalized approach.

4.5.4 Production Rules

Production rules system consist of (condition, action) pairs which mean, 'If condition then action'. Using this rule, the agent checks for the condition. If the condition exists, the production rule becomes applicable and the corresponding action is performed.

If we have multiple if conditions or if-else statements in a program, the production rule can also be designed to determine which rule may be applied to a problem. The action part specifies steps that must be performed to solve a problem. This complete process is called a recognize-act cycle.

A production rule system has mainly three parts as sown in **Fig. 4.11**:

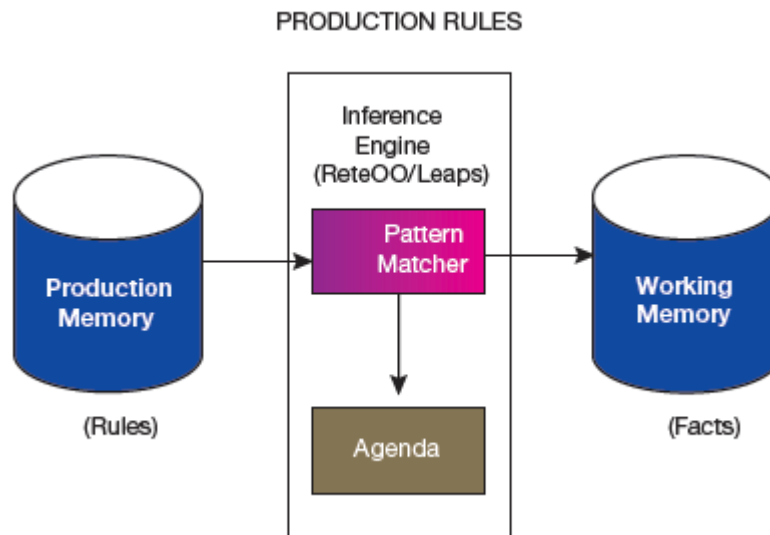


FIGURE 4.11 Components of a Production Rule System

1. The set of production rules
2. Working memory
3. The recognize-act-cycle

The working memory contains the description of the current state of problem-solving.

Rules are used to express knowledge about the working memory. This knowledge may fire other rules.

At times, it happens that multiple production rules are fired together. This situation is called conflict set. In such a situation, the agent needs to select one particular rule. This is often referred to as conflict resolution.

For example, consider the rules given below.

- IF (hungry) THEN action (eat)
- IF (on the bus AND bought ticket AND vacant seat) THEN action (sit down).

TABLE 4.4

Advantages	Disadvantages
• Production rules are expressed in natural language. • They are highly flexible as they can be easily modified.	• It does not have any learning capabilities • It does not store the result of the problem for future uses. • During program execution, multiple rules may be active. • Production rules are inefficient.

4.5.5 Propositional Logic (PL)

Propositional logic is the simplest form of logic in which all the statements are made using propositions. A proposition is a declarative statement which is either true or false but it cannot be both (true as well as false). Knowledge in a proposition is represented in logical as well as mathematical form.

Example:

- a) Today is a holiday. (True or False)
- b) The Sun rises from West (False)
- c) 1 + 1 = 11 (False)
- d) 19 is a prime number. (True)

We can see from the above examples that propositional logic provides better clarity on data and information, especially in an incomplete environment by deeper analysis and inference of the limited information pre-

sented to it. Propositional logic is, therefore, widely adopted in advanced search in AI in solving complex problems.

The following points must be noted.

1. We can express a propositional logic using logical operators.
2. A proposition formula which is always true is called tautology, and is also known as a valid sentence.
3. A proposition formula which is always false is called contradiction.
4. Statements which are questions, commands or opinions are not propositions. For example, 'Which course are you opting for?', 'Read the newspaper', 'I think it's all fiction', are not propositions.
5. An atomic proposition is said to be **satisfiable** if there is an interpretation for which it is true. In **Table 4.6**, $P \vee Q$ is satisfiable for first three propositional formula (statements).
6. A **contingent** statement is one which is neither a tautology nor a contradiction. For example, a proposition like 'I am not well, so I am taking leave today'. This statement is often neither correct nor incorrect. As we may be having a little dizziness, our main aim behind taking leave is to complete another work on a priority basis.

Proposition means sentences. Propositional logic applies the Boolean logic to convert real-world information into a format that is readable and understandable by the computer.

4.6 Syntax of Propositional Logic

The syntax of propositional logic defines the allowable sentences for the knowledge representation. Two types of propositions include the following:

1. Atomic proposition: Atomic propositions are simple propositions that are made of a single proposition symbol. Such propositions can be either true or false.

Example:

1. $2+2$ is 4, is an atomic proposition that is true.
2. The Sun rises in the west is an atomic proposition that is false.

2. Compound proposition: Compound propositions are formed by combining simpler atomic propositions, using parenthesis and logical operators.

Example:

1. The Sun rises in the east and sets in the west.
2. Ruskin Bond prolifically authored inspiring children's books and was awarded the Sahitya Akademi Award to honour his work of literature.

4.7 Logical Connectives in Propositional Logic

Logical connectives are used to form compound propositions by connecting two simpler propositions. Some important and widely used connectives are given as follows:

Negation

A sentence such as $\neg P$ is called negation of P. A literal can be either positive literal or negative literal.

Conjunction

A sentence which uses 'and' to connect two propositions is said to be connected using a conjunction connective. It is represented using a $\wedge$ connective. Two propositions combined using a conjunction connective can be written as, $P \wedge Q$.

For example, the sentence 'I like to study Mathematics and Computer' can be written as:

P= I like to study Mathematics.

Q= I like to study Computer.

I like to study Mathematics and Computer $\rightarrow P \wedge Q$.

Disjunction

A sentence which uses 'or' to connect two propositions is said to be connected using a disjunction connective. It is represented using a $\vee$ connective. Two propositions combined using a conjunction connective can be written as $P \vee Q$.

For example, the sentence 'I would like to read a book authored by R.K. Narayanan or by Ruskin Bond' can be written as:

P = I would like to read a book authored by R.K. Narayanan

Q = I would like to read a book authored by Ruskin Bond

I would like to read a book authored by R.K. Narayanan or by Ruskin Bond $\rightarrow P \vee Q$.

Implication

A sentence P implies Q can be written as $P \rightarrow Q$. This is known as implication or as an if-then rule. For **example**: If you are hungry, then eat some sprouts. Let P = If you are hungry, and Q = Eat some sprouts, can be represented as $P \rightarrow Q$

Biconditional

A sentence of the form $P \Leftrightarrow Q$ is called a biconditional sentence as P implies Q and Q implies P . For **example**: If you are a consistent learner, then you will never stop learning.

P = You are a consistent learner

Q = You will never stop learning

These two statements can be represented as $P \Leftrightarrow Q$.

Table 4.5 shows different propositional logic connectives.

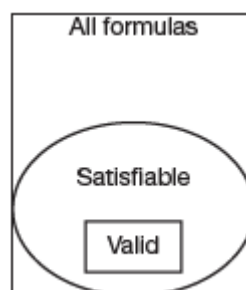
TABLE 4.5 Propositional logic connectives

Connective symbols	Word	Technical term	Example
$\wedge$	AND	Conjunction	$A \wedge B$
$\vee$	OR	Disjunction	$A \vee B$
$\rightarrow$	Implies	Implication	$A \rightarrow B$
$\Leftrightarrow$	If and only if	Biconditional	$A \Leftrightarrow B$
$\neg$ or $\sim$	Not	Negation	$\neg A$ or $\neg B$

4.7.1 Logical Equivalence

Two propositions are said to be logically equivalent if and only if the columns in the truth table are identical to each other. For example, given two propositions P and Q , column for $\neg P \vee Q$ and $P \rightarrow Q$, are identical hence logically equivalent. In addition, note that,

P is valid iff $\neg P$ is unsatisfiable. And, P is satisfiable iff $\neg P$ is not valid. We can verify this from **Fig. 4.12**.

**FIGURE 4.12** Logical Equivalence

A satisfiable sentence has at least one interpretation under which the sentence can evaluate to be true.

A sentence is valid if it is true in all interpretations. This means that its negation is not satisfiable.

4.7.2 Truth Table of Propositional Connectives

The following truth table (**Table 4.6**) shows propositional logic, showing all the possible combinations with logical connectives in a tabular format.

Note that the truth table follows Boolean logic and propositional calculus.

TABLE 4.6 Propositional Connectives**Negation**

P	$\sim P$
TRUE	FALSE
FALSE	TRUE

Conjunction

P	Q	$P \wedge Q$
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

Disjunction

P	Q	$P \vee Q$
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

Implication

P	Q	$\sim P$	$P \wedge Q$	$P \rightarrow Q$ can be understood as, $\{\sim P \vee (P \wedge Q)\}$
FALSE	FALSE	TRUE	FALSE	TRUE
FALSE	TRUE	TRUE	FALSE	TRUE
TRUE	FALSE	FALSE	FALSE	FALSE
TRUE	TRUE	FALSE	TRUE	TRUE

Logical Equivalence

P	Q	$\sim P$	$\sim P \vee Q$	$P \rightarrow Q$
FALSE	FALSE	TRUE	TRUE	TRUE
FALSE	TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	FALSE	FALSE
TRUE	TRUE	FALSE	TRUE	TRUE

Bi-directional implication

P	Q	$P \Leftrightarrow Q$
FALSE	FALSE	TRUE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

4.7.3 Precedence of Connectives

We have an operator precedence chart in any programming language. Similarly, there is a precedence order for propositional connectors (shown in **Table 4.7**). This order is strictly followed while evaluating a propositional problem.

TABLE 4.7 Propositional Connectors

Precedence	Operators
First precedence	Parenthesis
Second precedence	Negation
Third precedence	Conjunction (AND)
Fourth precedence	Disjunction (OR)
Fifth precedence	Implication
Sixth precedence	Biconditional

4.7.4 Properties of Operators

Commutativity

$P \wedge Q = Q \wedge P$, or $P \vee Q = Q \vee P$.

Associativity

$(P \wedge Q) \wedge R = P \wedge (Q \wedge R)$, or $(P \vee Q) \vee R = P \vee (Q \vee R)$

Identity Element

$P \wedge \text{True} = P$ or $P \vee \text{True} = \text{True}$

Distributive

$P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$ or $P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$.

DE Morgan's Law

$$\neg (P \wedge Q) = (\neg P) \vee (\neg Q) \text{ or } \neg (P \vee Q) = (\neg P) \wedge (\neg Q).$$

Double-Negation Elimination

$$\neg (\neg P) = P.$$

4.7.5 Limitations of Propositional Logic

Not all relationships can be represented using propositional logic. For example, relations like ALL, some or none cannot be represented with propositional logic. For example,

All students are studious

Some students are outstanding.

Statements which cannot be answered absolutely are called open statements. It is not possible to describe such statements in terms of their properties or logical relationships using propositional logic. For example, 'It is so easy' is an open statement as it may be simple for some but difficult for others.

4.8 Inference Rules

We know that AI is being used to create intelligent computers that can generate create new logic from old ones or by evidence. This process of generating conclusions from evidence and facts is known as inference.

Inference rules are the templates for generating valid arguments that are used to derive proofs to obtain conclusions that finally lead to the desired goal. Some important terminology frequently used in inference rules include the following.

Implication

Logical connectives are represented as $P \rightarrow Q$. It is a Boolean expression.

Converse

The converse of implication, which means that the right-hand side proposition goes to the left-hand side and vice-versa can be written as $Q \rightarrow P$.

Contrapositive

The negation of converse represented as $\neg Q \rightarrow \neg P$, is known as contrapositive.

Inverse

The negation of implication is called inverse. It can be represented as $\neg P \rightarrow \neg Q$.

The truth table given below (**Table 4.8**) illustrates these relationships. We can see that $P \rightarrow Q$ is equivalent to $\neg Q \rightarrow \neg P$, and $Q \rightarrow P$ is equivalent to $\neg P \rightarrow \neg Q$.

TABLE 4.8 Inverse Relations

P	Q	$P \rightarrow Q$	$Q \rightarrow P$	$\neg Q \rightarrow \neg P$	$\neg P \rightarrow \neg Q$
T	T	T	T	T	T
T	F	F	T	F	T
F	T	T	F	T	F
F	F	T	T	T	T

4.8.1 Types of Inference Rules

Modus Ponens: The Modus Ponens rule is an important inference rule that states if P is True and $P \rightarrow Q$ is True, then we can infer that Q will also be true. This can be represented as,

$$\text{Notation for modus ponens: } \frac{P \rightarrow Q, P}{\therefore Q}$$

Example:

Statement-1: ‘If I am studying thoroughly then I have a test next day.’ $\Rightarrow$

$P \rightarrow Q$

Statement-2: ‘I am studying thoroughly’ $\Rightarrow P$

Conclusion: ‘I have a test next day.’ $\Rightarrow Q$.

In the last row, $P \rightarrow Q$ is true, P is true and therefore, Q is also true (**Fig. 4.13**).

P	Q	$\sim P$	$P \rightarrow Q (\sim P \vee Q)$
FALSE	FALSE	TRUE	TRUE
FALSE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	FALSE
TRUE	TRUE	FALSE	TRUE

FIGURE 4.13 Truth Table for Modus Ponens

Hence, we can say that, if $P \rightarrow Q$ is true and P is true then Q will also be true. Figure proves this inference rule with the help of a truth table.

Modus Tollens: The Modus Tollens rule states that if $P \rightarrow Q$ is True and $\sim Q$ is True, then $\sim P$ will also true. It can be written as,

$$\text{Notation for Modus Tollens: } \frac{P \rightarrow Q, \sim Q}{\sim P}$$

P	Q	$\sim P$	$\sim Q$	$P \rightarrow Q (\sim P \vee Q)$
FALSE	FALSE	TRUE	TRUE	TRUE
FALSE	TRUE	TRUE	FALSE	TRUE
TRUE	FALSE	FALSE	TRUE	FALSE
TRUE	TRUE	FALSE	FALSE	TRUE

In the last row, $P \rightarrow Q$ is true, $\sim Q$ is true indicates that $\sim P$ is also true.

Statement-1: ‘If I am studying thoroughly then I have a test next day.’ $\Rightarrow$

$P \rightarrow Q$

Statement-2: 'I am not having a test tomorrow' $\Rightarrow \sim Q$

Statement-3: We can conclude from the above two statements that 'I am not studying thoroughly' $\Rightarrow \sim P$

Hypothetical Syllogism: The Hypothetical Syllogism rule states that $P \rightarrow R$ is True iff $P \rightarrow Q$ is true, and $Q \rightarrow R$ is true.

Example:

Statement-1: If you study hard then you will get good knowledge. $P \rightarrow Q$

Statement-2: If you have good knowledge then you can get recruited easily. $Q \rightarrow R$

Conclusion: If you study hard then you can get recruited easily. $P \rightarrow R$

This rule can be proved using a truth table (**Table 4.9**) as shown below.

TABLE 4.9 Truth Table for Hypothetical Syllogism

P	$\sim P$	Q	$\sim Q$	R	$\sim R$	$P \rightarrow Q$ ($\sim P \vee Q$)	$Q \rightarrow R$ ($\sim Q \vee R$)	$P \rightarrow R$ ($\sim P \vee R$)
FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE
TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE

Notation of disjunctive syllogism: $\frac{P \vee Q, \sim P}{Q}$

Disjunctive Syllogism: The Disjunctive syllogism rule states that if $P \vee Q$ is True, and $\sim P$ is True, then Q will be true. It can be expressed as,

Example:

Statement 1: Today I will study AI or Java. $\implies P \vee Q$

Statement 2: Today I will not study AI. $\implies \neg P$

Conclusion: I will Study Java. $\implies Q$

TABLE 4.10 Truth Table for Disjunctive Syllogism

P	$\neg P$	Q	$P \vee Q$
FALSE	TRUE	FALSE	FALSE
FALSE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
TRUE	FALSE	TRUE	TRUE

Statement-1: Today is Sunday or Monday.

Statement-2: Today is not Sunday.

Conclusion: Today is Monday.

This rule is also proved using truth table given (**Table 4.10**).

Addition: The Addition rule is a common inference rule that states, if P is true, then $P \vee Q$ will be true.

$$\text{Notation of addition: } \frac{P}{P \vee Q}$$

Example:

Statement: I would like to study Mathematics. $\implies P$

Statement-2: I would like to study Artificial Intelligence. $\implies Q$

Conclusion: I would like to study Mathematics or Artificial Intelligence.
 $\implies (P \vee Q)$

Simplification: The simplification rule state that if $P \wedge Q$ is true, then both Q as well as P will also be true. It can be represented as:

$$\text{Notation of simplification rule: } \frac{P \wedge Q}{Q} \text{ or } \frac{P \wedge Q}{P}$$

Resolution: The Resolution rule state that if $P \vee Q$ and $\neg P \wedge R$ is true, then $Q \vee R$ will be true. It can be represented as

$$\text{Notation of resolution: } \frac{P \vee Q, \neg Q \wedge R}{P \vee R}$$

Proof by Truth table (**Table 4.11**):

TABLE 4.11 Truth Table for Resolution

P	Q	$\neg Q$	R	$P \vee Q$	$\neg Q \vee R$	$P \vee R$
FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE
FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE
FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE
FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE

4.8.2 First-Order Logic

In the previous section, we learnt how propositional logic can be used to represent statements. Although it is a very powerful technique, it suffers from a major shortcoming. It can represent the facts, which are either true or false. Complex sentences or natural language statements cannot be represented using propositional logic. Thus, it has a very limited expressive power. For example, consider the statements given below. These statements cannot be expressed in propositional logic.

‘Some boys are hard-working’, or

‘Sania likes badminton.’

We need a more powerful logic, such as first-order logic (also known as predicate logic) to represent such statements. First-order logic (FOL) is another way of knowledge representation that can be thought of as an extension to propositional logic. It can easily represent natural language statements in a concise way.

Like natural language, first-order logic assumes that the world contains facts (like propositional logic) as well as the following things in the world:

Objects

People, alphabets, numbers, colours, theories, squares, pits, wumpus, etc.

Relations

A relation can be a simple unary relation such as red, round, is adjacent, or an n-ary relation like the sister of, brother of, has colour, comes between.

Function

Mother of, best friend, second match of, end of,.....

Moreover, like natural language, first-order logic also has two main parts—syntax and semantics.

4.8.2.1 Syntax of First-Order Logic

The syntax of FOL determines the set of symbols that can be used in a logical expression in first-order logic. These symbols, which are the basic elements of a FOL sentence, comprise the following.

Atomic Sentences

Atomic sentences are the most basic sentences of FOL. They are formed using a predicate followed by a parenthesis with a sequence of terms. The general syntax of writing an atomic sentence in FOL can be given as,

Figure 4.14

Predicate (term1, term2,....., term n).

For example,

Ria and Krisha are sisters can be expressed in FOL as Sisters(Ria, Krisha).

Bruno is a dog. $\Rightarrow$ dog(Bruno).

Constant	1, -10, F, Myra, Delhi, cow,....
Variables	x, y, z, a, b,....
Predicates	Father, Enjoys, Reading,....
Function	sqrt, LeftLegOf,....
Connectives	$\wedge$, $\vee$, $\neg$, $\Rightarrow$, $\Leftrightarrow$
Equality	$=$
Quantifier	$\forall$, $\exists$

FIGURE 4.14 Basic elements of a FOL sentence

Complex Sentences

Complex sentences, as the name suggests, are formed by combining atomic sentences using connectives. A complex sentence in FOL has two main parts—subject and predicate.

Subject is the main part of the statement. It specifies what (or whom) the sentence is about.

Predicate is a relation which binds two atoms together in a statement. It tells something about the subject.

Consider the statement: ‘Ahana is a good student.’. It consists of two parts—the first part ‘Ahana’ is the subject and the second part ‘is a good student’ is the predicate.

Quantifiers

A quantifier is an element that generates quantification. Quantification specifies the quantity of specimen in the universe of discourse. In simple words, it identifies the range and scope of the variable in the logical expression. There are two types of quantifiers: universal quantifier, (for all, everyone, everything) and existential quantifier, (for some, at least one).

Universal quantifier: Universal quantifier is a symbol of logical representation written as $\forall$, which specifies that the statement (within its range) is true for everything or every instance of a particular thing. That is, if x is a variable, then $\forall x$ is read as- For all x , For each x , For every x .

In universal quantifier, implication ($\rightarrow$) is used as the main connective.

For **example:** Every student who works hard succeed can be represented as,

$\forall x \text{ student}(x) \rightarrow \text{work\_hard}(x, \text{succeed})$.

Existential quantifier: Existential quantifiers, denoted by the logical operator $\exists$, are used to express that the statement within its scope is true for at least one instance of something. In simple terms, if x is a variable, then $\exists x$ or $\exists(x)$ will be read as,

- There exists a 'x.'
- For some 'x.'
- For at least one 'x.'

In existential quantifier, AND or Conjunction symbol ($\wedge$) is used as the main connective.

For example, the statement, 'Some students are brilliant', can be written as,

$\exists x: \text{students}(x) \wedge \text{brilliant}(x)$

4.8.2.2 Properties of Quantifiers

Before writing predicate rules, always keep the following properties of quantifiers in your mind.

1. In universal quantifier, $\forall x \forall y$ is similar to $\forall y \forall x$.
2. In existential quantifier, $\exists x \exists y$ is similar to $\exists y \exists x$.
3. However, $\exists x \forall y$ is not similar to $\forall y \exists x$.

Let us look at some more examples of FOL using quantifiers.

1. Every child cries.

Here, the predicate is 'cries(child).' Since every child cries, it can be represented using the universal quantifier by writing,

$\forall x \text{ child}(x) \rightarrow \text{cry}(x)$.

2. Every student writes exams.

Here, the predicate is 'write(x, y),' where x = student, and y= exam.

Again, we will use the universal quantifier to represent this fact.

$\forall x \text{ student}(x) \rightarrow \text{writes}(x, \text{exam})$.

3. Some students have participated in the Olympiad.

Here, the predicate is 'participate(x, y)' where x= students, and y= Olympiad. Since only few and not every student has participated, we will represent this statement using existential quantifier by writing,

$\exists x \text{ students}(x) \rightarrow \text{participate}(x, \text{Olympiad})$.

4. Not every student likes both Mathematics and Computers.

Here, the predicate is 'like(x, y),' where x= student, and y= subject (Mathematics, Computer).

Note that here, we have used every but with negation. Therefore, this statement can be written as,

$\neg \forall (x) [\text{student}(x) \rightarrow \text{like}(x, \text{Mathematics}) \wedge \text{like}(x, \text{Computers})]$.

5. Only one student participated in the Olympiad.

Here, the predicate is 'participated (x, y),' where x= student, and y= Olympiad.

Now that only one student participated in Olympiad, we will use existential quantifier and write,

$\exists (x) [\text{student}(x) \rightarrow \text{participated}(x, \text{Olympiad}) \wedge \forall (y) [\neg (x=y) \wedge \text{student}(y) \rightarrow \neg \text{participated}(y, \text{Olympiad})]]$.

4.8.2.3 Free and Bound Variables

The quantifiers usually interact with variables. These variables are of two types.

Free variable: A variable in a formula is said to be a free variable if it occurs outside the scope of the quantifier. For example,

$$\forall x \exists (y)[P(x, y, z)]$$

Here, neither existential nor universal quantifier is applicable on variable z . hence, it is a free variable.

Bound variable: On the contrary, a bound variable in a formula occurs within the scope of the quantifier. For example,

$$\forall x [A(x) B(y)]$$

Here, x and y are the bound variables.

4.8.2.4 Inference in First-Order Logic

Inference in FOL is used to deduce new facts or sentences from existing ones. To understand the FOL inference rule, let us learn about some basic terminologies that will be frequently used in FOL.

Substitution

Substitution is a fundamental operation that can be easily performed on terms and formulas to derive a new inference. If we write $F[a/x]$, then it means replace constant value ' a ' with variable ' x '.

The substitution is complex in the presence of quantifiers in FOL.

Equality

FOL uses equality symbols to specify that the two terms refer to the same object. For example,

Teacher (AI) = Raman

The above sentence indicates that the object referred by the Teacher (AI) is similar to the object referred to by Raman. The equality symbol can also be used with negation to represent those two terms are not the same objects. For example, we can write $\neg(x=y)$ to represent $x \neq y$.

4.8.2.5 FOL Inference Rules for Quantifier

Like propositional logic, inference rules in FOL also follows some basic rules such as universal generalization, universal instantiation, existential instantiation and existential introduction.

Universal Generalization

Universal generalization states that for any arbitrary element c in the universe of discourse, if $P(c)$ is true, then we can conclude that

$$\frac{P(c)}{\forall x P(x)}$$

This formula indicates that every element has a similar property. Hence, x must not appear as a free variable.

For example, if $P(c)$: 'A nibble contains 4 bits'. We can use this information to give a universal generalization as,

$\forall x P(x)$ 'All nibbles contain 4 bits'

Universal Instantiation

Universal instantiation, also known as universal elimination or UI, is an inference rule that can be applied multiple times to add new sentences in the knowledge base. The new or the updated knowledge base is logically equivalent to the previous one.

According to UI, we can infer any sentence $P(c)$ by substituting a ground term, c (a constant within domain x) for the variable. Therefore,

$\forall x P(x)$ for any object in the universe of discourse, we can write

$$\frac{\forall x P(x)}{P(c)}$$

Example: IF ‘Every person reads books’ $\Rightarrow \forall x P(x)$, we can infer that

‘Navya reads books’ $\Rightarrow P(c)$.

Example: ‘All students who scored 90+ are outstanding.’ This fact can be added in the knowledge base by writing,

$$\forall x \text{ student}(x) \wedge \text{scored\_90+}(x) \rightarrow \text{Outstanding}(x)$$

Now, this information can be used to infer any of the following statements using UI.

- $\text{Student}(\text{Manya}) \wedge \text{Scored\_90+}(\text{Manya}) \rightarrow \text{Outstanding}(\text{Manya})$,
- $\text{Student}(\text{Shrey}) \wedge \text{Scored\_90+}(\text{Shrey}) \rightarrow \text{Outstanding}(\text{Shrey})$,
- $\text{Student}(\text{Sister}(\text{Prince})) \wedge \text{Scored\_90+}(\text{Sister}(\text{Prince})) \rightarrow \text{Outstanding}(\text{Sister}(\text{Prince}))$

Existential Instantiation

Existential instantiation or existential elimination is an inference rule in FOL that can be applied only once to replace the existential sentence.

The new knowledge base is not logically equivalent to the previous one, but it will be satisfiable if the older version of the knowledge base was. The existential instantiation inference rule states that $P(c)$ can be inferred from the formula given in the form of $\exists x P(x)$ for a new constant symbol c if and only if c is a new term for which $P(c)$ is true. It can be represented as:

$$\frac{\exists x P(x)}{P(c)}$$

For example, from the given sentence, $\exists x \text{ Winner}(x) \wedge \text{CashPrize}(x, 1000)$, we can infer: $\text{Winner}(K) \wedge \text{CashPrize}(K, 1000)$, only if K does not

appear in the knowledge base. Here, K is a constant symbol and is known as **Skolem constant**.

Existential Introduction

Existential introduction, also known as existential generalization, is an inference rule in FOL which states that if there is some element c in the universe of discourse having a property P , then we can infer that there exists something in the universe which has the property P . It can be represented as:

$$\frac{P(c)}{\exists x P(x)}$$

The existential instantiation is a special case of **Skolemization process**.

Example: If we say that Aman won a cash prize of Rs 1000, then we can infer that there is someone who won the cash prize of Rs 1000.

4.8.3 Generalized Modus Ponens Rule

In FOL, we have a single inference rule known as generalized modus ponens that is a lifted version of modus ponens. The generalized modus ponens states that if P implies Q and P is asserted to be true, then Q must be true.

In the context of atomic sentences p_i, p_i', q , Modus Ponens states that if there is a substitution θ such that $\text{SUBST}(\theta, p_i') = \text{SUBST}(\theta, p_i)$, then it can be represented as:

$$\frac{p_1', p_2', \dots, p_n', (p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q)}{\text{SUBST}(\theta, q)}$$

For example, let us use this rule to find some x such that x is student, and x is scoring above 90 and infer that x is outstanding.

p_1' is Student(Myra) p_1 is Student(x)

p_2' is Scored_90+(y) p_2 is Scored_90+(x)

θ is $\{x/\text{Myra}, y/\text{Myra}\}$ q is Outstanding(x)

SUBST(θ, q).

4.8.4 Unification in FOL

The unification process is used to make two different logical atomic expressions (or literals) identical by finding a substitution.

Let Ψ_1 and Ψ_2 be two atomic sentences and σ be a unifier such that $\Psi_1\sigma = \Psi_2\sigma$, then it can be expressed as UNIFY(Ψ_1, Ψ_2).

For example, to unify $\{\text{Student}(x), \text{Student}(\text{Jerry})\}$, we will write

$$\Psi_1 = \text{Student}(x), \Psi_2 = \text{Student}(\text{Jerry}),$$

Now, using the substitution process, we will substitute Jerry for x .
Therefore,

$\theta = \{\text{Jerry}/x\}$ is a unifier for these atoms. Using the substitution process, both expressions are found to be identical.

The substitution variables are called most general unifier or MGU.

4.8.5 Unification Algorithm

Unification is a very important part of first-order inference algorithms. The UNIFY algorithm unifies two atomic sentences and returns a unifier for those sentences, if one exists. However, if the two sentences are not identical, the UNIFY algorithm returns failure, indicating that the two sentences do not match.

Example: Given two expressions, $P(x, y)$, and $P(a, f(z))$, find out if they are identical to each other.

Let us perform the substitution process. 579

$$\begin{array}{ll} P(x, y) & \text{(i)} \\ P(a, f(z)) & \text{(ii)} \end{array}$$

In equation (i), substitute x with a , and y with $f(z)$. This can be written in FOL substitution as a/x and $f(z)/y$.

With both the substitutions, equation (i) will be identical to the second expression- $P(a, f(z))$.

Hence, the two expressions are identical when the substitution set is: $[a/x, f(z)/y]$.

Although unification seems simple and straight forward, we need to satisfy some basic conditions for this process, described below.

1. Predicate symbol must be the same. Expressions with different predicates cannot be unified.
2. Number of arguments in both the expressions must be identical.
3. Unification works only when different variables are used. Thus, the unification process will fail if two similar variables are present in the same expression.

Figure 4.15 gives the algorithm for unification.

```

Algorithm: Unify( $\Psi_1, \Psi_2$ )
Step 1: If  $\Psi_1$  or  $\Psi_2$  is a variable or constant, then:
    a) If  $\Psi_1$  or  $\Psi_2$  are identical, then return NIL.
    b) Else if  $\Psi_1$  is a variable,
        a. then if  $\Psi_1$  occurs in  $\Psi_2$ , then return FAILURE
        b. Else return  $\{(\Psi_2 / \Psi_1)\}$ .
    c) Else if  $\Psi_2$  is a variable,
        a. If  $\Psi_2$  occurs in  $\Psi_1$  then return FAILURE,
        b. Else return  $\{(\Psi_1 / \Psi_2)\}$ .
    d) Else return FAILURE.
Step 2: If the initial Predicate symbol in  $\Psi_1$  and  $\Psi_2$  are not same,
then return FAILURE.
Step 3: IF  $\Psi_1$  and  $\Psi_2$  have a different number of arguments, then
return FAILURE.
Step 4: Set Substitution set(SUBST) to NIL.
Step 5: For i=1 to the number of elements in  $\Psi_1$ .
    a) Call Unify function with the ith element of  $\Psi_1$  and ith
    element of  $\Psi_2$ , and put the result into S.
    b) If S = failure then returns Failure
    c) If S  $\neq$  NIL then do,
        a. Apply S to the remainder of both L1 and L2.
        b. SUBST= APPENDS, SUBST).
Step 6: Return SUBST.

```

FIGURE 4.15 Algorithm for unification

Note that in the algorithm, we have initially set the substitution set to be empty. Then, with each pass, every atomic sentence is unified, if the following conditions are applicable.

1. There is an identical expression match.
2. If one expression is a variable v_i , and the other is a term t_i which does not contain variable v_i , then,
 1. Substitute t_i / v_i in the existing substitutions.
 2. Add t_i / v_i to the substitution setlist.
3. If both the expressions are functions, then check if the function name and the number of arguments are the same in both the expressions.

Example: Find the MGU of $\{p(f(a), g(Y))$ and $p(X, X)\}$

Let $\Psi_1 = p(f(a), g(Y))$ and $\Psi_2 = p(X, X)$

SUBST $\theta = \{f(a) / X\}$

We get $\Psi_1 = p(f(a), g(Y))$ and $\Psi_2 = p(f(a), f(a))$.

Since $f(a)$ is not same as $g(Y)$, unification failed.

Example: Find the MGU of $\{p(b, X, f(g(Z)))$ and $p(Z, f(Y), f(Y))\}$

Let $\Psi_1 = p(b, X, f(g(Z)))$ and $\Psi_2 = p(Z, f(Y), f(Y))$

SUBST $\theta = \{b/Z\}$

We get $\{p(b, X, f(g(b)))$ and $p(b, f(Y), f(Y))\}$.

Now, SUBST $\theta = \{f(Y) / X\}$

We get, $\{p(b, f(Y), f(g(b)))$ and $p(b, f(Y), f(Y))\}$

SUBST $\theta = \{g(b) / Y\}$

We finally get,

$\{p(b, f(g(b)), f(g(b)))$ and $p(b, f(g(b)), f(g(b)))\}$

Hence, unification process is successfully done and the unifier = $\{ b/Z, f(Y) / X, g(b) / Y\}$.

Example: Find the MGU of $\{p(X, X)$, and $p(Z, f(Z))\}$

Let, $\Psi_1 = \{p(X, X)$, and $\Psi_2 = p(Z, f(Z))$

SUBST $\theta = \{X/Z\}$

We get, $\{p(Z, Z)$ and $p(Z, f(Z))\}$

Since the two expressions are not identical, unification for these expressions is not possible.

Example: Find the MGU of UNIFY(prime(11), prime(y))

Let, $\Psi_1 = \{\text{prime}(11)\}$, and $\Psi_2 = \{\text{prime}(y)\}$

SUBST $\theta = \{11/y\}$

We get, $\text{prime}(11)$ and $\text{prime}(11)$

Hence, successfully unified and Unifier is $\{11/y\}$.

Example: Find the MGU of $Q(a, g(x, a), f(y))$, $Q(a, g(f(b), a), x)$.

Let $\Psi_1 = Q(a, g(x, a), f(y))$, and $\Psi_2 = Q(a, g(f(b), a), x)$

SUBST $\theta = \{f(b)/x\}$

We get $Q(a, g(f(b), a), f(y))$ and $Q(a, g(f(b), a), f(b))$

Now, SUBST $\theta = \{b/y\}$

Finally, $Q(a, g(f(b), a), f(b))$ and $Q(a, g(f(b), a), f(b))$

Hence, successfully unified and the Unifier is $[a/a, f(b)/x, b/y]$.

Example: UNIFY($\text{knows}(\text{Richard}, x)$, $\text{knows}(\text{Richard}, \text{John})$)

Let $\Psi_1 = \text{knows}(\text{Richard}, x)$, and $\Psi_2 = \text{knows}(\text{Richard}, \text{John})$

SUBST $\theta = \{\text{John}/x\}$

We get, $\text{knows}(\text{Richard}, \text{John})$ and $\text{knows}(\text{Richard}, \text{John})$.

Hence, successfully unified with unifier as $\{\text{John}/x\}$.

4.8.6 Resolution

Resolution is a theorem proving technique invented by mathematician John Alan Robinson in 1965. The technique stresses on building refutation proofs, that is, proofs by contradictions. Resolution is usually used when there are multiple statements and we have to prove a conclusion.

Unification is an important part in proving by resolutions. Resolution derives a single inference rule which can efficiently operate on the conjunctive normal form (CNF) or clausal form. Here,

Clause is a disjunction of literals or an atomic sentence.

Conjunctive normal form is a sentence represented as a conjunction of clauses.

Resolution technique is used in both propositional as well as first-order predicate logic in different ways for proving the satisfiability of a sentence. For this, resolution applies **Proof by Refutation** in which the knowledge base and the negated goal is used to obtain a null clause. The null clause indicates contradiction. While the knowledge base itself is consistent, the contradiction is of course introduced by a negated goal thereby indicating that the original goal is true. Hence, the name, proof by refutation.

4.8.7 The Resolution Inference Rule

A resolution can resolve two clauses if they contain complementary literals. The underlying assumption here is that the literals are standardized and share no variables with each other.

$$\frac{l_1 \vee \dots \vee l_k, \quad m_1 \vee \dots \vee m_n}{\text{SUBST}(\theta, l_1 \vee \dots \vee l_{i-1} \vee l_{i+1} \vee \dots \vee l_k \vee m_1 \vee \dots \vee m_{j-1} \vee m_{j+1} \vee \dots \vee m_n)}$$

where l_i and m_j are complementary literals.

This rule is also called the binary resolution rule because it only resolves exactly two literals. In simple terms, to show that a formula X is valid, we show that $\neg X$ is unsatisfiable. That is, given a consistent set of axioms, A and a goal B , to show $A \models B$, we show that $(A \wedge \neg B)$ is unsatisfiable.

Example: Resolve the two clauses given below:

[Animal ($g(x) \vee \text{Loves}(f(x), x)$)] and [$\neg \text{Loves}(a, b) \vee \neg \text{Kills}(a, b)$]

Here, we have two complementary literals: $\text{Loves}(f(x), x)$ and $\neg \text{Loves}(a, b)$

We can perform unification of these literals with unifier $\theta = [a/f(x), b/x]$ to generate a resolvent clause:

[Animal (g(x) $\vee$ $\neg$ Kills(f(x), x))]

4.8.8 The Resolution Process

Given below are steps that need to be performed in resolution process.

Step 1: Convert facts into first-order logic statements.

Step 2: Convert FOL statements into CNF.

Step 3: Negate the statement which needs to be proved as we will prove by contradiction

Step 4: Draw the resolution graph (unification).

Example:

- Mannat likes all kind of food.
- Mango and LadyFinger are food
- Anything anyone eats and not killed is food.
- Krish eats almonds and still alive
- Reyansh eats everything that Krish eats.

Now, prove by resolution that Mannat likes almonds

Step-1: Convert these facts into FOL statements.

$\forall x: \text{food}(x) \rightarrow \text{likes}(\text{Mannat}, x)$
 $\text{food}(\text{mango}) \wedge \text{food}(\text{LadyFinger})$
 $\forall x \forall y: \text{eats}(x, y) \wedge \neg \text{killed}(x) \rightarrow \text{food}(y)$
 $\text{eats}(\text{Krish}, \text{almonds}) \wedge \text{alive}(\text{Krish})$
 $\forall x \text{eats}(\text{Krish}, x) \rightarrow \text{eats}(\text{Reyansh}, x)$
 $\forall x: \neg \text{killed}(x) \rightarrow \text{alive}(x)$
 $\forall x: \text{alive}(x) \rightarrow \neg \text{killed}(x)$
 $\text{likes}(\text{Mannat}, \text{almonds})$

$\updownarrow$ Newly added predicates as they
 will be required for the purpose

Step-2: Conversion of FOL into CNF as CNF makes easier for resolution proofs.

For this, first eliminate all implication. This is done by replacing $\rightarrow$ rewriting the expression. This means, replace

$a \rightarrow b$ with $\neg a \vee b$

If there is a biconditional implication as, $a \leftrightarrow b$, then write $a \rightarrow b$ and $b \rightarrow$

a

Therefore, we now get,

$\forall x \neg \text{food}(x) \vee \text{likes}(\text{Mannat}, x)$

$\text{food}(\text{mango}) \wedge \text{food}(\text{LadyFinger})$

$\forall x \forall y \neg [\text{eats}(x, y) \wedge \neg \text{killed}(x)] \vee \text{food}(y)$

$\text{eats}(\text{Krish}, \text{almonds}) \wedge \text{alive}(\text{Krish})$

$\forall x \neg \text{eats}(\text{Krish}, x) \vee \text{eats}(\text{Reyansh}, x)$

$\forall x \neg [\neg \text{killed}(x)] \vee \text{alive}(x)$

$\forall x \neg \text{alive}(x) \vee \neg \text{killed}(x)$

$\text{likes}(\text{Mannat}, \text{almonds})$

Now, move negation ($\neg$) inwards and rewrite the expressions. For example,

$\neg(\forall x p)$ will be written as $\exists x \neg p$

$\neg(\exists x p)$ will be written as $\forall x \neg p$

$\neg(a \vee b)$ will be written as $\neg a \wedge \neg b$

$\neg(a \wedge b)$ will be written as $\neg a \vee \neg b$

$\neg\neg a$ will be written as a

Therefore, we get,

$\forall x \neg \text{food}(x) \vee \text{likes}(\text{Mannat}, x)$

$\text{food}(\text{mango}) \wedge \text{food}(\text{LadyFinger})$

$\forall x \forall y \neg \text{eats}(x, y) \vee \text{killed}(x) \vee \text{food}(y)$

$\text{eats}(\text{Krish}, \text{almomds}) \wedge \text{alive}(\text{Krish})$

$\forall x \neg \text{eats}(\text{Krish}, x) \vee \text{eats}(\text{Reyansh}, x)$

$\forall x \text{killed}(x) \vee \text{alive}(x)$

$\forall x \neg \text{alive}(x) \vee \neg \text{killed}(x)$

$\text{likes}(\text{Mannat}, \text{almonds})$

Rename variables or standardize variables to ensure that every statement has a unique variable name. We should not repeat 'x' or any other variable in any subsequent statement. This is important to make unification step easier and unambiguous. Therefore, we will write,

$\forall x \neg \text{food}(x) \vee \text{likes}(\text{Mannat}, x)$

$\text{food}(\text{mango}) \wedge \text{food}(\text{LadyFinger})$

$\forall u \forall v \neg \text{eats}(u, v) \vee \text{killed}(u) \vee \text{food}(v)$

$\text{eats}(\text{Krish}, \text{almomds}) \wedge \text{alive}(\text{Krish})$

$\forall x \neg \text{eats}(\text{Krish}, w) \vee \text{eats}(\text{Reyansh}, w)$

$\forall x \text{killed}(z) \vee \text{alive}(z)$

$\forall x \neg \text{alive}(g) \vee \neg \text{killed}(g)$

$\text{likes}(\text{Mannat}, \text{almonds})$

Eliminate existential instantiation quantifier ($\exists$) by replacing it with skolem constant. This process is known as **Skolemization**. For example,

$\exists y \text{Intelligent}(y)$ is now written as $\text{Intelligent}(C_1)$

Here, C_1 is the skolem constant.

In our example problem, since there is no existential quantifier, all the statements will remain same in this step.

Drop Universal quantifiers ($\forall$) and rewrite the statements.

$\neg \text{food}(x) \vee \text{likes}(\text{Mannat}, x)$

$\text{food}(\text{mango}) \wedge \text{food}(\text{LadyFinger})$

$\neg \text{eats}(u, v) \vee \text{killed}(u) \vee \text{food}(v)$

$\text{eats}(\text{Krish}, \text{almonds}) \wedge \text{alive}(\text{Krish})$

$\neg \text{eats}(\text{Krish}, w) \vee \text{eats}(\text{Reyansh}, w)$

$\text{killed}(z) \vee \text{alive}(z)$

$\neg \text{alive}(g) \vee \neg \text{killed}(g)$

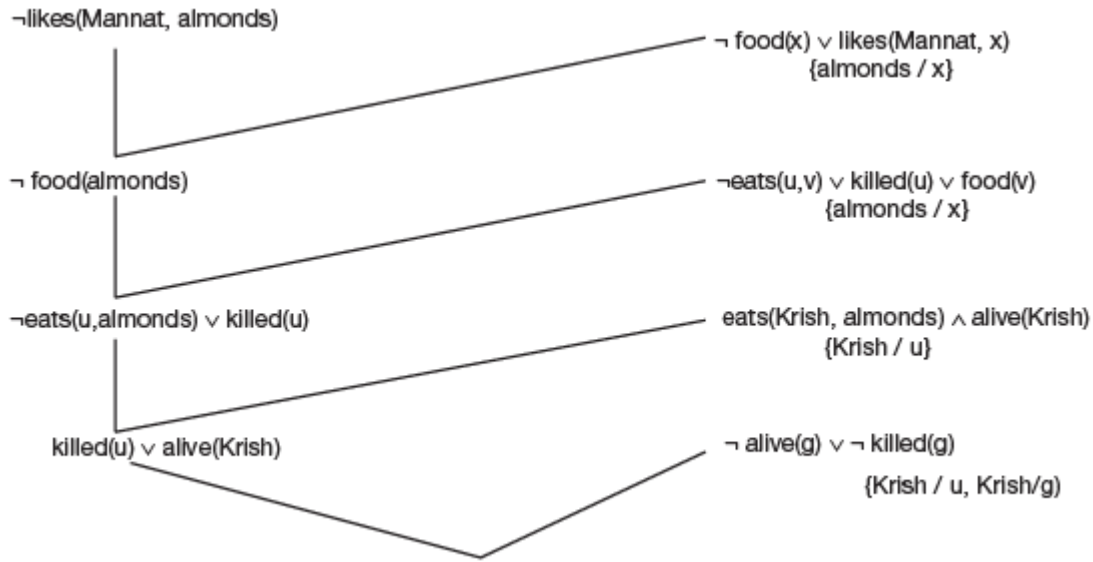
$\text{likes}(\text{Mannat}, \text{almonds})$

Distribute conjunction $\wedge$ over disjunction $\vee$. This step will not make any change in this problem.

Step-3: Negate the statement to be proved. So we will write

$\neg \text{likes}(\text{Mannat}, \text{almonds})$

Step-4: Draw Resolution graph using substitution.



To construct the resolution graph, note that on the LHS we would write the statement that is the negation of what has to be proved.

On the RHS, we will write predicate that has the same predicate as that on the LHS. Since we have likes on the LHS, we choose to write a predicate 'likes' on the RHS. Moreover, we will substitute variable x with almonds. Now, eliminate likes and not likes so that $\neg \text{food}(\text{almonds})$ is left.

Repeat the process. Find a clause having the predicate food to eliminate the common ones. Going like this, we will come to a point when no predicates would be left. An empty set indicates the negation of the conclusion has been proved as a complete contradiction with the given set of statements.

4.8.9 Explanation of Resolution Graph

In the first step of a resolution graph, $\neg \text{likes}(\text{Mannat}, \text{almonds})$ and $\text{likes}(\text{Mannat}, x)$ get resolved or cancelled by substitution of $\{\text{almonds}/x\}$. We are left with $\neg \text{food}(\text{almonds})$.

In the second step of the resolution graph, $\neg \text{food}(\text{almonds})$ and $\text{food}(v)$ get resolved or cancelled by substitution of $\{\text{almonds}/v\}$. We are left with $\neg \text{eats}(u, \text{almonds}) \vee \text{killed}(y)$.

In the third step of the resolution graph, $\neg \text{eats}(u, \text{almonds})$ and $\text{eats}(\text{Krish}, \text{almonds})$ gets resolved by substitution $\{\text{Krish}/u\}$. We are left

with killed(Krish) and alive (**Krish**).

In the fourth step of the resolution graph, both the predicates get resolved after substituting Krish for the variables. Now, no predicates are left to be resolved.

4.9 Forward Chaining and Backward Chaining in AI

We know that inference engine is the component of the intelligent system that applies logical rules to the knowledge base to infer new information from the known facts. The first inference engine was used in an expert system and it mainly uses two modes—**forward chaining and backward chaining**.

Artificial intelligence (AI) system uses both forward as well as backward chaining to perform tasks like planning, classification, reasoning, etc. The importance of these two techniques is immense and hence there is a need to understand their basics.

4.9.1 Horn Clause and Definite Clause

Horn clause and definite clause are the forms of sentences, which facilitates knowledge base to use a more restricted and efficient inference algorithm. These logical inference algorithms apply forward and backward chaining approaches. But before applying these approaches, the knowledge base must be in the form of the **first-order definite clause**.

Definite Clause

A clause which is a disjunction of literals with exactly one positive literal is known as a definite clause or strict horn clause.

Horn Clause

A clause which is a disjunction of literals with at most one positive literal is known as horn clause. Hence, all the definite clauses are horn clauses.

Example: $(\neg p \vee \neg q \vee k)$. It has only one positive literal k.

It is equivalent to $p \wedge q \rightarrow k$.

4.9.2 Forward Chaining

Forward chaining also known as a forward deduction or forward reasoning and starts with atomic sentences in the knowledge base and applies inference rules (Modus Ponens) in the forward direction to extract more data until a goal is reached. In simpler terms, forward-chaining algorithm starts from known facts, applies rules to deduce new facts and then adds those new facts to the known facts. This process is repeated until the problem is solved (or the goal state is attained).

Forward chaining is a popular implementation strategy used for improving and developing expert systems and modelling the human brain in AI.

Forward chaining approach is also said to be data-driven approach as we reach the goal using available data. It works using bottom-up approach, as it moves from bottom to the top of the decision-making tree. Forward chaining uses PROLOG language to deal with inference rules.

4.9.3 Forward Chaining Algorithm

As discussed, a forward-chaining algorithm starts with known facts and proceeds to trigger all the inference rules whose premises are satisfied. The new data derived from the existing one is then added to the known facts. This process is repeated till the goal is achieved or the problem is solved.

The inference engine plays an important role here. It scans the inference rules till it finds one for which the antecedent (If clause) is known to be true. Once such a rule is found, the inference engine can deduce or conclude the consequent (**Fig 4.16**) (Then clause). This results in the addition of new data.

The forward-chaining algorithm uses the following step by step process to complete a task and reach the logical or accurate conclusion.

Step 1: The system is fed with information about the problem.

Step 2: This information is stored in the working memory in the form of rules.

Step 3: The inference engine scans the rules present in the working memory in a predefined sequence to find a rule that matches contents.

Step 4: The system evaluates the rules to identify new matches.

Step 5: This process is repeated until no further matches are found.

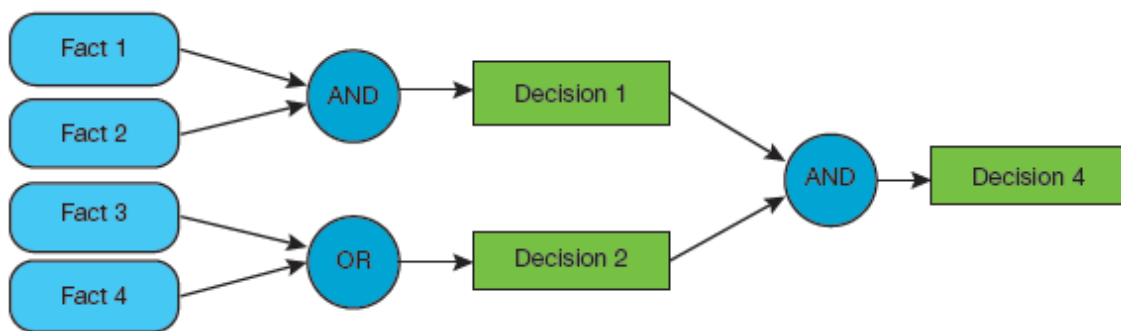


FIGURE 4.16 Forward Chaining

Example: Use the facts given below to prove that ‘Raza is criminal.’

‘As per the law, it is a crime for an Indian to sell weapons to enemy nations. ABC is an enemy of India and has some missiles, and all the missiles were sold to it by Raza, who is an Indian.’

The first step is to convert all the facts into first-order definite clauses.

Facts Conversion into FOL

It is a crime for an Indian to sell weapons to hostile nations.

Using variables p, q and r, we can write,

Indian (x) $\wedge$ (i)
weapon(y) $\wedge$
sells (x,y,z) $\wedge$
enemy(p,
India) $\rightarrow$
Criminal(x)

ABC is an
enemy of
India.

This can be (ii)
written as,
enemy(ABC,
India)

ABC has
some
missiles.

Owns(ABC, (iii)
p) $\wedge$
Missile(p)

All of the
missiles
were sold to
ABC by Raza.

$\forall p$ (iv)
Missiles(p) $\wedge$
Owns (ABC,
p) $\rightarrow$ Sells
(Raza, p,
ABC)

Missiles are
weapons.

Missile(p) → (v)
Weapons (p)

Raza is
Indian.

Indian(Raza). (vi)

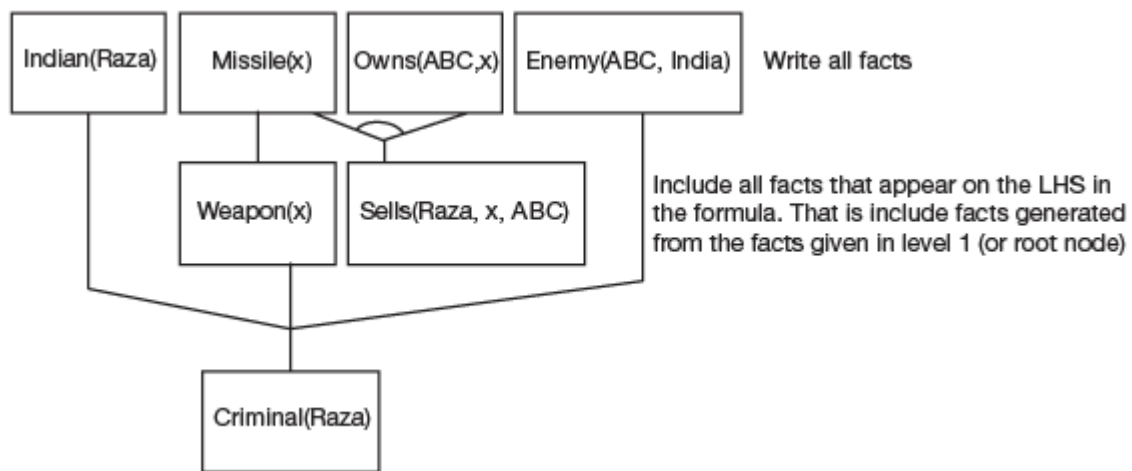


FIGURE 4.17 Forward chaining proof

Hence from **Fig. 4.17**, it is proved that Raza is a criminal using forward chaining approach.

4.9.4 Why Do We Use Forward Chaining?

Forward chaining strategy is used when the final outcome is not known, but there are facts available about the domain in which the AI system functions. Facts available in the knowledge base are used to reach the goal state or the final outcome.

For this reason, forward chaining is generally used for planning, monitoring, controlling and interpreting applications. It is an implementation strategy that is very widely used in expert systems, business and production rule systems.

Advantages of Forward Chaining

1. It is more adaptable than backward chaining as there is no restriction on the information that can be obtained from it.
2. It gives a decent premise to come to conclusions.
3. There are no limits on the data that can be derived from it. This makes it more flexible than backward chaining.
4. It can be used to draw multiple conclusions

Disadvantages of Forward Chaining

1. It takes time to synchronize and eliminate accessible information.
2. It is slower than backward chaining.
3. It generates new information without understanding what is important and relevant to the problem solution.
4. It requires lot of data as input.
5. Much of the data stored in the knowledge base may not even be relevant to the final conclusion.
6. The hypothesis performed by the forward chaining is at times not clear.
7. It might give different conclusions, which may result in ambiguity.
8. Forward chaining is good to be used on problems that have a single starting point and multiple possible endpoints. It is an inefficient algorithm when used on problems that have several starting points and just a single endpoint.

4.9.5 Backward Chaining

Backward chaining, also known as a backward deduction or backward reasoning, starts with the goal and works backward, chaining through rules to find known facts that support the goal.

It is a top-down approach that is based on modus ponens inference rule and uses a depth-first search strategy for proof. In backward chaining, the goal is broken into sub-goal(s) to prove the facts true. It is a goal-driven approach, as a list of goals decides which rules are selected and used. Backward chaining algorithm follows a process similar to that of

hypothesis testing in human problem-solving. The only difference is that backward chaining algorithm is focused on the required rules.

Backward chaining algorithm is used in game theory, automated theorem proving tools, inference engines, proof assistants and various other AI applications.

Example: Use the facts given below to prove that ‘Raza is criminal.’

‘As per the law, it is a crime for an Indian to sell weapons to enemy nations. ABC is an enemy of India and has some missiles, and all the missiles were sold to it by Raza, who is an Indian.’

The first step is to convert all the facts into first-order definite clauses.

Facts Conversion into FOL:

It is a crime for an Indian to sell weapons to hostile nations.

Using variables p, q and r, we can write,

$$\begin{aligned} &\text{Indian}(x) \wedge I && (i) \\ &\text{weapon}(y) \wedge \\ &\text{sells}(x,y,z) \wedge \\ &\text{enemy}(p, \\ &\text{India}) \rightarrow \\ &\text{Criminal}(x) \end{aligned}$$

ABC is an
enemy of
India.

This can be (ii)
written as,
enemy(ABC,
India)

ABC has
some
missiles.

Owns(ABC, (iii)
p) $\wedge$
Missile(p)

All of the
missiles
were sold to
ABC by Raza.

$\forall p$ (iv)
Missiles(p) $\wedge$
Owns (ABC,
p) $\rightarrow$ Sells
(Raza, p,
ABC)

Missiles are
weapons.

Missile(p) $\rightarrow$ (v)
Weapons (p)

Raza is
Indian.

Indian(Raza). (vi)

Backward-Chaining proof:

In backward chaining, we will start with our goal predicate (refer **Fig. 4.18**), which is `Criminal(Raza)`, and then infer further rules. Our motive is to prove that all the facts true.

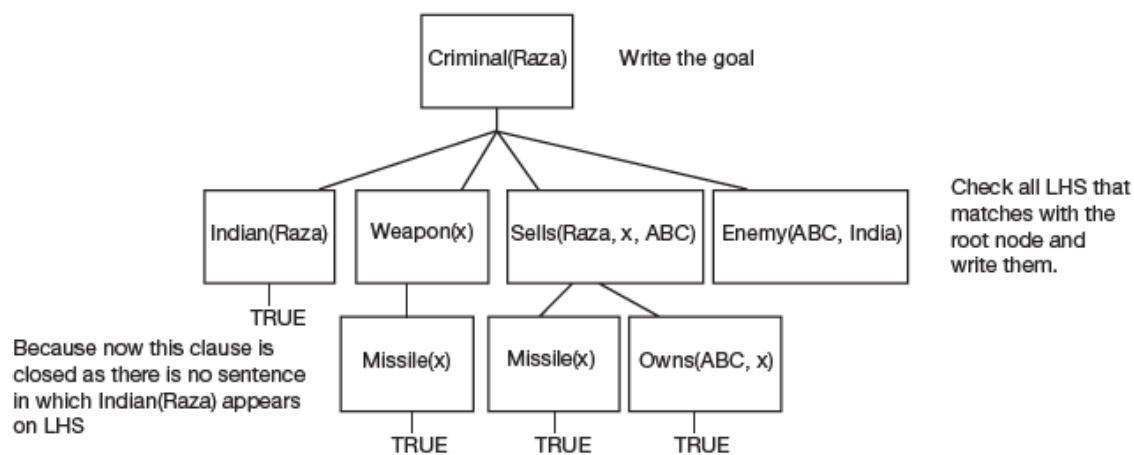


FIGURE 4.18 Backward Chaining Proof

Note that at the second level, we need to infer other facts from the goal fact which satisfy the rules. Similarly, at level three, we extract further facts that can be inferred from facts in level 2 and satisfy the rules. Now that all the rules have been proved true, we have successfully proved that Raza is a criminal using backward chaining technique.

4.9.6 Pros and Cons of Backward Chaining

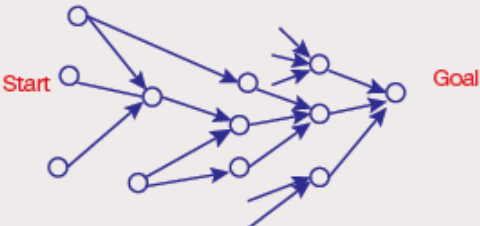
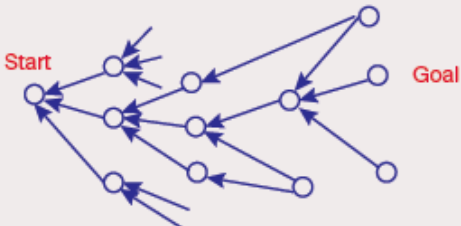
Advantages of Backward Chaining

1. If pre-decided principles are met by the inference engine, then the right arrangements can be easily inferred.
2. As compared to forward chaining, backward chaining is a faster strategy as the endpoint is accessible.
3. Since the goal is known, it becomes easier to deduce inferences.
4. It helps to solve complex problems into easy, small, fixed sections.
5. It is suitable for problems involving diagnostics, prescription and debugging.

Disadvantages of Backward Chaining

1. It just infers information that is required, which makes it less adaptable than forward chaining.
2. It cannot be used to conclude multiple solutions.
3. The path towards reasoning begins only if the goal is known.
4. It may follow a particular path of reasoning and search even though it eventually proves to be impractical.

TABLE 4.12 Forward Chaining vs Backward Chaining

Forward Chaining	Backward Chaining
It follows a bottom-up approach.	It follows a top-down approach.
It uses breadth-first search strategy.	It uses depth-first search strategy.
It works in the forward direction.	It works in the backward direction.
The technique is data-driven that uses available data to reach the goal state.	The technique is goal-driven.
Reception of new data results in new inferences, which makes the engine more suitable for dynamic situations.	It is not dynamic when compared with forward chaining.
It is used in planning, monitoring, controlling and interpreting applications.	It is employed in automated inference engines, theorem proofs, proof assistants, etc.
When there is no new data in the system, using forward chaining makes sense.	If it is a query to which a response is required, backward chaining is preferred.
Examples include XCON, the rule-based system; it is a forward chaining expert system that is used to configure large computer systems.	MYCIN is an example of backward chaining expert systems that used AI to identify severe infections caused by bacteria.
	



PROGRAMMER’S ZONE

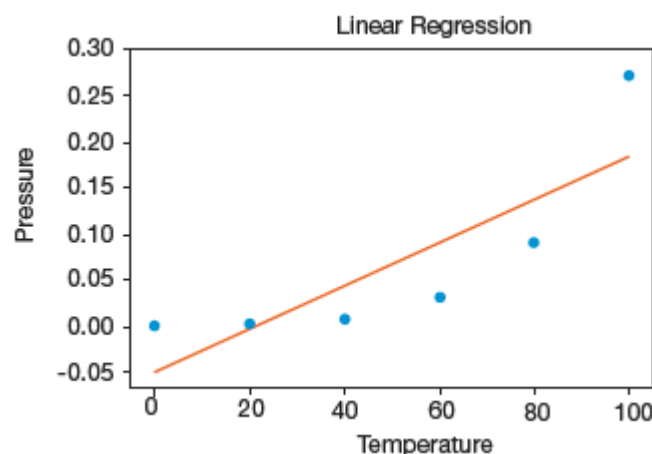
Both forward chaining and backward chaining have their pros and cons. Therefore, new systems are created using hybrid chaining inference technique that deliver the advantages of both forward and backward chaining. These systems are more powerful and are capable of performing tasks that were earlier termed unachievable.

A Polynomial Regression algorithm is also called Polynomial Linear Regression because it does not depend on the variables, instead, it depends on the coefficients, which are arranged in a linear fashion.

Download a sample data set from

<https://media.geeksforgeeks.org/wp-content/uploads/data.csv> and compare the performance of linear and polynomial regression.

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
# Importing the dataset
datas = pd.read_csv('data1.csv')
print(datas )
# Dividing the dataset into 2 components
X = datas.iloc[:, 1:2].values
y = datas.iloc[:, 2].values
# Fitting Linear Regression to the dataset
# Fitting Linear Regression to the dataset
from sklearn.linear_model import LinearRegression
lin = LinearRegression()
lin.fit(X, y)
# Visualising the Linear Regression results
plt.scatter(X, y, color = 'blue')
plt.plot(X, lin.predict(X), color = 'red')
plt.title('Linear Regression')
plt.xlabel('Temperature')
plt.ylabel('Pressure')
plt.show()
```

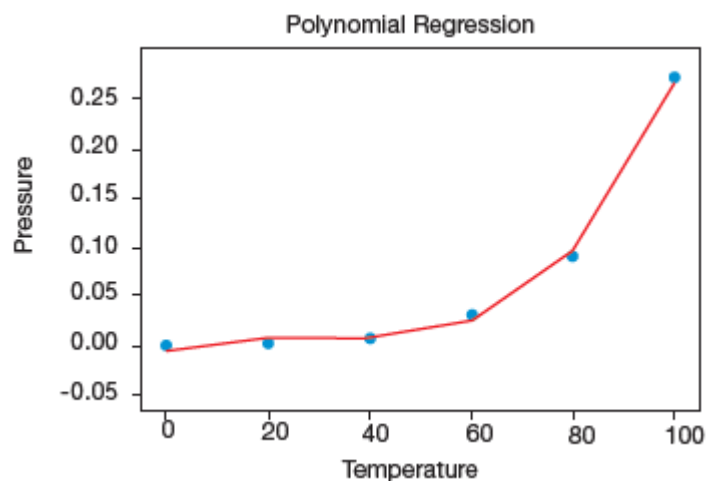


```

# Fitting Polynomial Regression to the dataset
from sklearn.preprocessing import PolynomialFeatures
poly = PolynomialFeatures(degree = 4)
X_poly= poly.fit_transform(X)
poly.fit(X_poly, y)
lin2 = LinearRegression()
lin2.fit(X_poly, y)
# Visualising the Polynomial Regression results
plt.scatter(X, y, color = 'blue')
plt.plot(X, lin2.predict(X_poly), color =
'red')
plt.title('Polynomial Regression')
plt.xlabel('Temperature')
plt.ylabel('Pressure')
plt.show()

```

To prevent over-fitting, we can add more training samples so that the algorithm doesn't learn the noise in the system and can become more generalized.



Generate a random data set having 10 samples and 5 features. Use Ridge Regression with an alpha value =0.2 to create a model. Print the score of this model and also the value of attributes.

```

from sklearn.linear_model import Ridge
import numpy as np
n_samples, n_features = 10, 5
#np.random.RandomState() constructs a random number generator.
rng = np.random.RandomState(0)
y = rng.randn(n_samples)
X = rng.randn(n_samples, n_features)
rdg = Ridge(alpha = 0.2)
rdg.fit(X, y)
print(rdg.score(X, y))
print(rdg.coef_)
print(rdg.intercept_)

```

OUTPUT

```

0.8657654687023124
[ 1.1961396 -0.09574812 -0.96234511  0.06421247  1.66880754]
1.1808729929860098

```

Implementing lasso Regression on cancer data

```

import math
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
# difference of lasso and ridge regression is that some of the
coefficients can be zero i.e. some of the features are
# completely neglected
from sklearn.linear_model import Lasso
from sklearn.linear_model import LinearRegression
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
cancer = load_breast_cancer()
#print cancer.keys()
cancer_df = pd.DataFrame(cancer.data, columns=cancer.feature_names)
#print cancer_df.head(3)
X = cancer.data
Y = cancer.target
X_train,X_test,y_train,y_test=train_test_split(X,Y, test_size=0.3,
random_state=31)
# instantiate the Lasso Regression model
lasso = Lasso()
# fits the model to the training data.
lasso.fit(X_train,y_train)
train_score=lasso.score(X_train,y_train)
test_score=lasso.score(X_test,y_test)
coeff_used = np.sum(lasso.coef_!=0)
print('training score:', train_score )
print('test score: ', test_score)
print('number of features used: ', coeff_used)

```

OUTPUT

```
training score: 0.5600974529893081  
test score: 0.5832244618818156  
number of features used: 4
```

Implement ElasticNet Regression on the Boston Housing Data Set.

OUTPUT

**Implement binary logistic regression on random data values.
Calculate the accuracy of the model and plot the confusion matrix.
Set the value of $C = 10$ and check whether the model performs better
or not.**

OUTPUT



KEY TERMS

Meta-knowledge: It is knowledge about knowledge.

Heuristic knowledge: It refers to knowledge about a specific topic.

Procedural knowledge or imperative knowledge: It refers to knowledge that gives information about how to achieve or do something. This knowledge includes rules, strategies, procedures, agendas, etc. that can be directly applied to perform any task.

Declarative knowledge: This is the information that we have about an object. This knowledge helps us to describe a particular concept, fact, object and its attributes.

Structural Knowledge: This is the basic knowledge to solve complex problems which describes relationships between various concepts or objects such as kind of, part of, and grouping of something.

Logic: It is the main component of any knowledge that facilitates a person to draw conclusion by filtering required information from large volumes of sentences.

Frame: It is a record-like structure that describes a real world using a collection of attributes and values.

Proposition: It is a declarative statement which is either true or false but it cannot be both (true as well as false).

Atomic proposition: It is a simple proposition made of a single proposition symbol. Such propositions can be either true or false.

Compound proposition: These propositions are formed by combining simpler atomic propositions, using parenthesis and logical operators.

Inference: It is the process of generating conclusions from evidence and facts.

Inference rules: These are the templates for generating valid arguments that are used to derive proofs to obtain conclusions that finally lead to the desired goal.

Free Variable: It is a variable in a formula that occurs outside the scope of the quantifier.

Bound Variable: It is a variable in a formula that occurs within the scope of the quantifier.

Skolemization: Eliminating existential instantiation quantifier ($\exists$) by replacing it with skolem constant.

Definite clause: This is a clause which is a disjunction of literals with exactly one positive literal.

Horn clause: This is a clause which is a disjunction of literals with at most one positive literal.



RECAPITULATION POINTS

- Knowledge representation and reasoning (KR, KRR) is a part of artificial intelligence which focuses on how AI agents think and how their thinking contributes to intelligent behaviour.
- A knowledge-based system represents information about the real world in such a way that a computer can easily understand and utilize this knowledge to solve complex real-world problems.
- Knowledge-based agents represent knowledge that can be reasoned to act intelligently according to the requirements. Usually, these agents store knowledge about their surroundings in the form of sentences.
- The inference system derives new sentences from old ones and also facilitates addition of new sentence(s) to the knowledge base.
- Logical representation is a language with some definite rules.
- The production rules system consists of (condition, action) pairs which mean, 'If condition then action'.
- At times, it happens that multiple production rules are fired together. This situation is called conflict set. In such a situation, the agent needs

to select one particular rule. This is often referred to as conflict resolution.

- While universal quantifier, represented by $\forall$, specifies that the statement (within its range) is true for everything or every instance of a particular thing, existential quantifiers (denoted as $\exists$) express that the statement within its scope is true for at least one instance of something.
- The generalized modus ponens states that if P implies Q and P is asserted to be true, then Q must be true.
- The unification process is used to make two different logical atomic expressions (or literals) identical by finding a substitution. Resolution is usually used when there are multiple statements and we have to prove a conclusion.
- Forward chaining starts with atomic sentences in the knowledge base and applies inference rules (modus ponens) in the forward direction to extract more data until a goal is reached.
- Backward-chaining starts with the goal and works backward, chaining through rules to find known facts that support the goal.



ANSWER THE QUESTIONS

1. What knowledge needs to be represented in an AI system?
2. Broadly classify the types of knowledge.
3. Define logic and give its main components.
4. Explain the components of an AI system.
5. Give the properties of a knowledge representation system.
6. What are knowledge-based agents? Explain their role and architecture.
7. With the help of relevant examples, state the operations performed by a knowledge-based agent.
8. What are the different levels at which a knowledge-based agent can be viewed?
9. Explain the different techniques of representing knowledge.
10. What do you understand by recognize-act cycle and conflict resolution?
11. What different logical connectives are used in propositional logic?

12. Explain the basic rules in First Order Logic.
13. Give the conditions that must be satisfied for unification process.
14. List the steps to be performed in a resolution process.
15. Represent the following statements in the form of nodes and arcs.

Statements:

1. Jerry is a cat.
 2. Jerry is a mammal.
 3. Jerry is owned by Priya.
 4. Jerry is brown coloured.
 5. All mammals are animals.
16. Consider an entity, Achyut, who is a 25-year-old engineer by profession and lives in Rameshwaram, India. Design a frame to record this information.
 17. Write the rules for the following statements.
 1. If it rains today and one does not carry umbrella, he will be drenched.
 2. If it rains today and Arjun does not carry umbrella, then he will be drenched.
 18. Can the following statements be represented using propositional logic? Justify your answer.
 1. 'Earth is round'
 2. 'Some children are lazy'
 3. 'Garlic tastes wonderful'
 19. If, P = Aakash goes to temple and Q = Aakash is religious
 1. Prove using Modus Tollens that Akash does not go to temple.
 2. What does Modus Ponens say in this case?
 20. Given the facts, if Sheero is intelligent, then Sheero is smart. Sheero is intelligent. Prove that Sheero is smart.
 21. If Ram is the friend of Shyam and Shyam is the friend of Rahul, then prove that Ram is the friend of Rahul.
 22. Represent the fact, 'Myra is not beautiful or she is obedient'.
 23. Given the facts, 'All human beings are mortal.' 'Sachin is a human being.' Prove that 'Sachin is mortal'.
 24. Let P be 'it snows.' Let Q be 'I will study.' Let R be 'I will get an A grade.'

This can conclude that 'If it snows, then I will study' and 'If I study, I will get an A grade.' Therefore, 'If it snows, I will get an A grade.'

Using which rule, can we get this conclusion? Explain.

25. Name the tautology and give an example.

1. $((p) \wedge (q)) \rightarrow (p \wedge q)$
2. $((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$
3. $((p \rightarrow q) \wedge (r \rightarrow q) \wedge (p \vee r)) \rightarrow q$
4. $((p \rightarrow q) \wedge (r \rightarrow s) \wedge (p \vee r)) \rightarrow (q \vee s)$
5. $(p \rightarrow q) \wedge (r \rightarrow s) \wedge (\neg q \vee \neg s) \rightarrow (\neg p \vee \neg r)$

26. Given the assumptions, prove by resolution.

1. Assumptions

1. Fido is a dog
2. All dogs are animals
3. All animals will die

Goal- Fido will die.

2. Axioms

1. $\text{allergies}(X) \rightarrow \text{sneeze}(X)$
2. $\text{cat}(Y) \wedge \text{allergic-to}(X, Y) \rightarrow \text{allergies}(X)$
3. $\text{cat}(\text{Felix})$
4. $\text{allergic-to}(\text{Liz}, \text{Felix})$

Goal: $\text{sneeze}(\text{Liz})$

3. Assumptions

1. All hounds howl at night.
2. Anyone who has any cat will not have any mice.
3. Light sleepers do not have anything which howls at night.
4. John has either a cat or a hound.

Goal: If John is a light sleeper, then John does not have any mice.

4. Assumptions

1. Every Austinite who is not conservative loves some armadillo.
2. Anyone who wears maroon-and-white shirts is an Aggie.
3. Every Aggie loves every dog.
4. Nobody who loves every dog loves any armadillo.
5. Clem is an Austinite, and Clem wears maroon-and-white shirts.

Goal: Is there a conservative Austinite?

5. Assumptions

1. Anyone whom Mary loves is a football star.
2. Any student who does not pass does not play.
3. John is a student.
4. Any student who does not study does not pass.

5. Anyone who does not play is not a football star.

Goal: If John does not study, then Mary does not love John.

6. Assumptions

1. Every coyote chases some roadrunner.
2. Every roadrunner who says “beep-beep” is smart.
3. No coyote catches any smart roadrunner.
4. Any coyote who chases some roadrunner but does not catch it is frustrated.

Goal: If all roadrunners say “beep-beep”, then all coyotes are frustrated.

7. Assumptions

1. Anyone who rides any Harley is a rough character.
2. Every bike rider [something that is] either a Harley or a BMW.
3. Anyone who rides any BMW is a yuppie.
4. Every yuppie is a lawyer.
5. Any nice girl does not date anyone who is a rough character.
6. Mary is a nice girl, and John is a biker.

Goal: If John is not a lawyer, then Mary does not date John.

8. Assumptions

1. Every child loves anyone who gives the child any present.
2. Every child will be given some present by Santa if Santa can travel on Christmas eve.
3. It is foggy on Christmas eve.
4. Anytime it is foggy, anyone can travel if he has some source of light.
5. Any reindeer with a red nose is a source of light.

Goal: If Santa has some reindeer with a red nose, then every child loves Santa.

9. Assumptions

1. Every investor bought [something that is] stocks or bonds.
2. If the Dow-Jones Average crashes, then all stocks that are not gold stocks fall.
3. If the T-Bill interest rate rises, then all bonds fall.
4. Every investor who bought something that falls is not happy.

Goal: If the Dow-Jones Average crashes and the T-Bill interest rate rises, then any investor who is happy bought some gold stock.

10. Assumptions

1. Every child loves every candy.
2. Anyone who loves some candy is not a nutrition fanatic.
3. Anyone who eats any pumpkin is a nutrition fanatic.
4. Anyone who buys any pumpkin either carves it or eats it.
5. John buys a pumpkin.
6. Lifesavers is a candy.

Goal: If John is a child, then John carves some pumpkin.

27. Unify the following pairs of predicates and give the resulting substitutions.

Assumptions

1. Every child loves Santa.
2. Everyone who loves Santa loves any reindeer.
3. Rudolph is a reindeer, and Rudolph has a red nose.
4. Anything which has a red nose is weird or is a clown.
5. No reindeer is a clown.
6. Scrooge does not love anything which is weird.

Goal: Scrooge is not a child.

28. Write short notes on the following:

1. Inference rules
2. First order logic
 1. Differentiate between the following:
 1. Declarative and procedural approach of designing KBA
 2. Universal and existential quantifier
 3. Free and bound variables
 4. Forward and backward chaining



1. An intelligent agent acts by _____ its environment.
2. If we remove the _____ component, then the decision-maker will not be able to sense the environment accurately and take appropriate decisions.
3. Knowledge representation models intelligent behaviour for an _____.
4. The knowledge about knowledge is known as _____.
5. In AI, knowledge is represented using _____.

6. _____ uses inference algorithms to think all the possible reasons that can give a proper result.
7. _____ efficiency helps the knowledge representation system to easily acquire new knowledge.
8. Knowledge-based agents store knowledge about their surroundings in the form of _____.
9. _____ is the knowledge-based system engine that infers new knowledge in the system.
10. _____ represents knowledge through graphical networks.
11. _____ system consists of (condition, action) pairs which mean, 'If condition then action'.
12. _____ is a declarative statement which is either true or false but it cannot be both.
13. A proposition formula which is always true is called _____.
14. A proposition formula which is always false is called _____.
15. Two propositions are said to be _____ if and only if the columns in the truth table are identical to each other.
16. Law of associativity states that $(P \vee Q) \vee R =$ _____.
17. Statements which cannot be answered absolutely are called _____ statements.
18. The process of generating conclusions from evidence and facts is known as _____.
19. The negation of implication is called _____ which is represented as _____.
20. The _____ syllogism rule states that if $P \vee Q$ is True, and $\neg P$ is True, then Q will be true.
21. A complex sentence in FOL has two main parts— _____ and _____.
22. _____ specifies the quantity of specimen in the universe of discourse.
23. _____ operator specifies that the two terms refer to the same object.
24. The _____ states that if P implies Q and P is asserted to be true, then Q must be True.
25. The substitution variables are called _____.
26. _____ process is used to make two different logical atomic expressions identical by finding a substitution.
27. _____ is a sentence represented as a conjunction of clauses.
28. A clause which is a disjunction of literals with exactly one positive literal is known as a _____ clause.

29. _____ chaining starts with atomic sentences in the knowledge base and applies inference rules (Modus Ponens) in the forward direction to extract more data until a goal is reached.
30. If today is Monday is represented as P, then today is not a Monday will be written as _____.
31. If Ram is a cricket player is written as P and Ram is a hockey player is represented as Q, then Ram plays both cricket and hockey is represented by _____.
32. If Ram leaves for Mumbai is represented by P and Ram leaves for Chennai is written as Q, then Ram leaves for Chennai or Mumbai is represented by _____.
33. The clause if it is Sunday (P), then I will go to Movie (Q), is represented as _____.
34. If I have Rs 500 then only I will buy the book. The converse condition that I will buy this book if and only if I have Rs 500 is represented as _____.
35. It rains in July can be written as _____.



STATE TRUE OR FALSE

1. Logic is the main component of any knowledge that facilitates a person to draw conclusion by filtering required information from large volumes of sentences.
2. Syntax is used to check if a syntactically correct sentence is logically correct and meaningful or not.
3. Inferential adequacy directs the inferential knowledge mechanism to generate appropriate results.
4. Learning element of the KBA updates the knowledge base every few days.
5. In the procedural approach, the desired behaviour is directly encoded in the agent as a program code.
6. Propositional Logic (FoL) represents the objects in quantifiers and predicates.
7. It is impossible to represent all facts from human's memory into a semantic network.

8. Statements which are questions, commands or opinions are not propositions.
9. The semantics of propositional logic defines the allowable sentences for the knowledge representation.
10. $(P \wedge Q) \wedge R = P \wedge (Q \wedge R)$ represents the law of distributive.
11. $\neg (P \wedge Q) = (\neg P) \vee (\neg Q)$
12. Every relationship can be represented using propositional logic.
13. Subject is a relation which binds two atoms together in a statement.
14. In existential quantifier, AND or conjunction symbol ($\wedge$) is used as the main connective.
15. A free variable occurs within the scope of the quantifier.
16. Universal instantiation is an inference rule in first-order logic that can be applied only once to replace the existential sentence.
17. The unification process will fail if two similar variables are present in the same expression.
18. A clause which is a disjunction of literals with at most one positive literal is known as definite clause.
19. Forward chaining can be used to draw multiple conclusions.



MULTIPLE CHOICE QUESTIONS

1. _____ represents information and facts about actions which occur in the real world.
 1. Objects
 2. Events
 3. Performance
 4. Meta-knowledge
2. _____ knowledge helps us to describe a particular concept, fact, object and its attributes.
 1. Meta
 2. Procedural
 3. Declarative
 4. Structural
3. _____ component retrieves data from the environment, discovers the source(s) of noise, checks if the AI was damaged by anything and defines response to be given when any sense has been detected.

1. Perception
 2. Learning
 3. Planning and Execution
 4. Knowledge representation and reasoning
4. With _____ component, the system focuses on self-improvement through knowledge acquisition, inference, acquisition of heuristics, faster searches, etc.
1. Perception
 2. Learning
 3. Planning and execution
 4. Knowledge representation and reasoning
5. Which of the following is not a function planning and execution component?
1. Select an initial state
 2. Enumerate preconditions and effects
 3. Enumerate a sequence of actions to achieve the goal state
 4. Reasoning and suggestions for self-improvement
6. _____ manipulates the representational structures to produce new knowledge corresponding to the existing structure.
1. Representational accuracy
 2. Inferential adequacy
 3. Inferential efficiency
 4. Acquisitional efficiency
7. _____ derives new sentences from old ones and also facilitates addition of new sentence(s) to the knowledge base.
1. Knowledge representation system
 2. Knowledge base system
 3. Inference engine
 4. Knowledge-based agent
8. _____ operation informs the knowledge base which action was selected and performed by the agent.
1. ASK
 2. TELL
 3. PERFORM
 4. Any of these
9. Sentences are encoded into different logics at _____ level.
1. Knowledge

2. Logical
 3. Implementational
 4. All of these
10. _____ representation is a language with some definite rules.
1. Logical
 2. Frame
 3. Semantic
 4. All of these
11. We can express a propositional logic using _____ operators.
1. Arithmetic
 2. Relational
 3. Logical
 4. Membership
12. Identify the incorrect statement.
1. An atomic proposition is said to be satisfiable if there is an interpretation for which it is true.
 2. A contingent statement is one which is neither a tautology nor a contradiction.
 3. Modus Tollens rule states that if $P \rightarrow Q$ is True and $\neg Q$ is True, then $\neg P$ will also true.
 4. The Disjunctive Syllogism rule states that $P \rightarrow R$ is True iff $P \rightarrow Q$ is true, and $Q \rightarrow R$ is true.
13. A sentence of the form $P \Leftrightarrow Q$ represents _____
1. Negation
 2. Disjunction
 3. Implication
 4. Bi-directional
14. $P \wedge Q = Q \wedge P$ denotes the law of _____.
1. Commutativity
 2. Associativity
 3. Identity Element
 4. Distributive
15. $P \wedge \text{True} = \text{_____}$.
1. 0
 2. 1
 3. P
 4. True

16. _____ are the templates for generating valid arguments that are used to derive proofs to obtain conclusions that finally leads to the desired goal.
1. Inferences
 2. Inference rules
 3. Compound proposition
 4. Implications
17. _____ is an inference rule that states if P is True and $P \rightarrow Q$ is True, then we can infer that Q will also be true.
1. Modus Ponens
 2. Modus Tollens
 3. Hypothetical syllogism
 4. Disjunctive syllogism
18. In universal quantifier, _____ is used as the main connective.
1. Negation
 2. Disjunction
 3. Implication
 4. Bi-directional
19. Which of the following is not similar?
1. $\forall x \forall y$ and $\forall y \forall x$
 2. $\exists x \exists y$ and $\exists y \exists x$
 3. $\exists x \forall y$ and $\forall y \exists x$
 4. None of these
20. _____ allows us to infer any sentence $P(c)$ by substituting a ground term, c for the variable.
1. Universal Generalization
 2. Universal Instantiation
 3. Existential Instantiation
 4. Existential introduction
21. _____ is usually used when there are multiple statements and we have to prove a conclusion.
1. Unification
 2. Substitution
 3. Resolution
 4. Instantiation
22. Resolution technique is used in _____ logic.
1. Propositional

2. First-order predicate
 3. Both of these
 4. None of these
23. Identify the correct statement about forward chaining.
1. It is said to be data-driven approach.
 2. It works using bottom-up approach.
 3. It uses PROLOG language to deal with inference rules.
 4. All of these
24. Identify the incorrect statements.
1. Forward chaining is slower than backward chaining.
 2. Forward chaining might give different conclusions, which may result in ambiguity.
 3. Backward chaining uses bottom-up approach.
 4. Backward chaining uses a depth-first search strategy for proof.



ANSWERS

Fill Up

1. Sensing
2. Knowledge
3. agent
4. Meta-knowledge
5. Logics
6. logical inference
7. Acquisitional
8. sentences
9. Inference engine (IE)
10. Semantic network
11. Production rules
12. Proposition
13. tautology
14. Contradiction.
15. logically equivalent
16. $P \vee (Q \vee R)$
17. Open

18. Inference
19. Inverse, $\neg P \rightarrow \neg Q$.
20. Disjunctive
21. subject and predicate
22. Quantification
23. Equality
24. Generalized Modus Ponens
25. Most general unifier
26. Unification
27. Conjunctive normal form
28. Definite
29. Forward
30. $\neg P$
31. $P \wedge Q$
32. $P \vee Q$
33. $P \rightarrow Q$
34. $P \Leftrightarrow Q$
35. Rains (July)

State True or False

1. True
2. False
3. False
4. False
5. True
6. False
7. True
8. True
9. False
10. False
11. True
12. False
13. False
14. True
15. False
16. False

- 17. True
- 18. False
- 19. True

Multiple Choice Questions

- 1. b
- 2. c
- 3. a
- 4. b
- 5. d
- 6. b
- 7. c
- 8. b
- 9. b
- 10. a
- 11. c
- 12. d
- 13. d
- 14. a
- 15. c
- 16. b
- 17. a
- 18. c
- 19. c
- 20. b
- 21. c
- 22. c
- 23. d
- 24. c

Previous chapter

< [3. Artificially Intelligent Machine](#)

Next chapter

[5. Reasoning and Learning in Artificial Intelligence](#)

>

Table of contents expanded

5

Reasoning and Learning in Artificial Intelligence

5.1 Reasoning

The primary aim of any artificially intelligent system is to think like humans and mimic their actions to perform tasks faster in a smarter fashion. In this process, reasoning plays a vital role as it helps draw conclusions, make predictions and construct approaches towards a particular thought with the help of existing knowledge.

The term 'reasoning' is associated with thinking, intellect and cognition that directs the process of generating one idea from the other. It involves analysing the given information, comparing it with what is known and then coming to a conclusion. Therefore, reasoning is the key to making decisions, solving problems and evaluating actions. For example, Siri (virtual assistant) uses reasoning to offer suggestions and recommendations based on commands like nearest location, the date for tomorrow, etc.

5.1.1 Features of a Reasoning System

The four qualities of a reasoning system are discussed as follows:

1. A reasoning system is a **self-contained model or system** as it is not modified externally by a programmer or user for carrying out the tasks at hand. It is a collection of models or algorithms or systems that is untouched in the course of learning new tasks.
2. A reasoning system can solve a different set of problems either through examples, or instruction or even trial-and-error. Some of these problems may be new, unique and unusual.
3. It gets better through a limited number of trials through practice.

4. A reasoning system should solve new problems in a limited time frame and from a small number of examples.

5.1.2 Why Focus on Reasoning System?

Before reading about why to go for a reasoning system, let us first try to understand the difference between prediction, inference and reasoning.

Prediction is done by pattern matching. A prediction system accepts a lot of data and then maps input to output to predict outputs.

Inference is the process of generating new information. It is an automatic process that has no goals. An inference system can be thought of as a machine that takes some inputs and processes it to derive new information as output.

Reasoning system, on the other hand, is targeted or is directed to achieve the goal of deducing an inference. For this, inputs are carefully selected, organized and processed to generate the desired result.

Machines use reasoning, along with knowledge representation, logic and learning for analysis, problem-solving and making conclusions.

Nowadays, extensive research has been done to make machine learning algorithms meticulously perform prediction tasks when the problem is well-defined and a large amount of data is available. However, these systems are prone to adversarial attacks. To improve their capabilities, AI systems must themselves identify and recover from failure, and solve problems without any external help. Let us now try to understand the different methods of reasoning.

5.1.3 Methods of Reasoning

We know that reasoning is the mental process of deriving a logical conclusion and making predictions from available knowledge, facts and beliefs. It helps us to derive valid conclusions by thinking rationally to infer facts from existing data. In artificial intelligence, reasoning facilitates ma-

chines to think rationally and perform like a human. Reasoning is classified into the following types:

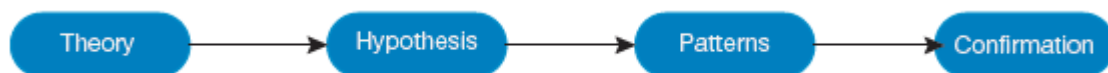
Deductive Reasoning

It is a strategic approach that uses available facts, information or knowledge to draw valid conclusions. Every result is based on facts and ideas and is drawn using a top-down approach. In deductive reasoning, the arguments can be valid or invalid based on the value of the premises (rules or facts). A conclusion is true only if the value of the premises is true.

Deductive reasoning scans a generalized statement to provide a valid conclusion. For example:

- Young people in the age group 15 –25 are active on social media.
- Out of 60 students in a class, the ratio of boys is more than that of girls.

Thus, deductive reasoning uses formal logic to produce logically certain results. This process of drawing conclusion involves different steps described as follows.



Deductive reasoning usually starts from the general premises to get to specific conclusion(s). Compared to inductive reasoning (discussed below), it is difficult to use deductive reasoning as it requires true facts. For example:

Premise-1: All humans breathe.

Premise-2: Naina is a human.

Conclusion: Naina breathes.

Inductive Reasoning

Contrary to deductive reasoning, inductive reasoning also known as cause-effect reasoning; it generates hypothesis rather than drawing

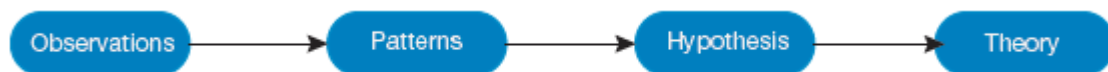
conclusion(s) to the facts at the beginning of the process.

Inductive reasoning follows a bottom-up approach to make generalized statements from specific facts and knowledge. Therefore, there may be a case when the premises are true but the conclusion may not be true as the inductive argument may be either strong or weak. Some examples of inductive reasoning are as follows:

- All the students scoring 100% are from state X.
- City X recorded the lowest temperature in north India.

Reasoning allows AI technologies to extract critical information from large sets of structured and unstructured data.

Inductive reasoning is preferred in cases with limited set of facts and data to deduce a conclusion. In such a scenario, the inductive reasoning technique uses historical data to produce a generic rule whose conclusion is supported by the premises. Therefore, it is said to be exploratory in nature as it allows for uncertain but likely results. The steps involved in inductive reasoning include the following:



Therefore, inductive reasoning is fast and easy, as it requires evidence instead of facts. It can derive conclusions using limited data by the process of generalization (start with specific data to a general statement or conclusion). A comparative analysis of inductive and deductive reasoning is given in **Table 5.1**.

TABLE 5.1 Inductive vs Deductive reasoning

Attribute	Deductive Reasoning	Inductive Reasoning
Definition	Deductive reasoning is a form of valid reasoning that deduces new information from known related facts and information.	Inductive reasoning arrives at a conclusion by the process of generalization using specific facts or data.
Approach	It follows a top-down approach.	It follows a bottom-up approach.
Starts from	It starts from the premises (or rules).	It starts from the conclusion.
Validity	Conclusion must be true if the premises are true.	Truth of premises does not guarantee the truth of conclusions.
Usage	It is difficult to use as facts that must be true are required.	It is fast and easy to use, as we need evidence instead of true facts. It is mostly used in our daily lives.
Process	Theory → hypothesis → patterns → confirmation.	Observations → patterns → hypothesis → theory.
Argument	Arguments may be valid or invalid.	Arguments may be weak or strong.
Structure	It reaches from general facts to specific facts.	It reaches from specific facts to general facts.
Conclusion	The conclusions are certain.	The conclusions are probabilistic.

Common Sense Reasoning

It is the most frequently used reasoning in daily life and usually comes from day-to-day experiences. When we humans face a different situation in life, we gain some knowledge. If, in the future, we encounter the same situation, we use our past experiences to draw a conclusion to handle that situation. Some of the examples are as follows:

- If, in a lockdown, we went out of our houses without a mask and had to pay a heavy fine, then the next time, we go out of the house, we are aware about consequences and will not repeat the same action.
- While crossing roads, we always recall a set of rules to be followed.

Inductive and deductive reasoning are the forms of propositional logic.

Common sense reasoning helps machines simulate human ability to make decisions and perceptions about real-world events. In AI systems, it is mainly used in applications involving computer vision and robotic manipulation.

1. Common sense reasoning is an informal form of reasoning, which can be gained through experiences.
2. It simulates the human ability to make presumptions about events which occurs on every day.
3. It relies on good judgment rather than exact logic
4. Common sense works with **heuristic knowledge** and **heuristic rules**.

Any theorem proving is an example of monotonic reasoning.

Monotonic Reasoning

Monotonic reasoning uses facts, information and knowledge to draw a conclusion. However, once a conclusion is taken, it always remains constant. Addition of new facts and knowledge in the database cannot alter the conclusion drawn at an earlier point in time. It is mainly used in conventional reasoning systems and logic-based systems. Examples of monotonic reasoning are as follows:

- The Harappan civilization was located in the Indus river valley.
- Kashmir is hailed all over the world for its incredible natural beauty.

Human beings' perceptions for various things in daily life is an example of non-monotonic reasoning.

However, since monotonic reasoning does not modify the conclusion based on the new data, it cannot be used in systems that accept real-time inputs to generate outputs.

Monotonic reasoning is used in conventional reasoning systems, and a logic-based system is monotonic.

Advantages

1. Each proof will always remain valid.
2. Facts deduced from existing facts will always remain valid.

Abductive reasoning is a form of logical reasoning which starts with observation(s) to find the most likely explanation or conclusion for the observation.

Disadvantages

1. Real-world scenarios cannot be expressed using monotonic reasoning.
2. Hypothesis knowledge cannot be expressed. Monotonic reasoning represents only true facts.
3. It allows deriving conclusions from the old proofs, so new knowledge from the real world cannot be added.

Non-Monotonic Reasoning

Unlike monotonic reasoning, non-monotonic reasoning labels a conclusion as invalid if on addition of new or more information, a previously generated conclusion becomes contradictory. This type of reasoning usually deals with uncertain and incomplete models.

Non-monotonic logics are devised to capture and represent defeasible inferences (draw tentative conclusions). These conclusions can be retracted when further evidences are available.

Consider the following example. If we have, in our knowledge base, statements like, 'Typically, birds fly. Penguins do not fly. Tweety is a bird'. It is plausible to conclude that Tweety flies. However, if at a later point in time, we add in the knowledge base that Tweety is a penguin, then the previous conclusion must be retracted and a new conclusion will be drawn that would say, Tweety does not fly.

Advantages of Non-Monotonic Reasoning

1. Non-monotonic reasoning is very useful for real-world systems such as robot navigation.
2. It helps to choose probabilistic facts or can make assumptions.

Disadvantages of Non-Monotonic Reasoning

1. Old facts may be invalidated by adding new sentences.
2. It cannot be used for theorem proving.

Abductive Reasoning

This type of reasoning is quite different from those discussed above. Abductive reasoning is an extension of deductive reasoning, and starts with an incomplete set of facts, information and knowledge to find the most appropriate explanation and conclusion. Conclusions are based on facts known at the present. It does not give any importance to outdated facts and information. Such type of reasoning is very important in making decisions in our daily lives. For example:

- A doctor draws conclusions about a patient's health based on the present test reports.
- You will not plan a vacation to a country that is currently facing a war-like situation irrespective of how beautiful and adventurous it has been in the past.

Thus, abductive reasoning, is a form of logical inference that uses theories to define observations having the simplest and likely explanation to give the most accurate suggestions and guesses. It is usually associated with troubleshooting and decision-making. However, the main drawback is that it does not rectify the plausible conclusion. So, it must be used only in case of data certainties. Moreover, abductive reasoning can justify any belief (even absurd ones). Therefore, it is not always reliable.

5.2 Probabilistic Reasoning in Artificial Intelligence

5.2.1 Uncertainty

Recall that, in first-order logic and propositional logic, knowledge is represented by certainty. This means that we were sure about the predicates. So, if we say, $A \rightarrow B$, then it indicates that if A is true, then B is also true. But what if there is a situation where we are not sure about whether A is true or not? How will we express this statement? This situation is known as uncertainty.

To represent uncertain knowledge, where we are not sure about the predicates, we need uncertain reasoning or probabilistic reasoning. However,

before going to probability reasoning, let us first look at some causes of uncertainty in the real world.

1. Information from unreliable sources
2. Experimental errors
3. Equipment fault
4. Temperature variation
5. Climate change

In AI, we use probability to indicate uncertainty in knowledge. In probabilistic reasoning, probability theory is combined with logic to handle uncertainty. For example, in real world, the statement, 'It will rain today', is a probable sentence as we can only assume that it will happen but are not sure about it. Thus, probabilistic reasoning can be applied in the following circumstances:

- When there are unpredictable outcomes.
- When specifications or possibilities of predicates become too large to handle.
- When an unknown error occurs during an experiment.

Now, before jumping into probabilistic reasoning to solve problems with uncertain knowledge, let us recapitulate some often-used key terms.

5.2.2 Probability

Probability can be defined as a chance that an uncertain event will occur. It indicates the likelihood that an event will occur. The value of probability is between 0 and 1. Here, probability = 0 means event will never happen and probability = 1 means that event will definitely take place. Technically, $P(A)$, that is, the probability of an event A is: $0 \leq P(A) \leq 1$

$P(A) = 0$, indicates total uncertainty and

$P(A) = 1$, indicates total certainty in an event A .

The probability of an uncertain event can be expressed as

$$\text{Probability of occurrence} = \frac{\text{Number of desired outcomes}}{\text{Total number of outcomes}}$$

- $P(\neg A)$ = probability of A not happening event.
- $P(\neg A) + P(A) = 1$.

Event: Each possible outcome of a variable is called an event. For example, when a coin is tossed, you can either get a head (event 1) or a tail (event 2).

Sample space: The collection of all possible events is called sample space.

Prior probability: The prior probability of an event is probability computed before observing new information.

Posterior probability: This refers to the probability that is calculated after all evidence or information has been collected. It combines prior probability and new information.

Independent Event: It is an event which is not affected by other events. For example, tossing a coin as each toss of a coin is not affected by any toss in the past. The probability of getting a head will always be 1-in-2, or 50%.

Dependent Events: These events can be affected by previous events. For example, if there are 2 blue and 3 red marbles in a bag, then the chances of getting a blue marble is 2 in 5. But after taking one red marble earlier, the chance of taking a blue marble next is 2 in 4. Similarly, had we taken a blue marble before, the chance of a blue marble next is 1 in 4. Since the next event depends on what happened in the previous event, it is called a dependent event.

Conditional probability is the possibility of an event based on the existence of a previous event. It is calculated by multiplying the probability of the preceding event with that of the succeeding event.

The conditional probability of event A is the probability that event A will occur given that another event B has already occurred. There are two

types of events—dependent events or independent events. Conditional probability for these events can be calculated as follows:

$$P(A|B) = P(A \text{ and } B) / P(B),$$

if A and B are two dependent events.

If A and B are independent events, then conditional probability is given by, $P(A|B) = P(A)$

Marginal probability: The unconditional probability of an event P(A) is known as marginal probability. This is not dependent on the occurrence of any other event. For example, the probability that a card drawn from a deck of cards is red ($P(A) = 0.5$), and the probability that the card is drawn is 8 ($P(B)=1/13$). Both these events are independent events since the outcome of one event is not affected by the occurrence of the other.

Joint Probability: A joint probability is the probability of event A and event B happening, P(A and B) together. Actually, it is the likelihood of the intersection of two or more events and thus, written as intersection of A and B, that is, $P(A \cap B)$. For example, the likelihood that a card drawn from a deck of cards is red and numbered 8 is equal to $P(A \text{ and } B) = 2/52 = 1/26$. (Since we have two Black-8 in a deck of 52).

Conditional probability: Conditional probability is the probability of occurring an event when another event has already happened. This can be expressed as

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

In the above formula, we are calculating the event A when event B has already occurred. Here,

$P(A \cap B)$ is the joint probability of A and B

$P(B)$ = Marginal probability of B.

Example: In a class, 70% of the students want to study R Language and 40% of the students want to learn Python Programming. What is the percent of students who want to study R as well as Python?

Solution:

Let A be an event that a student wants to study Python.

B is an event that a student wants to study R.

$$P(A | B) = \frac{P(A \wedge B)}{P(B)} = \frac{0.4}{0.7} = 57\%$$

Hence, 57% are the students who want to study R also like Python.

5.3 Bayes' Theorem

Bayes' theorem, also known as **Bayes' rule**, **Bayes' law** or **Bayesian reasoning**, determines the probability of an event with uncertain knowledge.

In probability theory, Bayes' rule is related to the conditional probability and marginal probabilities of two random events. It facilitates computing the value of $P(B | A)$ when the value of $P(A | B)$ is known. Moreover, the probability prediction of an event can be easily updated when new information is added.

For example, if cancer corresponds to one's age, then we can determine the probability of cancer in a patient more accurately by using Bayes' theorem.

Bayes' theorem can be derived using product rule and conditional probability of event A with known event B. Thus, we can write:

$$P(A \wedge B) = P(A | B) P(B) \quad (i)$$

or

Similarly, the probability of event B with known event A:

$$P(A \wedge B) = P(B | A) P(A) \quad (\text{ii})$$

Equating right-hand side of equations (i) and (ii), we get,

$$P(A | B) = \frac{P(B | A) P(A)}{P(B)} \quad (\text{iii})$$

Equation (iii) is known as **Bayes' theorem**. It is fundamental to most modern AI systems for **probabilistic inference**. Here,

$P(A | B)$ is **posterior probability** and will be read as probability of hypothesis A when an evidence B has occurred.

$P(B | A)$ is the likelihood. If the hypothesis is true, we calculate the probability of evidence.

$P(A)$ is the **prior probability**, that is, the probability of hypothesis before considering the evidence.

$P(B)$ is called **marginal probability**, pure probability of an evidence.

We can also write equation (iii) as

$$P(A_i | B) = \frac{P(A_i) * P(B | A_i)}{\sum_{i=1}^k P(A_i) * P(B | A_i)}$$

where $A_1, A_2, A_3, \dots, A_n$ is a set of mutually exclusive and exhaustive events.

Bayes' rule can also be used to perceive the effect of some unknown cause. In such a situation, Bayes' rule becomes:

$$P(\text{cause} | \text{effect}) = \frac{P(\text{effect} | \text{cause}) P(\text{cause})}{P(\text{effect})}$$

Bayes' theorem is extensively used in AI to calculate the next step of the robot when the already executed step is given. It is also used in weather forecasting.

Example: What is the probability that a patient with a stiff neck is suffering from meningitis, given that

Meningitis causes a patient to have a stiff neck 80% of the time?

The known probability that a patient has meningitis is 1/30,000.

The known probability that a patient has a stiff neck is 2%.

Solution: Let A be the proposition that a patient has stiff neck.

Let B be the proposition that a patient has meningitis.

So, $P(A | B) = 0.8$.

$P(B) = 1/30000$.

$P(A) = .02$.

$$P(B | A) = \frac{P(A | B) P(B)}{P(A)} = \frac{0.8 * (1/30000)}{0.02} = 0.0013333$$

Hence, 1 patient out of 750 patients has meningitis disease with a stiff neck.

Example: From a standard deck of playing cards, a single card is drawn. The probability that the card is king is 4/52. Find out the posterior probability $P(\text{King} | \text{Face})$, which means the drawn face card is a king card.

Solution:

Let A be the probability that card A is a king.

Let B be the probability that card B is a face card.

$P(B | A)$ be the probability of face card being a king.

We know that

$$P(A) = 4 / 52 = 1 / 13$$

$$P(B) = 1 / 13$$

$$P(B | A) = 1$$

So,

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)} = \frac{1 * (1 / 13)}{3 / 13} = 1 / 3$$

Hence, probability that a face card is a king card = 1 / 3.

5.3.1 Bayesian Belief Network in Artificial Intelligence

In statistics, probabilistic models are used to define the relationship that exists between variables and the probabilities of each variable. However, in real-world problems, a number of variables is just too many. In such a scenario, a large amount of data is required to cover each case of probability functions. Calculating this data in real time is another big issue.

Therefore, we need a model that preserves the conditional dependencies between random variables and conditional independence in other cases. These requirements from a model are fulfilled by Bayesian networks, which help us to effectively visualize the probabilistic model for each domain and study the relationship between random variables in the form of a user-friendly graph.

Bayesian belief network, also known as a **Bayes network**, **belief network**, **decision network** or **Bayesian model**, is an important concept in computing that deals with probabilistic events to solve a problem which has uncertainty. We can define a Bayesian network as a probabilistic graphical model which represents a set of variables and their conditional dependencies using a directed acyclic graph.

The Bayesian network graph does not contain any cycles. Hence, it is known as a directed acyclic graph or DAG.

Bayesian networks are said to be probabilistic because they are created using probability distribution. Moreover, they use probability theory for prediction and anomaly detection. Since real-world applications are probabilistic in nature, we need a Bayesian network to represent the relationship between multiple events. For example, in various tasks including prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction and decision-making under uncertainty, Bayesian networks are used extensively.

A Bayesian network is created using available data and experts' opinions. It is made of two parts as follows:

- Directed acyclic graph (DAG) (refer **Fig. 5.1**)
- Table of conditional probabilities

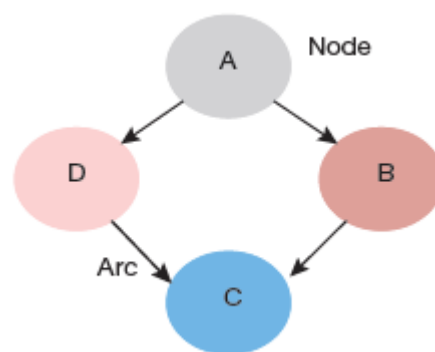


FIGURE 5.1 Directed acyclic graph (DAG)

A Bayesian network graph has multiple nodes and each node is connected to another node with an arc (or directed link). Here, the following points are noted.

1. Each **node** represents a random variable which can either be **continuous** or **discrete**.
2. **Arc or directed arrows** connect a pair of nodes in the graph and are used to depict the causal relationship between random variables.
3. An arc also indicates that one node is directly influencing the other node. In case there is no directed link between two nodes, then it simply means that nodes are independent with each other.

The generalized form of Bayesian network that represents and solves decision problems under uncertain knowledge is known as an **influence diagram**.

In **Fig. 5.1**, A, B, C and D are random variables represented by the nodes in the graph. If node B is connected to node A by a directed arrow, then node A is called the parent of node B. Note that node C is independent of node A.

- Each node in the Bayesian network has condition probability distribution $P(X_i | \text{Parent}(X_i))$, which determines the effect of the parent on that node.
- Bayesian network is based on joint probability distribution and conditional probability.

5.3.2 Joint Probability Distribution

If we have variables $x_1, x_2, x_3, \dots, x_n$, then the probabilities of a different combination of $x_1, x_2, x_3, \dots, x_n$, are known as joint probability distribution. That is, in terms of the joint probability distribution, $P[x_1, x_2, x_3, \dots, x_n]$, can be written as,

$$\begin{aligned} &= P[x_1 | x_2, x_3, \dots, x_n] P[x_2, x_3, \dots, x_n] \\ &= P[x_1 | x_2, x_3, \dots, x_n] P[x_2 | x_3, \dots, x_n] \dots P[x_{n-1} | x_n] P[x_n]. \end{aligned}$$

In general, for each variable X_i , the equation can be given as,

$$P(X_i | X_{i-1}, \dots, X_1) = P(X_i | \text{Parents}(X_i)).$$

5.3.3 Understanding the Semantics of Bayesian Network

There are two ways to understand the semantics of the Bayesian network.

1. To understand the network as the representation of the joint probability distribution. This will be helpful to understand the construction of the network.
2. To understand the network as an encoding of a collection of conditional independence statements to design the inference procedure.

We have seen that a Bayesian belief network is a graphical representation of different probabilistic relationships among random variables in a particular set. It classifies without any dependency on attributes. This means that the classifier is condition-independent.

The probability in a Bayesian belief network is specified using the condition, $P(\text{attribute}/\text{parent})$, which gives the probability of an attribute, true over parent attribute.

Local Markov Property

Bayesian networks satisfy the Local Markov Property which states that a node is conditionally independent of its non-descendants (given its parents). For example, if $P(D | A, B)$ is equal to $P(D | A)$ because D is independent of its non-descendant, that is, node B . This property very useful in simplifying the joint distribution.

Example of Bayesian Networks

Assume that you need to model a student's marks (m) for an exam. Consider the Bayesian network graph given below that shows that marks depend upon two other variables—exam level and IQ level—which can be explained as,

exam Level (e) is a discrete variable which specifies the difficulty of the exam. Its value is either 0 (for easy) or 1 (for difficult).

The IQ Level represents the intelligence quotient level of a student. It is also a discrete variable and has a value 0 (for low) or 1 (for high). The IQ level of the student influences the aptitude Score of the student (s).

A student's admission to a particular university depends on the marks obtained and is given as follows:

In **Fig. 5.2**, conditional probabilities table or CPTs are given that represent the probability distribution values of the given variables. Note the following:

- The sum of the CPT values in each row is equal to 1.

- A Boolean variable having k parents, and 2^K probability values. That is, if there is only 1 parent, then there will be $2^1 = 2$ values in CPT. If there are 2 parents, then there are $2^2 = 4$ values in CPT, and so on.

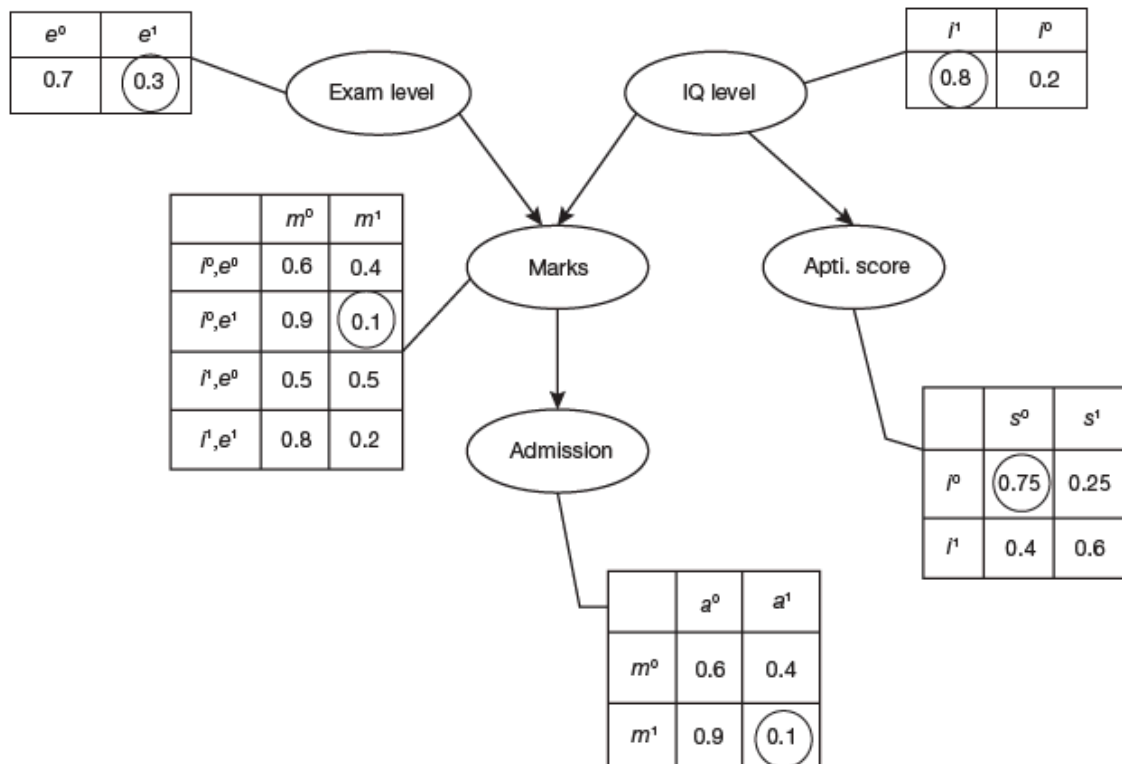


FIGURE 5.2 Conditional Probabilities for Admission Process

Let us first enumerate our five variables.

1. Exam level (e)
2. IQ level (i)
3. Aptitude score (s)
4. Marks (m)
5. Admission (a)

To calculate the joint probability distribution of the five variables, the formula is given by

$$P[a, m, i, e, s] = P(a \mid m) \cdot P(m \mid i, e) \cdot P(i) \cdot P(e) \cdot P(s \mid i)$$

which clearly indicates the following:

1. $P(a \mid m)$ denotes the conditional probability of the student getting admission based on the marks scored in the examination.

2. $P(m \mid i, e)$ represents the marks obtained by the student is determined by his/her IQ level and difficulty level of the exam.
3. $P(i)$ is the probability of the IQ Level a.
4. $P(e)$ is the probability of the exam level.
5. $P(s \mid i)$ is the conditional probability of the student's aptitude score, given his IQ level.

Now, let us use the probabilities to calculate the joint probability distribution of the entire Bayesian network. We will consider two cases here.

Case 1: Calculate the probability when the exam level is difficult, the student has a low IQ level and a low aptitude score, but still manages to pass the exam and secure admission to the university.

Here, the joint probability distribution can be given as follows:

$$P[a=1, m=1, i=0, e=1, s=0]$$

Look at the conditional probability tables and note down the values for the given conditions.

Probability when $a = 1$ and $m = 1$ is 0.1,

m is dependent on i and e , and as per the question, we have to find the value given: $m = 1, i = 0, e = 1$. This probability = 0.1.

Since IQ level is low ($i = 0$), the value of probability for i is 0.8.

The exam level is difficult, so, $e = 1$. The probability for $e = 1$, will be 0.3.

The IQ level is low, thus score = 0. Hence, the value is 0.75.

Now, the joint probability can be given as

$$= 0.1 * 0.1 * 0.8 * 0.3 * 0.75 = \mathbf{0.0018}.$$

Case 2: Calculate the probability that a student has high IQ level and aptitude score, the exam was easy but could not manage to pass the exam.

The formula for the JPD in this case is

$$P[a=0, m=0, i=1, e=0, s=1].$$

Looking at the tables, we get

No admission because of less marks ($a=0, m=0$) = 0.6

Less marks despite high IQ and easy exam ($i=1, e=0$) = 0.5

Exam easy = 0.7

High IQ ($i=1$) = 0.2

Good score ($s=1$) = 0.6

$$= 0.6 * 0.5 * 0.2 * 0.7 * 0.6$$

$$= 0.0252.$$

5.4 Learning

Herbert Simon defined learning as changes in a system that enable a system to do the same task more efficiently the next time.

To understand the term 'learning' in the context of computers, let us go back to 1997 when Mitchell proposed, 'A computer program is said to learn from experience 'E' with respect to some class of tasks 'T' and performance measure 'P', if its performance at tasks in 'T', as measured by 'P', improves with experience E '.

Taking the concept of learning in computers to the next level, machine learning, is defined as a branch of computer science that is specially dedicated to study and design algorithms that can produce predicates from the given dataset.

Machine learning facilitates computers to learn without being explicitly programmed. The agents learn not only to perform a particular task but also to improve their future performance.

The following tasks must be learned by a machine learning agent.

1. Predict the result state for an action.

2. Know the values for each state and interpret which state has high or low value.
3. Record all relevant precepts.

Why do we require machine learning?

1. Machine learning helps in improving and understanding the efficiency of human learning.
2. Machine learning is used to discover new things that were otherwise not known to us.

5.4.1 Components of a Learning System

The main components of a learning system include the following.

1. ***Performance Element:*** It is the agent that acts in the world. The action is performed by interpreting (or precepting) events in real world and then deciding what actions need to be taken.
2. ***Learning Element:*** The task of this element is to make improvements in the system by accepting knowledge about the performance element and providing feedback to determine ways to make changes in the performance element.
3. ***Critic:*** As the name suggests, this element tells the learning element how the agent is doing. For this, it compares the agent's performance with the fixed standard of performance.
4. ***Problem Generator:*** This component suggests problems or actions that will generate new examples or experiences. This additional knowledge would help the system to further train itself.

For example: Self-driving cars

5. ***Performance Element:*** It contains knowledge about procedures for driving. For example, knowledge about turning, applying breaks and accelerating.
6. ***Learning Element:*** It formulates goals. For example, the learning element learns the procedure to apply breaks, accelerate the car, take a safe turn, learn the geography of the city, etc.
7. ***Critic:*** It observes real-world scenario and communicates the information to the learning element. For example, the element observes the

reaction of other drivers when a sharp turn is taken by the car.

8. **Problem Generator:** It generates a new problem like 'try the congested north city road'.

5.4.2 Forms of Learnings

There are different types of learnings which are discussed below.

5.4.2.1 Rote Learning

Rote learning has the following features.

1. It is based on memorization.
2. It ignores the inner complexities.
3. The learner just recalls the stored knowledge.

For example, human learners learn a poem by repeatedly reciting it without understanding its actual meaning.

5.4.2.2 Induction Learning

Induction learning has the following features.

1. It is also known as learning by example.
2. It is based on supervised learning.
3. In the learning process, a general rule is induced by the system from a set of observed instances.
4. Class definitions are constructed using any classification technique.

For example, if ' f ' is a function of input x , then $f(x)$ denotes the output function applied to ' x '.

To find the hypothesis h such that $h \approx f$, we can plot three graphs.

Consider **Fig. 5.3** (a) that shows points (x,y) given in a plane so that $y = f(x)$. Our job is to find a function $h(x)$ that fits the point well.

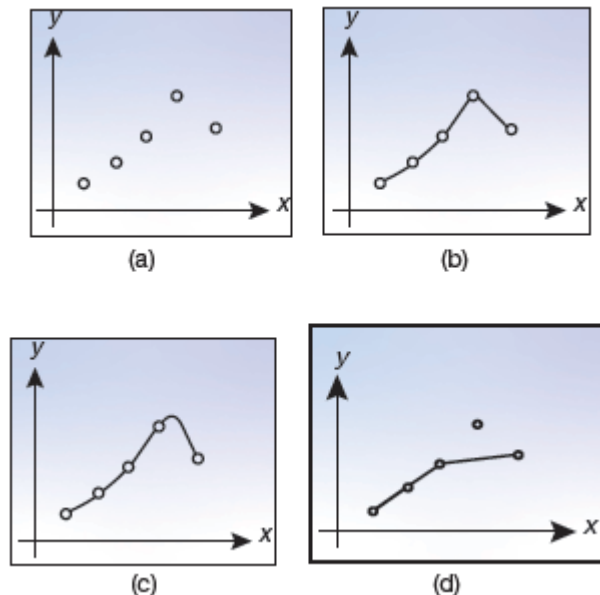


FIGURE 5.3 Plotting graphs for $y = f(x)$ to find a solution that best fits the points

Note that in **Fig. 5.3 (b)**, a linear function ‘h’ is connecting the points. In **Fig. 5.3 (c)**, a more complicated function is used. So, even though input points are the same, the output is different. **Figure 5.3 (d)** is different from the graphs given in **Figs 5.3 (b)** and **5.3 (c)** as it completely points and fits others with a simple function.

5.4.2.3 Learning By Taking Advice

This type of learning has the following features.

1. It is the easiest way of learning.
2. It follows a simple technique.
3. A programmer writes a program to give some instructions to perform a task. Once it is programmed or learned, the system will perform its tasks.
4. We human learners take advice from several sources including other humans (experts), the Internet, etc.
5. This type of learning works by deriving inference rather than just doing rote learning.
6. With more and more stored knowledge getting transformed into an operational form, the reliability of the knowledge source improves.

5.4.2.4 Explanation-Based Learning (EBL)

This type of learning has the following features.

1. It deals with an idea of single-example learning.
2. It requires a substantial number of training instances.
3. The main issue is that having a number of training instances is difficult.
4. It helps to learn certain things effectively, specially when sufficient knowledge exists.
5. While instance-based learning is data-driven and thus data-intensive, EBL, on the other hand, is more knowledge-intensive and knowledge-driven.
6. The process of EBL can be explained in the following manner:
 1. Initially, an EBL system accepts a training example.
 2. The operability criteria and domain theory are analysed. This includes a set of rules that describe relationships between objects and actions in a domain.
 3. The training example is then generalized to describe the goal concept and to satisfy the operability criteria.
 4. Thus, several applications are possible for knowledge acquisition and engineering aspects.

5.4.2.5 Reinforcement Learning (Problem Solving)

We have already seen in **Chapter 1** that reinforcement learning is a feedback-based machine learning algorithm in which an agent learns to act in a restricted environment. For this, it performs certain actions and then observes the results of those actions. For each correct action, the agent gets a reward and for each bad action, the agent gets negative feedback or a penalty.

Although the program designer sets the reward policy, no hint or suggestion is given to the learner about how to solve the task at hand. The learner has to figure out how to perform the task in such a way that as to get the maximum reward. For this, it starts from totally random trials and ends with sophisticated tactics and superhuman skills.

Reinforcement learning is the best way to enhance a machine's creativity. When run on a powerful parallel computer infrastructure, machines can be made to learn from the experiences of thousands of other learners learning parallelly. Let us quickly summarize the concept of problem solving before going further.

1. Humans prefer to learn by solving various real-world problems.
2. Reinforcement learning is based on this form of solving real-world problems.
3. We know that repeating certain actions results in a desirable outcome but avoiding those actions can lead to undesirable outcomes.
4. Therefore, we need to evaluate outcomes using a utility function. The utility function is the worth of a particular outcome.
5. There are several research issues which include the identification of the learning rate, time and complexity of the algorithm, convergence, representation (frame and qualification problems), handling of uncertainty (ramification problem), adaptivity and 'unlearning', etc.

In reinforcement learning, the system knows the desirable outcomes but does not know which actions should be performed to get those outcomes.

In such a case, the effects of performing the actions are usually compounded with side-effects. This makes it impossible to specify the actions to be performed in accordance to the given parameters.

Q-Learning is the most widely used reinforcement learning algorithm. It uses a simple value iteration update technique. For each state 'S', from the state set S, and for each action, a, from the action set 'A', it is possible to calculate an update to its expected reduction reward value, with the following expression:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t(s_t, a_t)[r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

where

r_t is a real reward at time t ,

$\alpha_t(s, a)$ are the learning rates such that $0 \leq \alpha_t(s, a) \leq 1$ and

γ is the discount factor such that $0 \leq \gamma < 1$.

5.2.4.6 Rote Learning

We all know that rote learning completely ignores the underlying complexities in the subject. It just focuses on memorizing the content so that the learner can recall it exactly in the same way it was learnt (read or heard).

Learning can be done either by memorization or repetition. In memorization, the learner just retains the concepts learnt without understanding them. In repetition, the learner says the same thing again and again to remember it by heart. For example, a child recites a poem multiple times to retain it.

5.2.4.7 Chunking Memory

Chunking is more efficient than normal memorizing. Here, chunking refers to the organization of information. Chunking is the process of encoding memories into long-term memory for later recall. It helps to get above the problem of short-term memory. According to a research study by George Miller in 1956, a person can handle between five and nine pieces of information at the same time.

Miller also introduced the term ‘chunking’ to help people retain more pieces of information.

For example, let us look at our phone number, 001 – 725-258-8901. This American phone number is quite long. To memorize this number, we can break it in chunks and then learn those chunks by associating a meaning with it. Here, 001 is the country code, 725 is the area code and the last seven digits is the actual number.

Teachers and parents also use chunking to help students memorize information in their long-term memory. In our daily lives, we use chunking; for example, to recognize a person’s face, we use several stored pieces of information about that face.

A related concept called cognitive loading theory states that the amount of information and interactions that must be processed simultaneously can either underload or overload the finite working memory. If the memory is overloaded, then all the elements must be processed to learn the entire concept. More the information a person has to learn in a shorter period of time, higher is the difficulty to process that information.

Richard E. Mayer, an educational psychologist, has stated that people learn better when a complex continuous lesson is broken into separate segments. For example, a complex figure can be broken into two or more smaller figures. Each smaller figure should present a different part of the original one. In this way, the learner's working memory is less likely to be overloaded with essential processing.

Jerome Bruner proposed the concept of spiral learning which refers to the idea of revisiting basic ideas. Each time an idea is revisited, it is elaborated further.

5.2.4.8 Inductive Learning

There are two types of inductive learning—supervised and unsupervised.

Supervised Learning

In this type of learning, the machine has access to a teacher who corrects it. The machine learning algorithm uses an inference function to train it using labelled training data. The training data consist of an input-output pair. For example, if we input the image of a fruit, the machine learning algorithm must be able to identify whether that fruit is an apple or mango.

Unsupervised Learning

Contrast to supervised learning, unsupervised learning does not have any access to the teacher. Instead, the machine must itself search for 'order' and 'structure' in the environment. There is no desired output in this learning technique. Therefore, categorization is done so that the algorithm correctly differentiates between two (or even more) different entities. Unsupervised learning is usually done when there is no labelled

data. Here, we do not know which entity belongs to which category. For example, we do not know whether a person is suffering from COVID or not. We just see certain symptoms and then categorize that the person is COVID+. Similarly, the machine does not know whether a person will buy a new product or not. The machine learning algorithm just accepts data about the new customer and then categorizes him/her as a potential buyer by comparing his/her details with those who had bought that product in the past. If the new customer's data seem to be similar with existing customers, then he/she is a potential buyer; else not.

5.5 Clustering

Clustering is the process of grouping together objects in a single cluster that have similar properties. Objects in a cluster are similar to each other and objects in different clusters vary significantly from each other.

There are different algorithms for clustering data. For example, hierarchical clustering specifies the hierarchy of clusters and depicts the relationship that exists between two clusters. Distance-based clusters can be used to find out similarities between two data points as closer data points are more similar than those further away. Similarly, density-based clustering defines clusters based on the density of data sets and centroid-based clustering forms clusters by finding the nearest cluster centre (which may not be a data point).

Clustering is better than classification as it is adaptable to changes and identifies features that distinguish different groups.

Irrespective of the technique used for clustering, labels are assigned to each cluster. Note that clusters can vary depending on the clustering algorithm used. So, it is always recommended to perform multiple analyses by running different clustering algorithms on the same data to get better results.

Clustering, which is an example of unsupervised learning, is used when the target features are not given in the training examples. The technique

constructs clusters which can then be used to predict feature values for the examples in the class.

Different clustering algorithms give different prediction errors. But, at the end, the best clustering is technique, that minimizes the error, is chosen.

Example of Clustering: In college, your teacher may want to cluster students' learning behaviour so that he/she can implement different strategies to deal with different sets of students.

5.5.1 Clusters

Clustering is a set of techniques used to partition data into a number of groups, also called clusters. A cluster can be considered as a group of data objects that are similar to other objects in the same cluster and dissimilar to data objects in other clusters. Clustering helps data scientists identify two main qualities of data—meaningfulness and usefulness.

Meaningful clusters enhance domain knowledge. For example, in the healthcare industry, clustering is used to make groups of patients whose bodies responded differently to the same medical treatment.

Useful clusters serve as an intermediate step in a data pipeline. For example, businesses use this technique for making different clusters of customers depending on their purchasing patterns to create targeted advertising campaigns.

5.5.2 Overview of Clustering Techniques

Data objects can be clustered using various techniques. Each of these techniques has its own strengths and weaknesses. Depending upon the input data, we need to select the most appropriate technique that results in more natural cluster assignment. To select the most suitable clustering algorithm for the dataset is not a trivial task. Some important factors that affect this decision include the characteristics of the clusters, the features of the dataset, the number of outliers and the number of data objects.

Based on these features, three popular categories of clustering algorithms include the following:

1. Partitional clustering
2. Hierarchical clustering
3. Density-based clustering

Partitional Clustering

It divides data objects into non-overlapping groups (refer **Fig. 5.4**) in such a way that every object belongs to one and only one cluster, and every cluster must have at least one object.

In such algorithms, users can specify the number of clusters (indicated by the variable k). These algorithms iteratively assign subsets of data points into k clusters. Two examples of partitional clustering algorithms are k -means and k -medoids. Note that these algorithms are **nondeterministic**. That is, such algorithms produce different results from two separate runs even if the same data was used in every run.

Advantages

1. It works well when clusters have a **spherical shape**.
2. **It is scalable** with respect to algorithm complexity.



FIGURE 5.4 Partitional Clustering

Limitations

1. It does not perform well with **complex cluster shapes** having different sizes.
2. It does not work with clusters of different **densities**.

Hierarchical Clustering

It creates clusters in a hierarchy either by using a bottom-up or a top-down approach.

In agglomerative clustering, a bottom-up approach is used to merge two points that are the most similar. This process is repeated until all points have been merged into a single cluster.

In divisive clustering, a top-down approach is used in which the algorithm starts with all points as one cluster (refer **Fig. 5.5**). In each iteration, the data objects that are the least similar in the cluster are split until only single data points remain.

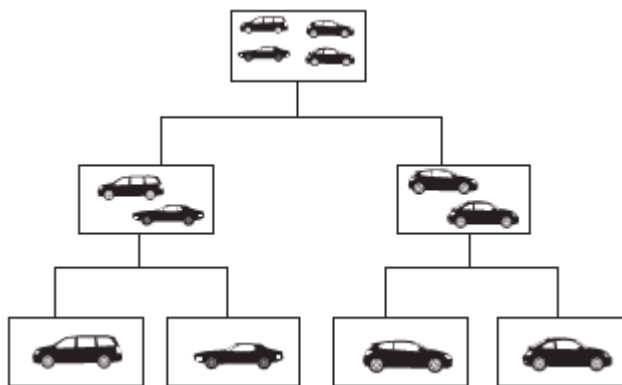


FIGURE 5.5 Divisive Clustering

In such an algorithm, a tree-based hierarchy of points called a **dendrogram** is created. Number of clusters to be created, that is, k is determined by the user. Clusters are created by cutting the dendrogram at a specified depth that results in k groups of smaller dendrograms.

Hierarchical clustering is a **deterministic** process. Therefore, cluster assignments do not change when the algorithm is executed two or more times using the same input data.

Advantages

1. They highlight the **relationships** between data objects at a finer level.
2. Results are easily **interpretable**.

Limitations

1. They're **computationally expensive** with respect to algorithm complexity.
2. They are adversely affected by **noise** and **outliers** in the data set.

Density-Based Clustering

This technique creates clusters based on the density of data points in a region. A cluster is created where there are high densities of data points separated by low-density regions (refer **Fig. 5.6**).

In contrast to other clustering categories, density-based approach does not require the user to specify the number of clusters. It uses a distance-based parameter that acts as a threshold. The threshold value helps the algorithm to determine how close points must be to be considered a cluster member.

Density-Based Spatial Clustering of Applications with Noise (or DBSCAN) and Ordering Points to Identify the Clustering Structure (or **OPTICS**) are popular examples of density-based clustering algorithms.

Advantages

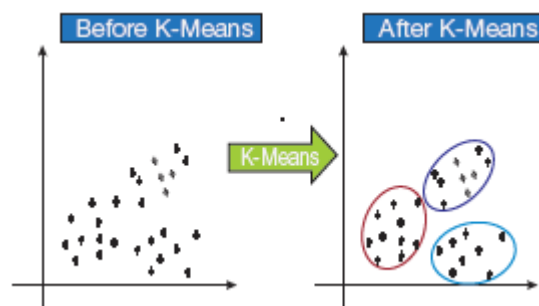


FIGURE 5.6 Density-Based Clustering

1. It works well when clusters are of **non-spherical shapes**.
2. It performs well even on a data set having noise or outliers.

Limitations

1. It is not well suited for clustering in **high-dimensional spaces**.
2. It is difficult to identify clusters of **varying densities**.

5.6 Explanation-Based Learning (EBL)

Explanation-based learning learns from a single training instance. With that instance and a functional description, EBL builds a general structural description (refer **Fig. 5.7**). An EBL system accepts four kinds of input described as follows:

1. A training example which states what the learning *sees* in the world.
2. A goal concept is a high-level description that states what the program is supposed to learn.
3. An operational criterion specifies a description of concepts that are usable. It states which other terms appear in the generalized result.
4. Domain theory contains a set of rules that describe relationships between objects and actions in a domain.

With these four inputs, the EBL computes a generalization of the training example that describes while satisfying the operational criterion. This process is completed in two steps.

Step 1: Explanation: The domain theory is used to prune all irrelevant aspects (with respect to the goal concept) of the training example.

Step 2: Generalization: The explanation obtained in Step 1 is generalized as far as possible in such a way that it still describes the goal concept. Generalization ensures that all cases are covered and not just specific ones.

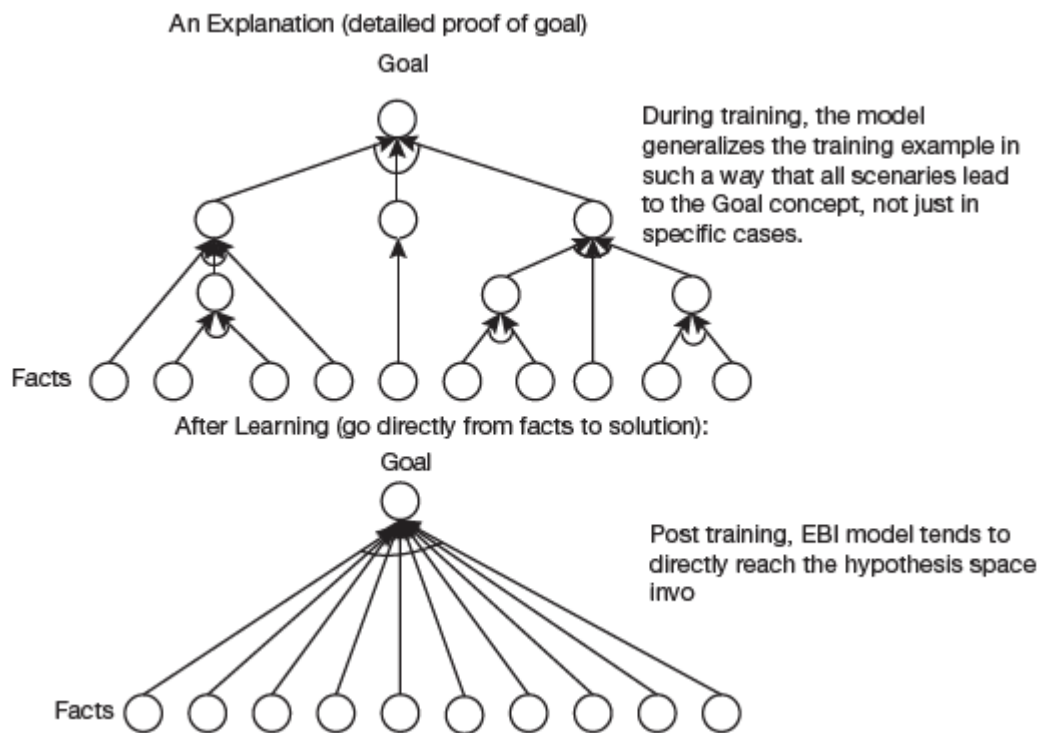


FIGURE 5.7 Explanation-Based Learning

Therefore, we see that EBL uses an explicit **domain theory** to generate an **explanation** or **proof** of a training example. The proof is then generalized and the new knowledge acquired is then applicable to even non-training situations.

EBL differs from **inductive** learning as new knowledge is added in the domain theory. Therefore, EBL is sometimes called **deductive learning** or **analytic learning**.

Example: Learning that an Object is a Mug

Target Concept: $\text{mug}(C) \text{ :- premise}(C)$.

Domain Theory:

```
mug(X) :- liftable(X),
        holds_liquid(X).
holds_liquid(Z) :- component(Z,W),
                  OpenCylinder(W),
                  points_up(W).
liftable(X) :- less_weight(X), component(Y,handle).
less_weight(X) :- small(X).
```

We will use knowledge specified in the domain theory to determine when some object is called a mug. We want to derive an explicit rule that specifies when something is identified as a mug.

Training Example: An Instance of a mug

```
mug(obj1) small(obj1) component(obj1,handle)
has(Krish,obj1) component(obj1,bottom)
      component (obj1, OpenCylinder)
points_up(OpenCylinder)
looks_like(OpenCylinder)
color(obj1,yellow)
```

Proof (or Explanation): First, we will prove that **obj1** is a mug as shown in **Fig. 5.8**.

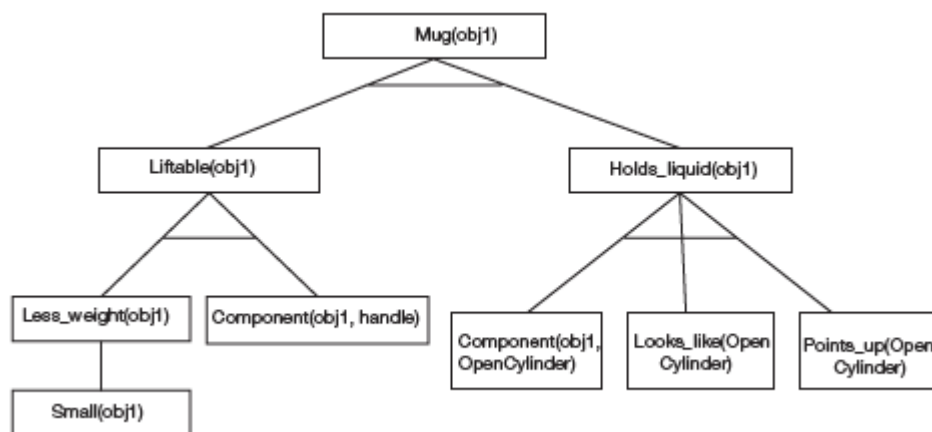


FIGURE 5.8 Proof of **obj1** is a mug.

Next: Generalize the proof to: X is a mug. In this step, we generalize all constants that depend on the training example. In our training example (refer **Fig. 5.9**), **OpenCylinder** and **obj1** are the constants.

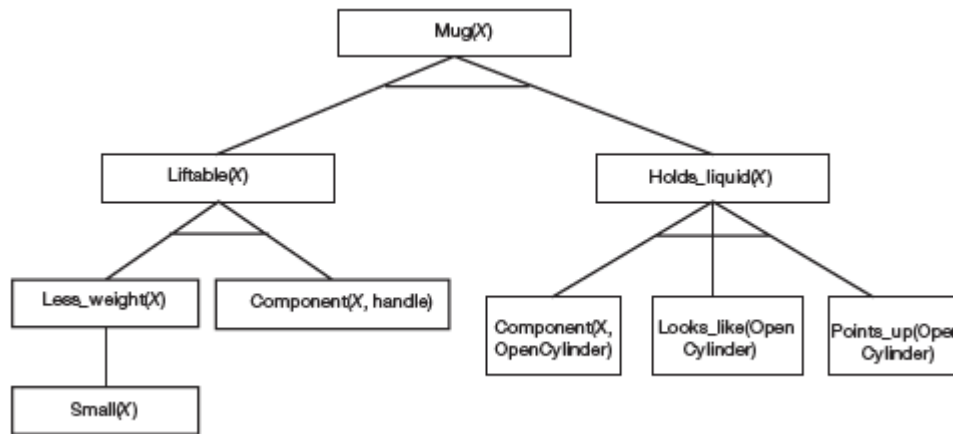


FIGURE 5.9 Generalize the proof to: X is a mug

Next: Add the new *chunk of knowledge* to the domain theory.

```

mug(X) :- small(X),
          component(X,W),
          looks_like(W),
          points_up(W).
  
```

‘Pure’ EBL only learns how to solve faster, not how to solve problems previously insoluble.

Note that we have ignored all irrelevant information in the training example while formulating the proof or adding new knowledge to the domain theory. Once added, the new rule(s) can be used to identify any similar object as a mug.

Thus, the EBL agent learns by examining particular situations and relating them to gained knowledge base. The agent uses this gained knowledge to solve similar types of problems. Let us now try to understand the architecture of EBL shown in **Figs 5.10** and **5.11**.

The **problem solver** takes inputs from the environment. These inputs include a training example, inference rules and the goal concept. The problem solver processes these inputs and gives justification to generalizer. This consists of the following areas:

Goal concept is the problem statement stated in a complex form. The EBL agent needs to learn it.

Training examples are facts which explain an instance of the goal concept.

Inference rules represent the facts that the learner already knows.

The generalizer accepts the output of the problem solver as its input. Its job is to compare the explanation of the problem solver with the knowledge base. The output of this comparison is passed to the operability pruner.

The operability pruner accepts the output of the generalizer and the operability standard as inputs. The operability standard describes the final concept and specifies the form in which learned concept should be expressed.

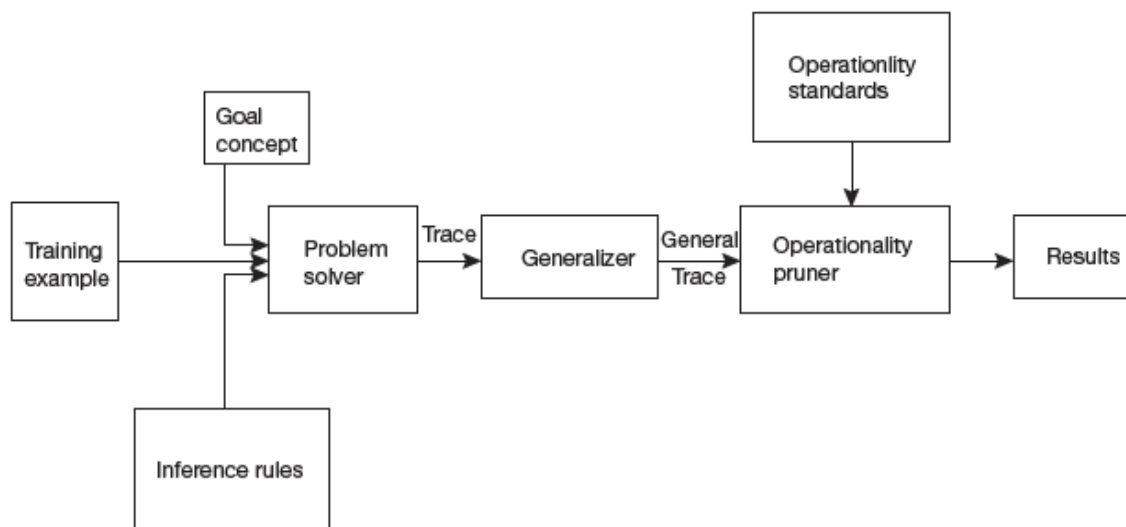


FIGURE 5.10 Architecture of EBL

Example: Consider the domain theory and the facts given below to make the EBL learner to identify that $\text{Robot}(r) \wedge \text{R2D2}(r) \rightarrow \text{Robust}(r)$

DOMAIN THEORY

An individual that can fix itself is robust

A habil individual that can see another entity can fix that entity

All robots can see themselves

R2D2-class in individuals are habil

$Fixes(x,x) \Rightarrow Robust(x)$

$Sees(a,b) \wedge Habile(a) \Rightarrow Fixes(a,b)$

$Robot(x) \Rightarrow Sees(x,x)$

$R2D2(a) \Rightarrow Habile(a)$

FACTS

Robot(Num5)

R2D2(Num5)

Age(Num5,5)

GOAL

Robust(Num5)

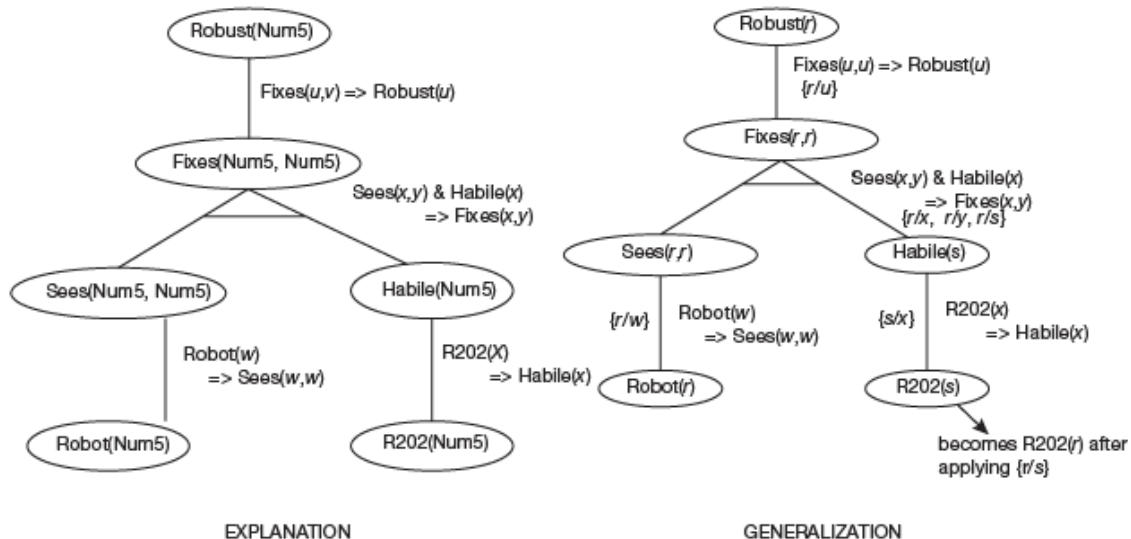


FIGURE 5.11 Architecture of EBL based on given Domain Theory

5.7 Reinforcement Learning

Reinforcement learning is an area of machine learning that facilitates machines to find the best possible behaviour or path that should be taken in a specific situation. Reinforcement learning is different from supervised learning. While in supervised learning, the training data has labels that are used to train the models, reinforcement learning, on the other hand, is used when no labels exist. The agent decides actions to be performed to complete the given task. In the absence of a training dataset, it is bound to learn from its experience.

Example: The agent wants to take the reward, but there are many hurdles in between. The agent has to find the best possible path to reach the reward. In **Fig. 5.12**, note that our agent is a robot, the reward is a diamond and the hurdle is fire. The robot learns by trying all the possible

paths and then selects the one which gives it the reward with the least hurdles.

Each step correctly taken will get the robot, a reward. Correspondingly, a wrong step will subtract the reward of the robot. The total reward is calculated when the robot reaches the diamond.

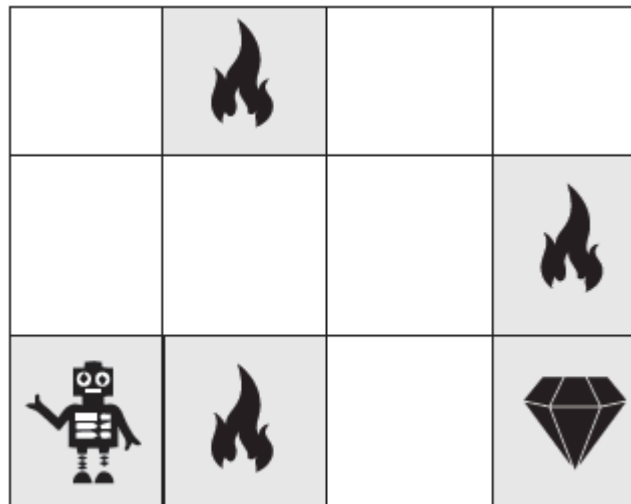


FIGURE 5.12 The Agent and the Reward

5.7.1 Components of Reinforcement Learning

A reinforcement learning system contains the following components.

1. **Input:** The input of a reinforcement learning system is an initial state from which the agent starts.
2. **Output:** Since there can be different solutions to a particular problem, a reinforcement learning system can have many possible outputs.
3. **Training:** Training depends on input. The agent returns a state and the user observes the output to decide whether the agent should get a reward or be punished.

Apart from these, we have the following components as well (as shown in **Fig. 5.13**):

1. **Agent or the learner:** An entity which performs actions in an environment to gain some reward is called an agent. The agent perceives

and explores the environment before acting upon it.

FIGURE 5.13 Components of Reinforcement Learning

2. **Action:** They are the moves taken by an agent in the environment.
3. **Environment:** This is the scenario in which the agent interacts and performs actions. The environment is assumed to be random in nature.
4. **Reward (R):** An immediate return given to the agent on performing a correct action or task is a reward. It acts as feedback to evaluate the action of the agent.
5. **State (s):** It refers to the current situation of the environment after an action is taken by the agent.
6. **Policy (π):** It is a strategy applied by the agent in the current state to decide the next action.
7. **Value (V):** It specifies long-term return with discount. It does not cater to short-term reward.
8. **Value Function:** It is the function which returns the value of a state. This value returns the total amount of reward. The agent is expected to start from this state.
9. **Model of the environment:** It is a model that mimics the actual behaviour of the environment. It helps to draw inferences as it helps to observe how the environment will behave in a particular scenario.

The agent continues to learn until it finds the best solution that gives the maximum reward.

5.7.2 Example: Reinforcement Learning in Real World

When you bring home a pet, you train it to perform actions in an acceptable manner. How do you do that?

Since our pet does not understand our language, we cannot directly instruct what has to be done and how it has to be done. So, we emulate a situation in which the pet has to act in different ways. If the pet responds in the desired way, we give it a piece of flesh as the reward, else frown at it to make it understand that the act was undesirable.

After receiving the reward or the stern looks, the pet learns that if it performs a particular set of actions again in future in a similar type of environment, then it will get a reward. So, the pet performs more enthusiastically the next time by learning from positive experiences. At the same time, it also learns what not to do when faced with negative experiences.

Here, the following can be noted:

1. The pet is an agent that interacts and acts in the environment.
2. The environment is your house.
3. An example of a state could be the pet sitting in a corner and you using a specific word for the pet to walk.
4. The agent reacts by performing an action. This results in transition from one state to another. For example, the pet goes from sitting to walking.
5. The policy is a method of selecting an action.
6. After the transition, the pet may get a reward or penalty in return.

5.7.3 Characteristics of Reinforcement Learning

1. There is no supervisor. Only rewards direct the actions to achieve outcomes.
2. It involves sequential decision making
3. It is based on the hit and trial method.
4. Time is an important factor in reinforcement problems.
5. The agent performs action and changes states based on the feedback of the previous action.

6. Feedback is often delayed and is not available right at that instant.
7. What data the agent will receive depends on the agent's actions.
8. The environment is stochastic (random), so the agent needs to explore it to get the maximum positive rewards.

5.7.4 Differences Between Reinforcement Learning and Supervised Learning

The differences between reinforcement learning and supervised learning are given in **Table 5.1**.

TABLE 5.1 Differences between Reinforcement learning and Supervised learning

5.7.5 Types of Reinforcement

Reinforcement learning (RL) is the science of making decisions and learning the optimal behaviour in an environment. This optimal behaviour is learned by continuously interacting with the environment and observing how it responds. The process of discovering an optimal solution is like a trial-and-error search. There are two types of reinforcement—positive and negative.

Positive Reinforcement

An event that occurs due to a particular behaviour and increases the strength and the frequency of the behaviour, thereby causing a positive

effect on it is known as positive reinforcement.

Advantages

1. Maximizes performance
2. Sustains change for a long period of time

Disadvantage

Too much reinforcement may result in overloaded states which can diminish the results.

Negative Reinforcement

An event that occurs due to a particular behaviour and increases the strength of the behaviour by stopping or avoiding a negative condition is known as negative reinforcement.

Advantages

1. Enhances behaviour
2. Helps you define the minimum standard of performance
3. Gives performance better than the minimum standard of performance
4. It just provides enough to meet up the minimum behaviour

Reinforcement learning algorithms can also be categorized as model-free and model-based.

Model-Free Algorithms

These algorithms do not build an explicit model of the environment. They are based on trial-and-error algorithms that run experiments with the environment using actions to directly derive optimal policy. Model-free algorithms are either value-based or policy-based.

Value-Based Algorithms

Value-based algorithms formulate optimal policy by estimating the value function of every state accurately. The aim is to maximize a value function $V(s)$ as the agent is expecting a long-term return of the current

states under policy π . The value function depends on the reward as, without reward, there could be no value. Thus, the values are estimated to achieve more rewards.

Some popular value-based algorithms are SARSA and Q-learning.

Policy-Based Algorithms

Policy-based algorithms compute the optimal policy without modelling the value function. It parametrizes the policy using learnable weights to render the learning problem into an explicit optimization problem. Popular policy-based Reinforcement Learning algorithms include Monte Carlo policy gradient (REINFORCE) and deterministic policy gradient (DPG). These approaches suffer from a high variance which results in instability during the training process.

Policy defines how an agent behaves at a given time. It maps the perceived states of the environment to the actions taken on those states.

In a policy-based RL method, the aim is to formulate a policy in such a way that the action performed in every state helps to gain maximum reward in the future. Two types of policy-based methods are as follows:

1. **Deterministic:** For any state, the same action is produced by the policy π .
2. **Stochastic:** Every action has a certain probability. And this probability will determine the action.

Hybrid algorithms combine value-based and policy-based approaches. Value-based approaches are more stable but are not suitable to model continuous action spaces. To gain the advantages of both the approaches, hybrid algorithms preferred. One of the most powerful RL algorithms—the actor-critic algorithm—is an example of a hybrid algorithm. In this algorithm, both the policy (actor) and the value function (critic) are parametrized to effectively use training data along with stable convergence.

Model-based reinforcement learning, however, focuses on the creation of a virtual model for each environment. This helps the agent to learn to perform in that specific environment. Note that there is no particular solution or algorithm for this approach because the model representation is different for each environment.

With the help of a model, inferences can be made to know how the environment will behave. For example, given a state and an action, a model can predict the next state and reward.

The model is usually used for planning as it facilitates taking up a series of actions by considering all future situations before actually experiencing those situations.

5.7.6 Applications of Reinforcement Learning

Applications of reinforcement learning include the following:

1. Robotics for industrial automation.
2. Aircraft control and robot motion control
3. Planning a business strategy
4. Machine learning and data processing
5. Design training systems, providing customized instruction and materials according to student's requirements

5.7.7 When to Use Reinforcement Learning in a Large Environment?

Reinforcement learning can be used in large environments in the following situations:

1. The environment is known, but an analytic solution is not known
2. Only a simulated environment is available.
3. The only way to collect information about the environment is to interact with it.

5.7.8 Benefits of Reinforcement Learning

Reinforcement learning is applicable to a wide range of complex problems that cannot be handled with other machine learning algorithms.

Reinforcement learning algorithms **seek a long-term goal** while exploring different possibilities autonomously, so they are said to be closer to artificial general intelligence (AGI).

Reinforcement learning algorithms focus on the problem as a whole. Unlike traditional machine learning algorithms that excel at specific sub-tasks, RL algorithms see the big picture. They do not divide the problem into subproblems. Rather, their goal is to maximize long-term rewards and for this, they may trade-off short-term rewards for long-term benefits.

RL algorithms **do not need a separate data collection step**. They obtain data through direct interaction of the agent with the environment. So, training data is available through the learning agent's experience and no additional data has to be fed to the algorithm.

RL algorithms work well in dynamic and uncertain environments as they are inherently adaptive and can quickly respond to changes in the environment. Unlike traditional machine learning algorithms, time and agent's experience are important factors in RL algorithms.

5.7.9 Challenges with Reinforcement Learning

Although RL algorithms can successfully solve complex problems in diverse simulated environments, their adoption in the real world has been slow due to the following reasons.

Needs Extensive Experience to Train the Agent

The data is collected when the agent interacts with the environment. Therefore, the rate of data collection is limited by the dynamics of the environment. Environments with high latency slow down the learning curve.

Finding a good solution requires extensive learning and exploration in complex environments with high-dimensional state spaces.

The learning agent has to trade off short-term rewards for long-term gains. But in situations where the outcome is unknown, until a large num-

ber of sequential actions are taken, it may be difficult for the agent to discover the optimal policy. For example, in a game of chess, the outcome is unknown until both players have made all their moves.

RL Algorithms Lack Interpretability

We know that an agent once trained to work with optimal policy and deployed in the environment, takes actions based on its experience. To an external observer, the reason for these actions might not be clear. This lack of interpretability hampers the trust between the agent and the observer. If an observer can explain the actions taken by the agent, then the problem can be understood in a better way and the limitations of the model can be easily discovered, especially in high-risk environments.

Other miscellaneous challenges include the following:

1. Parameters may affect the speed of learning.
2. Real-world environment may have partial observability and may be constantly changing.
3. Too much reinforcement at times results in overload of states that affects the results.

5.7.10 Future of Reinforcement Learning

Traditionally, reinforcement learning algorithms could perform only one task at a time and each task is learned by a separate agent. These agents do not share knowledge. This makes learning complex behaviour (like driving a car) inefficient and slow.

Since certain problems share a common information source, have related underlying structure, and are interdependent on each other, they can get a huge performance boost if multiple agents work together and share their information with each other. For example, A3C or Asynchronous Advantage Actor-Critic is an algorithm in which related tasks are learned concurrently by multiple agents. Such multi-task learning scenarios will ultimately drive reinforcement learning closer to AGI.

Moreover, the trend is towards deep reinforcement learning that uses deep neural networks to model the value function (value-based) or the agent's policy (policy-based) or both (actor-critic). With deep neural networks, relevant features are automatically generated during the training process, thereby allowing the agent to learn optimal policies in complex environments.

While in RL, complex features had to be engineered to train the algorithm, there is no such limitation in deep RL. So, reduced learning capacity, limiting the scope of RL to simple environments is no longer an issue with deep neural networks.

5.7.11 When Not to Use Reinforcement Learning?

Although reinforcement learning model is a promising technique, it is not suitable for all situations. For example, you should not consider using reinforcement learning in the following situations:

1. When there is not enough data to solve the problem with a supervised learning method
2. When there is limited time and computing resources as reinforcement learning algorithms require heavy computing resources and are time-consuming.



Perform k-nn on the iris data set

OUTPUT

Implementing Decision Tree Algorithm on Iris Data Set

OUTPUT

Implement Random Forest Algorithm on iris Data Set

OUTPUT

Download the Wholesale customers data from <https://archive.ics.uci.edu/ml/machine-learning-databases/00292/Wholesale%20customers%20data.csv> and perform k-means++ algorithm for clustering. Use elbow method to decide the optimum number of k. Also use the inertial value to evaluate metrics.

OUTPUT

Use the different values for weather and temperature to predict whether a child should go out to play out or not using Naïve Bayes.

OUTPUT

Write a program to implement Multi-layer Perceptron (MLP) on a simple data matrix

OUTPUT

[1 0]

Perform SVM on breast cancer data

OUTPUT

 KEY TERMS

Inference: It is the process of generating new information. It is an automatic process that has no goals.

Reasoning: The mental process of deriving logical conclusion and making predictions from available knowledge, facts and beliefs is called reasoning.

Deductive reasoning: It is a strategic approach that uses available facts, information or knowledge to draw valid conclusions.

Probability: It is a chance that an uncertain event will occur. It indicates the likelihood that an event will occur.

Event: Each possible outcome of a variable is called an event.

Sample space: The collection of all possible events is called sample space.

Conditional probability: It is the possibility of an event based on the existence of a previous event.

Joint Probability: It is the probability of event A and event B happening, $P(A \text{ and } B)$ together.

Bayesian network: It is a probabilistic graphical model which represents a set of variables and their conditional dependencies using a directed acyclic graph.

Machine learning: It is a branch of computer science that is specially dedicated to study and design algorithms that can produce predicates from the given dataset.

Clustering: It is the process of grouping together objects in a single cluster that are similar properties.

Cluster: It is a group of data objects that are similar to other objects in the same cluster and dis-similar to data objects in other clusters

Environment: It is a scenario in which the agent interacts and performs actions. The environment is assumed to be random in nature.

State (s): It refers to the current situation of the environment after an action is taken by the agent.

Policy (π): It is a strategy applied by the agent in the current state to decide the next action.

Value Function: It is the function which returns the value of a state. This value returns the total amount of reward. The agent is expected to start from this state.

Model of the environment: It is a model that mimics the actual behaviour of the environment. It helps to draw inferences as it helps to observe how the environment will behave in a particular scenario.



RECAPITULATION POINTS

- The term 'reasoning' is associated with thinking, intellect and cognition that directs the process of generating one idea from the other.

- A reasoning system is targeted or is directed to achieve the goal of deducing an inference.
- Inductive reasoning, also known as cause-effect reasoning, generates hypothesis rather than drawing conclusion(s) to the facts at the beginning of the process.
- Common sense reasoning helps machines simulate human ability to make decisions and perceptions about real-world events.
- Monotonic reasoning uses facts, information and knowledge to draw a conclusion. Non-monotonic reasoning labels a conclusion as invalid if, on addition of new or more information, a previously generated conclusion becomes contradictory.
- Abductive reasoning starts with an incomplete set of facts, information and knowledge to find the most appropriate explanation and conclusion.
- Bayes' theorem determines the probability of an event with uncertain knowledge.
- Bayesian networks help us to effectively visualize the probabilistic model for each domain and study the relationship between random variables in the form of a graph.
- Bayesian networks satisfy the Local Markov property which states that a node is conditionally independent of its non-descendants (given its parents).
- A computer program is said to learn from experience 'E' with respect to some class of tasks 'T' and performance measure 'P', if its performance at tasks in 'T', as measured by 'P', improves with experience E.
- Rote learning is based on memorization, ignores the inner complexities and the learner just recalls the stored knowledge.
- Cognitive loading theory states that the amount of information and interactions that must be processed simultaneously can either underload or overload the finite working memory.
- Explanation based learning (EBL) uses an explicit domain theory to generate an explanation or proof of a training example.
- *Value-based algorithms* formulate optimal policy by estimating the value function of every state accurately. The aim is to maximize a value function $V(s)$ as the agent is expecting a long-term return of the current states under policy π .



1. Define reasoning.
2. Give the basic features of a reasoning system.
3. Write a short note on common-sense reasoning.
4. Differentiate between the following:
 1. Deductive and inductive reasoning
 2. Monotonic and non-monotonic reasoning
 3. Supervised and unsupervised learning
 4. Agglomerative and divisive clustering
 5. Reinforcement learning and supervised learning
 6. Positive and negative reinforcement
5. Explain the different types of reasoning.
6. Why do we need probabilistic reasoning?
7. What are Bayesian networks?
8. Give the main components of a learning system.
9. What does the cognitive loading theory state?
10. Discuss the categories of clustering algorithms.
11. What are the different components of reinforcement learning?
12. Give the merits and demerits of reinforcement learning.
13. Consider the following sentences in the knowledge base.

Birds can fly. Penguins cannot fly. Pitty is a bird.

Can you conclude that Pitty can fly?

It becomes an example of non-monotonic reasoning. Justify this statement.
14. Identify the type of reasoning and explain it in detail.
 1. Premise: All of the pigeons we have seen in the zoo are white.
Conclusion: Therefore, we can expect all the pigeons to be white.
 2. Implication: Cricket ground is wet if it is raining
Axiom: Cricket ground is wet.
Conclusion It is raining.
15. Write a short note on the following:
 1. Conditional probability
 2. Bayes' theorem
 3. Machine learning

4. Clustering
5. Explanation-based learning
6. Model free model reinforcement model
16. A box contains three coins: two regular coins and one fake two-headed coin ($P(H)=1$).
1. You pick a coin at random and toss it. What is the probability that it lands heads up?
2. You pick a coin at random and toss it, and get heads. What is the probability that it is the two-headed coin?
17. In a group of 10 people, 4 people bought apples, 3 bought oranges and 2 bought apples and oranges. If a buyer chose randomly bought apples, using the conditional probability formula, find the probability they also bought oranges.
18. My neighbour has 2 children. I learn that she has a son, Adam. What is the probability that Adam's sibling is a boy?
19. A fair die is rolled. What is the probability of A given B where A is the event of getting an even number and B is the event of getting a number less than or equal to 2?
20. A die is rolled twice and two numbers are obtained. Let X be the outcome of the first role and Y be the outcome of the second roll. Given that $X + Y = 5$, what is the probability of $X = 4$ or $Y = 4$?
21. 70% of your friends like chocolate and 35% like chocolate and strawberry. What percent of those who like chocolate also like strawberry?
22. Two dies are thrown simultaneously, and the sum of the numbers obtained is found to be 7. What is the probability that the number 3 has appeared at least once?
23. In a group of 100 computer buyers, 40 bought CPUs, 30 purchased monitors and 20 purchased CPUs and monitors. If a computer buyer chose at random and bought a CPU, what is the probability they also bought a monitor?
24. In a group of 100 sports car buyers, 40 bought alarm systems, 30 purchased bucket seats and 20 purchased an alarm system and bucket seats. If a car buyer chosen at random bought an alarm system, what is the probability they also bought bucket seats?

25. Consider the contingency table given below.

What is the probability that a randomly selected person is male, given that they own a pet?

26. Harry installed a new burglar alarm at his home to detect burglary.

The alarm reliably responds at detecting a burglary but also responds for minor earthquakes. Harry has two neighbours—David and Sophia—who have taken responsibility to inform Harry at work when they hear the alarm. David always calls Harry when he hears the alarm, but sometimes, he got confused with the phone ringing and calls at that time too. On the other hand, Sophia likes to listen to high music, so sometimes, she misses to hear the alarm. Here, we would like to compute the probability of the burglary alarm.



Calculate the probability that the alarm has sounded, but there is neither a burglary, nor an earthquake occurred, and David and Sophia both called Harry.

1. _____ is a strategic approach that uses available facts, information or knowledge to draw valid conclusions.
2. Inductive reasoning generates _____.
3. Inductive and deductive reasoning are forms of _____ logic.
4. Common-sense reasoning is gained through _____.
5. Common sense works with _____ rules.
6. Vivekananda said, 'In a conflict between the heart and the brain, follow your heart'. is an example of _____ reasoning.
7. _____ reasoning is used in conventional reasoning logic-based system.
8. _____ reasoning is very useful for robot navigation.

9. In AI, we use _____ to indicate the uncertainty in knowledge.
10. _____ indicates the likelihood that an event will occur.
11. Each possible outcome of a variable is called an _____.
12. The collection of all possible events is called _____.
13. If A and B are independent events, then conditional probability is given by _____.
14. _____ theorem determines the probability of an event with uncertain knowledge.
15. _____ networks help us to effectively visualize the probabilistic model for each domain.
16. In a Bayesian network, each _____ represents a random variable which can either be _____ or _____.
17. Bayesian Networks satisfy the _____ property which states that a node is conditionally independent of its parents.
18. _____ element of a learning system tells the learning element how the agent is doing.
19. _____ is the most widely used reinforcement learning algorithm.
20. _____ learning is usually done when there is no labelled data.
21. Clustering which is an example of _____ learning.
22. _____ is a group of data objects that are similar to other objects in the same group and dissimilar to objects in another group.
23. Two examples of partitional clustering algorithms are _____ and _____.
24. The _____ is used to prune all irrelevant aspects of the training example.
25. _____ function returns the total amount of reward in reinforcement learning.
26. In reinforcement learning, the agent continues to learn until it finds the best solution that gives the maximum _____.
27. In _____ model, for any state, the same action is produced by the policy.
28. RL algorithms work well in _____ and _____ environments.

**STATE TRUE OR FALSE**

1. A reasoning system is not modified externally by a programmer.
2. A reasoning system cannot solve new and unusual problems.
3. A reasoning system improves using hit-and-trial technique.
4. Prediction is done by pattern matching.
5. Inductive reasoning starts from the general premises to get to specific conclusion(s).
6. In deductive reasoning, the arguments can be valid or invalid based on the rules.
7. Addition of new facts and knowledge in the database cannot alter the conclusion drawn at an earlier point in time.
8. Human's perceptions for various things in daily life is an example of non-monotonic reasoning.
9. Monotonic reasoning labels a conclusion as invalid on addition of new or more information.
10. In first-order logic and propositional logic, knowledge is represented with certainty.
11. The unconditional probability of an event $P(A)$ is known as joint probability.
12. Posterior probability $P(A | B)$ is read as probability of hypothesis A when an evidence B has occurred.
13. The Bayesian network graph may contain any cycles.
14. Machine learning is dedicated to study and design algorithms that can produce predicates from the given dataset.
15. Critics make improvements in the system by accepting knowledge about performance element and feedbacks.
16. While instance-based learning is data-driven, EBL is knowledge-driven.
17. In reinforcement learning, the system knows the desirable outcomes but does not know which actions should be performed to get those outcomes.
18. Chunking is less efficient than normal memorizing.
19. People learn better when a complex continuous lesson is broken into separate segments.

20. Unsupervised machine learning algorithm uses an inference function to train it using labelled training data.
21. Objects in a cluster are similar to each other and objects in different clusters vary significantly from each other.
22. Clustering is better than classification.
23. Divisive clustering follows a bottom-up approach.
24. Density-based approach does not require the user to specify the number of clusters.
25. EBL is also known as inductive learning.
26. In reinforcement learning, the environment is assumed to be random in nature.
27. Reinforcement learning should be used when the environment is known, but an analytic solution is not known.
28. Reinforcement learning divides the problem into subproblems.



MULTIPLE CHOICE QUESTIONS

1. _____ system is targeted or is directed to achieve the goal of deducing an inference.
 1. Knowledge representation
 2. Intelligent
 3. Reasoning
 4. Prediction
2. _____ is the mental process of deriving a logical conclusion and making predictions from available knowledge, facts and beliefs.
 1. Knowledge representation
 2. Intelligent
 3. Reasoning
 4. Prediction
3. In _____ reasoning, every result is based on the facts and ideas and is drawn using a top-down approach.
 1. Deductive
 2. Inductive
 3. Common sense
 4. None of these

4. Reasoning allows AI technologies to extract critical information from large sets of _____ data.
 1. Structured
 2. Unstructured
 3. Both of these
 4. None of these
5. Identify the incorrect statement.
 1. Deductive reasoning usually starts from the general premises to get to specific conclusion(s).
 2. Inductive reasoning does not draw conclusion(s).
 3. Inductive reasoning is preferred in cases with a limited set of facts.
 4. Deductive reasoning is fast and easy.
6. _____ reasoning helps machines to simulate human ability to make decisions and perceptions about real-world events.
 1. Deductive
 2. Inductive
 3. Common sense
 4. None of these
7. Any theorem proving is an example of _____ reasoning.
 1. Deductive
 2. Inductive
 3. Common sense
 4. Monotonic
8. _____ reasoning is mainly used in conventional reasoning systems and logic-based systems.
 1. Deductive
 2. Inductive
 3. Common sense
 4. Monotonic
9. Identify the incorrect statement about monotonic reasoning.
 1. Each proof will remain valid for a short duration.
 2. It cannot express real-world scenarios.
 3. Hypothesis knowledge cannot be expressed.
 4. New knowledge from the real world cannot be added.
10. _____ reasoning cannot be used for theorem proving.
 1. Deductive
 2. Non-monotonic

3. Common sense
4. Monotonic
11. Identify the incorrect statement about abductive reasoning.
 1. Conclusions are based on facts known at the present.
 2. It does not give any importance to outdated facts and information.
 3. It does not rectify the plausible conclusion.
 4. None of these
12. Probabilistic reasoning can be applied when _____.
 1. There are unpredictable outcomes.
 2. Specifications or possibilities of predicates become too large to handle.
 3. An unknown error occurs during an experiment.
 4. All of these
13. Bayesian network is based on _____ probability.
 1. Joint
 2. Conditional
 3. Both of these
 4. None of these
14. _____ element is the agent that acts in the world.
 1. Performance
 2. learning
 3. critic
 4. problem generator
15. Human learners learn a poem by _____ learning.
 1. Rote
 2. Induction
 3. Explanation Based
 4. Reinforcement
16. _____ learning is also known as learning by example.
 1. Rote
 2. Induction
 3. Explanation-based
 4. Reinforcement
17. Identify the incorrect statement about learning by taking advice.
 1. Derives inference rather than just doing rote learning.
 2. With more stored knowledge, the reliability of the knowledge source improves.

3. It is based on memorization.
 4. Programs are written to perform specific tasks.
18. _____ learning uses rewards and punishments to train the agent.
1. Rote
 2. Induction
 3. Explanation Based
 4. Reinforcement
19. Identify the incorrect statement about partial clustering.
1. Works well when clusters have a spherical shape.
 2. Scalable with respect to algorithm complexity.
 3. Performs well with complex cluster shapes having different sizes.
 4. Does not work with clusters of different densities.
20. OPTICS is a popular example of _____ clustering algorithms.
1. Partitional
 2. Hierarchical
 3. Density-based
 4. Agglomerative
21. Identify the incorrect statement about density-based clustering technique.
1. Works well when clusters are of non-spherical shapes.
 2. Performs well even on a data set having noise or outliers.
 3. Difficult to identify clusters of varying densities.
 4. Well suited for clustering in high-dimensional spaces.
22. _____ contains a set of rules that describe relationships between objects and actions in a domain.
1. Domain theory
 2. training example
 3. goal concept
 4. operational criterion
23. The _____ perceives and explore the environment before acting upon it.
1. Input
 2. Output
 3. Agent
 4. Environment
24. Identify the incorrect statement about reinforcement learning.

1. There is no supervisor. Only rewards direct the actions to achieve outcomes.
 2. Sequential decision-making
 3. It is based on hit and trial method.
 4. It cannot achieve artificial general intelligence (AGI).
25. If I put my hand in fire, it will burn. This is an example of _____ reasoning.
1. Deductive
 2. Non-monotonic
 3. Common sense
 4. Monotonic
26. The Earth revolves around the Sun. This is an example of _____ reasoning.
1. Deductive
 2. Non-monotonic
 3. Common sense
 4. Monotonic



Fill Up

1. Deductive reasoning
2. Hypothesis
3. propositional
4. experiences
5. heuristic
6. Monotonic.
7. Monotonic
8. Non-monotonic
9. Probability
10. Probability
11. event
12. sample space
13. $P(A | B) = P(A)$

14. Bayes
15. Bayesian
16. Node, continuous, discrete
17. Local Markov
18. critic
19. Q-Learning
20. Unsupervised
21. Unsupervised
22. cluster
23. k -means and k -medoids.
24. domain theory
25. Value
26. reward.
27. *Deterministic*
28. dynamic and uncertain

State True or False

1. True
2. False
3. True
4. True
5. False
6. True
7. True
8. True
9. False
10. True
11. False
12. True
13. False
14. True
15. False
16. True
17. True
18. False
19. True

20. False
21. True
22. True
23. False
24. True
25. False
26. True
27. True
28. False

Multiple Choice Questions

1. c
2. c
3. a
4. c
5. d
6. c
7. d
8. d
9. a
10. b
11. d
12. d
13. c
14. a
15. a
16. b
17. c
18. d
19. c
20. c
21. d
22. a
23. c
24. d
25. c

26. d

Previous chapter

< [4. Knowledge Representation](#)

Next chapter

[6. Computer Vision \(CV\)](#) >

Table of contents expanded

6

Computer Vision (CV)

Computer vision is a field of artificial intelligence that helps computers gain a high level of understanding from digital images or videos. It is done by acquiring images, processing them, analysing them and finally understanding the information they are conveying. If AI allows computers to think, computer vision helps them to see, observe and understand.

6.1 Human Vision vs Computer Vision

Computer vision works like human vision. The main advantage of human vision is that we have a brain that automatically learns to tell how far two objects are, whether they are moving or not, whether there is something wrong in an image and many more similar questions.

Computer vision techniques, in turn, train machines to perform these functions. The main challenge here is that machines have to analyse images coming from cameras using data and algorithms in a very limited time (refer **Fig. 6.1**). Once a system is trained to process thousands of images in a minute, it can quickly surpass human capabilities.

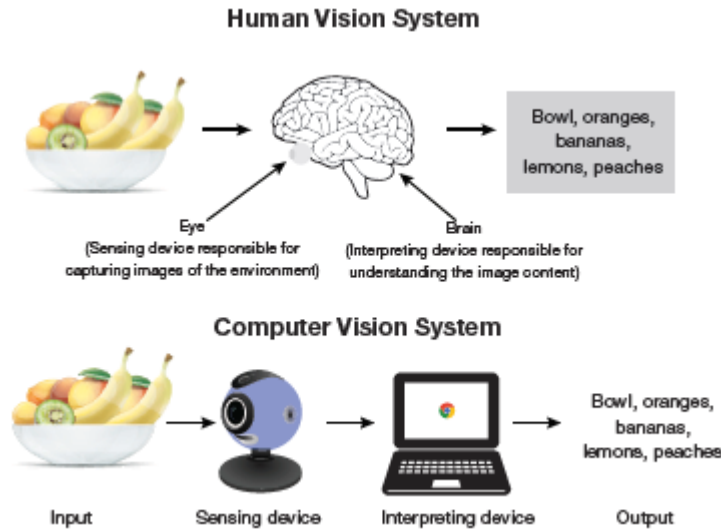


FIGURE 6.1 Human Vision vs Computer Vision

6.2 How Does Computer Vision Work?

Computer vision algorithms need large volume of data for analysing and recognizing images. For this, computer vision makes extensive use of deep learning (next level of machine learning) and Convolutional Neural Network (CNN).

Machine learning or deep learning algorithms help machines learn about the context of visual data and learn by itself how to identify and differentiate images. A CNN complements the machine learning or deep learning model by breaking images down into pixels that are given tags or labels. The CNN uses these labels to perform convolutions (a mathematical operation on two functions to produce a third function) to make predictions about what it is 'seeing.' Multiple iterations of the neural network are run and the accuracy of their predictions is evaluated. The process is repeated until the computer identifies the image with an acceptable accuracy. At this time, we say that the machine is recognizing or seeing images in a way similar to humans.

6.3 The Evolution of Computer Vision

Prior to deep learning/ machine learning, the performance of computer vision was very limited as a lot of manual coding and developer's efforts were required. For example, facial recognition could be done using the following steps.

Create a database of individual images of all the subjects to be tracked. Store them in a specific format.

Annotate images by entering key points/features of every individual image. For example, distance between the eyes, the width of nose bridge, distance between upper-lip and nose, etc.

Capture new images and repeat the measurement process to mark the key points on the image.

All this work was done manually. Automation comes into picture with the application comparing the measurements in the new image with the ones already stored in its database. The application matches the images and tells whether it corresponded with any of the profiles it was tracking. In this scenario, there was very little automation as most of the work was still being done manually. No doubt, possibility of error was also large in this context.

Machine learning and then deep learning revolutionized the process of solving computer vision problems. Now, developers had to no longer manually code every single rule into their vision applications. Rather, programs were written to extract features to detect specific patterns in images and ultimately classify those images in different categories as illustrated in **Fig. 6.2**.

6.4 Tasks in Computer Vision

Computer vision applications recognize things in photographs by using the following techniques.

Object classification to identify the broad category of object in the photograph.

Object identification to classify the type of a given object is in the photograph.

Object verification to confirm the presence of object in the photograph.

Object localization to locate the objects in the photograph. The goal of **localization** is to find the location of a single object in an image. Usually, the located object is enclosed with a rectangular box. For example, in **Fig. 6.3**, the cat is first classified and then localized. The standard way to perform localization is to define a bounding box enclosing the object in the image.

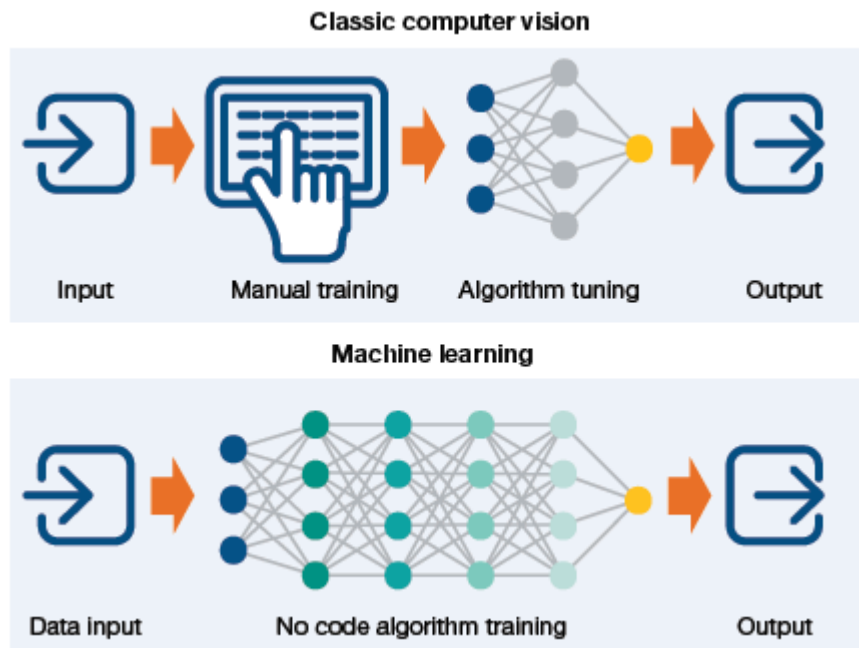


FIGURE 6.2 Classic Computer Vision vs ML Computer Vision

Object landmark detection to understand the key points for the object in the photograph.

Object detection or object identification task involves repeatedly locating and classifying all objects of interest in the image. The main aim here is to find and classify a *variable number of objects* in the given image.

Identifying unknown number of objects in a complex image is difficult for a human as some objects may be partially visible. These objects may be partly outside the frame, present in different sizes or may overlap each other. For example, object identification is extensively used to monitor real time images from security cameras to identify a specific person's face.

Object segmentation to enumerate the pixels that belong to the object in the image. **Instance or object segmentation** not only finds objects in the

image, but also creates a mask for each detected object that is as accurate as possible. It is extensively used by news channels to blur the face of a victim or any person whose identity is not to be revealed.

Object tracking inputs a video to track an object that is in motion over time. This is essentially useful in robotics (e.g., to stop a ball when a robot is used as a goalkeeper), in autonomous vehicles for spatial reasoning and path planning and in human tracking systems.



Credit: metamorworks / Shutterstock

FIGURE 6.3 Computer vision tasks

Object tracking applies object detection to capture the object being tracked and then discerns the movements of that object.

In addition, computer vision also includes methods for the following functions:

Video motion analysis to estimate the velocity of objects in a video, or the camera itself.

Scene reconstruction to create a 3D model of a scene inputted through images or video.

Image restoration uses machine learning-based filters to remove noise (to remove parts of image that are blurred).

To better realize the application of computer vision, play the game **Emoji Scavenger Hunt**.

6.5 How Computer Vision Works with Deep Learning?

With deep learning and CNN, the entire process of computer vision has been transformed. The following steps outline a general approach to building a computer vision model using CNNs.

Create a dataset of annotated images or use an existing one. Annotations include image classification, object detection (using bounding boxes), a pixel-wise instance segmentation of each object of interest present in the image.

From each image, **extract features** that are important for the current task. For example, extract facial features to recognize faces.

Train a deep learning model based on extracted features.

Evaluate the model using images that were not used in the training phase and compute the accuracy of the model.

This strategy comes under **supervised machine learning technique** as it requires a dataset from which the model can learn.

6.6 Applications of Computer Vision

Once the images are understood, computer vision is then used in the following areas.

1. **Facial recognition** systems to recognize faces in images and videos. Applications like Google photos, Snap chat, Facebook, Instagram, etc. use facial recognition system to identify people.
2. Almost every smart phone these days use facial-recognition algorithms (refer **Fig. 6.4**) to unlock phones. Facebook uses facial recognition to identify users in the posted pictures. In China, this technology is integrated with **payment technology** to allow users to do cashless

payment.



Credit: Fractal Pictures / Shutterstock

FIGURE 6.4 Facial-recognition algorithms

3. Some government agencies in different countries are also using facial technology for **surveillance**. Though the use has raised concerns among rights and privacy advocates, nonetheless, facial recognition techniques applications are on a constant rise.
4. **Content-based image retrieval systems** identify images by identifying image properties like composition, colour, texture, etc. Search engines like Google and Bing and software analysing CT scans and MRIs in hospitals, etc. used CBIR systems.
5. Computer vision is extensively used in smart interactions to give input to computers. This is especially more helpful in case of **gaming software** and in systems designed for differently-abled individuals.
6. Software used in home **security systems**, office security systems, drone-based surveillance systems, etc. (refer **Fig. 6.5**) also use computer vision techniques.
7. Computer vision is the heart of **augmented reality apps** that detects physical objects in real-time to place virtual objects within the physical environment.
8. Many agricultural organizations use computer vision techniques to monitor harvest and solve common agricultural problems including emergence of weeds or nutrient deficiency. Images collected from satellites, drones or planes are processed by computer vision algorithms to **detect potential problems** in the early phase, thereby avoiding un-

necessary losses at a later stage.

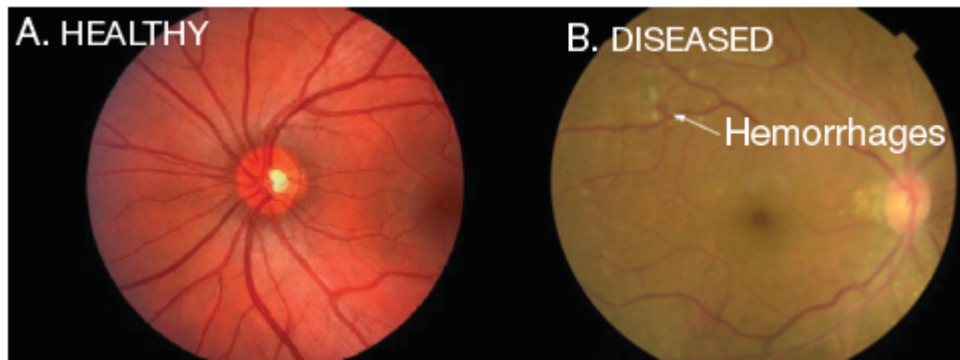


Credit: Robert Mandel / 123RF

FIGURE 6.5 Drones for Surveillance

9. **Self-driving cars (like Tesla)** use computer vision to learn from their surroundings. Every smart vehicle has cameras to capture videos from different angles. These videos act as an input to the computer vision software that processes them to detect objects like road marking, pedestrians or other cars, traffic lights, etc.
10. Websites like Facebook perform **content moderation** using computer vision techniques. Billions of posts posted by users every day are reviewed and any images/ videos that contain violence, extremism or pornography are immediately removed.
11. 90% of medical data is present in the form of images. So, most of the diseases can be easily diagnosed by processing images available in the form of **X-rays, MRI, mammography**, etc. For example, computer vision algorithms can detect diabetic retinopathy, which is the most prevalent cause of blindness (refer **Fig. 6.6**). Going further, these algorithms can also process pictures of the back of the eye and rate their severity (in case the disease is detected). Computer vision algorithms can even detect cancer with much higher precision than human doc-

tors.



Credit (Fig 6.6(a)): Timothy Mainiero / 123RF

FIGURE 6.6 Healthy Eye vs Diseased Eye

12. **CV In Augmented Reality (AR) and Mixed Reality (MR):** In AR and MR, computer vision enables computing devices (like smartphones, tablets and smart glasses) to overlay and embed virtual objects on real world imagery. For example, CV algorithms can help AR applications detect planes such as tabletops, walls and floors. Moreover, Snapchat Filters that overlay digital images onto your face is an application of AR. Another example of AR is Pokemon Go where users can walk around with their mobile phones and find Pokemon that are overlaid on the environment around the user.
13. **Inventory management:** CV algorithms can analyse images from security cameras to generate a **very accurate estimate of the items available in the store**. Results obtained through this analysis can then be used by store managers to detect when there is an unusual increase in demand to react early and efficiently.
14. **Manufacturing:** In this sector, most losses occur due to problems in machines or because of production of defective components. Computer vision algorithms can be used for **predictive maintenance**. Images obtained through cameras (probably fitted in the eyes of robots) can be analysed to detect potential problems they occur. CV can even be helpful to reduce defects. The system can identify defects in components throughout the entire production line to notify the manufacturers to take action in real time.
15. The system is trained to classify a defect as a serious defect or a mild defect. In case of a serious defect, further production is completely halted until the issue is resolved.

16. **Insurance:** CV is used in the insurance sector for processing claims. Clients can be asked to share images while documenting and claiming money. The CV system can analyse those images to estimate and adjust repair costs, determine if the insurance covers them and even check for possible fraud. In this way, clients get a better experience as the length of claim-cycle gets shortened due to automation.
17. A CV system can also be used to **avoid accidents** by preventing collisions when integrated into industrial machinery, cars, and drones. This provision will have a huge impact on the insurance industry.
18. **Defense and security:** CV systems are used in homes and companies with high security requirements (like banks) to **identify customers** based on analysing images from security cameras.
19. Such systems can also be used to improve cargo inspection at ports or for surveillance of sensitive places such as embassies, power plants, hospitals, railroads, and stadiums. In these places, computer vision algorithm analyses and classifies images, create a detailed description of a scene in real time to help decision-makers take right action at the right time. In defense, CV systems can also be used to identify enemy terrain or enemies in images.
20. **Visual search engines:** A **visual search engine retrieves images that meets the specified criteria**. For example, Google search engine allow users to present a source image so that it can find all images similar to it. Then, applications like **Google Lens**, can be used to obtain detailed information from images. For example, you can show an image of a dog to get information about its breed, or any other details.
21. In the e-commerce market, **Pinterest Lens** can be used to find new ways to wear the dresses. For example, when you present a photo of an item, Pinterest Lens will return outfit ideas that include compatible clothing items that you would like to buy.

Amazon Go Store uses Computer Vision technique to allow customers who have installed the Amazon Go application on their mobile devices to no longer stand in long queues in the super market. They just go to the store, pick up their items and come back without paying at the kiosks.

6.7 Which Vision Succeeded?

It is interesting to note that certain experiments were carried out in Germany to address the void between computer and human vision.

Experiment 1 was conducted to understand how neural networks perceive contours (or boundary). For this, both the human and the AI participant were shown an image and asked whether it has closed contour or not. The experiment was specifically done to check if deep learning algorithm can identify closed and open shapes or not.

Results showed that a well-trained neural network could identify a closed contour even when the contour was not made of straight lines.

Experiment 2 was performed to understand whether the deep learning algorithm could identify the relation between different shapes in the picture or not. While human vision could easily understand this, it was found that deep learning algorithms that were trained with less found it difficult to draw a conclusion.

Experiment 3 was specifically carried out to identify the recognition gap that strains computer vision technology to detect an object or a thing from closer proximity when it is zoomed. When we zoom in an image beyond a certain range, we are less likely to understand what it is. However, deep learning performed better than humans and could identify minuscule features that are imperceptible to the human eye and can be detected only when the image is zoomed in very closely.

6.8 Challenges in Computer Vision

Though computer vision has emerged as one of the top fields of machine learning, there are several obstacles that makes it difficult to become a leading technology. Human vision is a complicated and highly effective system which is difficult to replicate through technology.

Given below are certain issues and challenges faced by computer vision algorithms. These are related to the nature of the data or the application

to be created and its context. In this section, we will discuss these challenges to make the CV algorithms more robust and efficient.

1. **Real-time processing:** Many applications demand the computer system to analyse images and quickly take action without any time delay. Such systems need to be very accurate. Achieving this level of accuracy with specified time constraints is not easy to achieve.

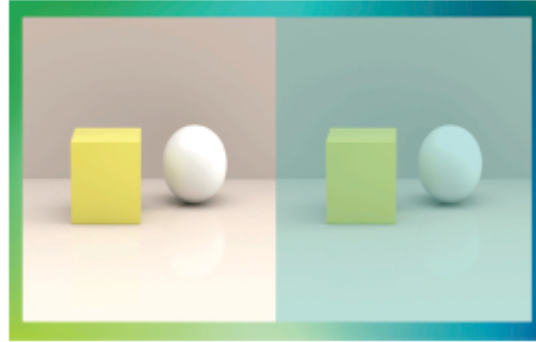


FIGURE 6.7 Color of the object changes when viewed in a different light

2. **Limited resources:** To successfully implement a computer vision algorithm, the computer system needs ample amount of memory, CPU and other computing resources.
3. **Different lighting:** Computer vision algorithms collect knowledge about the real-world objects in different kinds of lighting. For example, a white object in red light may look red in the image. Correspondingly, a red object under a red lamp becomes almost invisible (refer **Fig. 6.7**).
4. **Noise:** A noisy image or an image with incomplete/incorrect data makes it hard for computer vision to recognize objects. For example, when an image appears brighter or darker than they should be. Another example of noisy data could be images obtained from video-cams on the road are much less effective when it is raining or snowing outside.
5. **Unfamiliar angles:** An image must be fed in the system from several angles. Otherwise, a computer won't be able to recognize it if the angle

changes (refer **Fig. 6.8**).



Credit: Susan Schmitz / Shutterstock

FIGURE 6.8 Object viewed in different angles

6. **Overlapping:** When there are multiple objects on the image, they may overlap with each other and remain hidden, thereby making it difficult by machine to recognize them.
7. **Different types of objects:** Objects belonging to the same category may look totally different as shown in **Fig. 6.9**. This makes it very difficult for machine to successfully recognize them.
8. **Fake similarity:** Objects looking the same may actually belong to different categories. For example, samoyed puppies look similar to little polar bears in some pictures.



Credit: creativesunday / Shutterstock

FIGURE 6.9 Same object that looks different

9. **Adversarial Attacks:** An attacker can attack a machine learning model, to make the machine fail intentionally. These flawed models are difficult to identify and may affect the system adversely.

6.9 Understanding Image Pixels

Pictures displayed on a digital screen like a monitor or are simply in digital form are made up of thousands of tiny dots also known as pixels.

The word 'pixel' means 'picture element' indicating one single dot. In coloured images, pixels of different colors are arranged alongside each other to represent different shades of colors. This means that every digi-

tized image is broken up into tiny blocks with a description of a color value at each point. **Figure 6.10** shows a pixelated black and white image of an army officer.

Future of CV: Computer vision applications are growing at a very fast pace. Corporate giants like Google, Facebook, Apple and Microsoft are investing huge amounts in computer vision.

In a coloured image, each square, or pixel stores the value of just one of the three prime colours of red, green and blue (RGB).

As seen in **Fig. 6.10**, an image viewed on a screen is made up of pixels. Each pixel comprises three strips of red, green and blue. The intensity of these strips is changed to represent a colour. To be more specific, each pixel contains a series of numbers or digits, hence the term ‘digital’. These numbers dictate the colour values seen. However, when we see an image, we do not see these digits, we just see the colour that they represent.

Like RGB, CMYK is an image format that stores four values for each pixel. The pixel stores values of cyan, magenta, yellow, black.

Figure 6.10 shows the digits stored in each pixel in a black and white image as well as in a coloured image. The black-and-white image stores a ‘0’ for a black pixel and a ‘1’ for white pixel. This is known as a ‘bitmap’ image as it only requires one bit of information, 0 or 1, to describe the colour. But such a representation can be used with only two tones: black and white. Not even shades of grey can be represented using a bitmap. To represent more colours (grey shades), we usually use greyscale images that use a number from 0–255 to indicate the specific colour. A greyscale image can represent total 256 different colours.

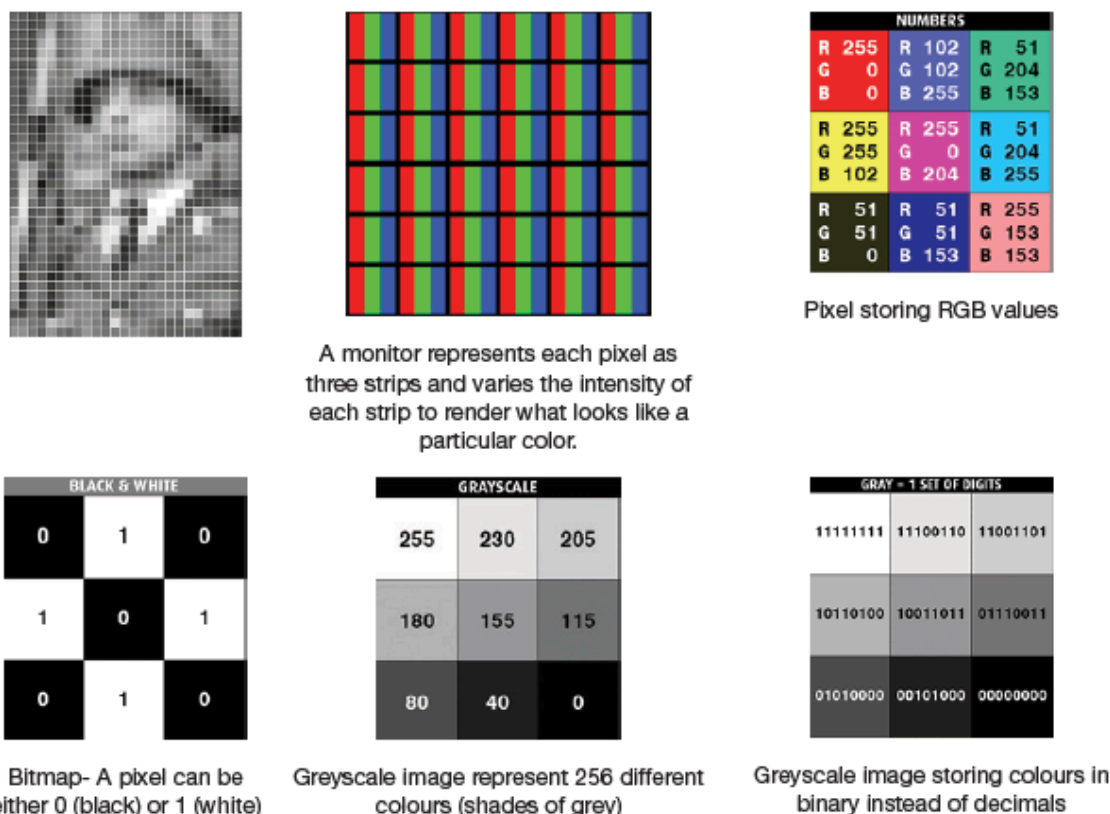


FIGURE 6.10 An Image is a composed of pixels

6.9.1 What is Resolution?

Resolution refers to the detail that an image holds. Higher the resolution, the more details the image stores and vice versa. More details mean better picture clarity. Correspondingly, less details mean less clarity. Image resolution is often measured in Dots Per Inch (DPI) and Pixels Per Inch (PPI).

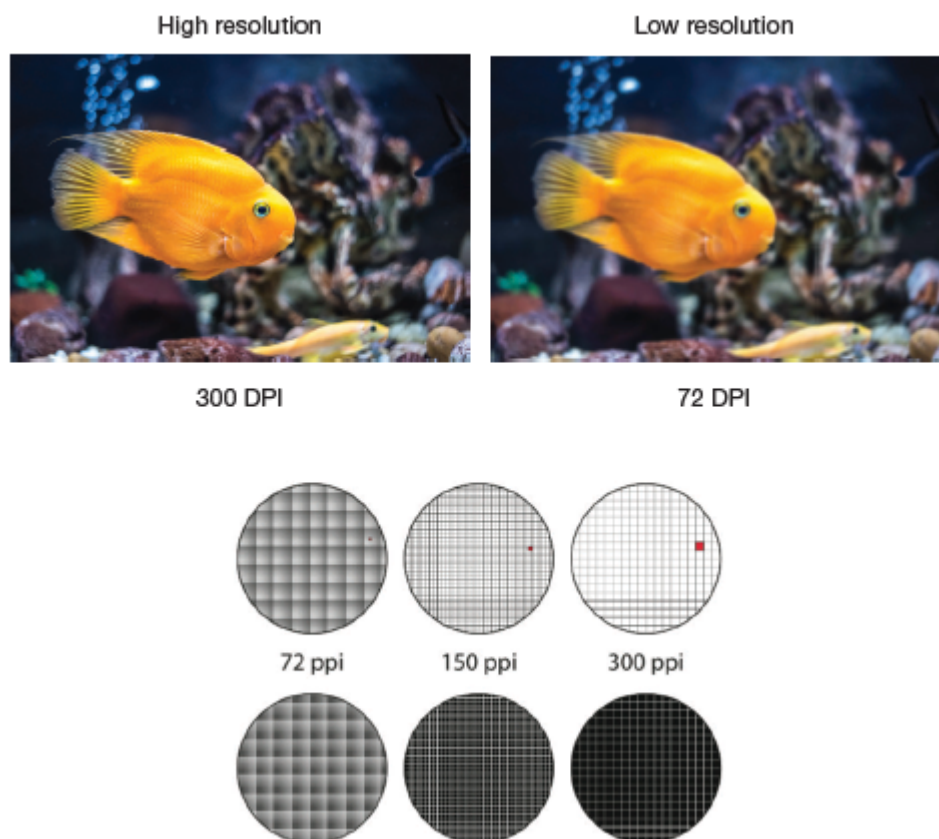
There are two conventions of specifying resolution. The first convention express resolution as the number of pixels in the width (a horizontal row) X number of pixels in height (a vertical column) of the screen. That is, if we say the resolution of the **computer monitor is 1280 X 1024**, there are 1280 pixels in a row from left-to-right and 1024 pixels from top to bottom.

The second convention expresses the number of pixels as a single number, like a **5 mega pixel camera** which means a camera having 5 million pixels (mega means millions). In our previous example, we had $1280 \times 1024 = 1,310,720$, or 1.31 megapixels.

Don't get confused between image size and image resolution. Image **size** is the dimension specified in length and width of an image. It is usually measured in pixels (px). But image **resolution**, or dpi, is the number of dots per square inch of an image when it is printed. While image **size** specifies how large the image will be viewed on monitors, **resolution** reflects printer quality.

6.9.2 DPI and PPI

DPI refers to the number of dots per inch on a printed image, whereas PPI refers to number of pixels per inch in a digital image. So, technically, it's PPI in a soft copy and DPI in a hardcopy of the image. A 300 PPI image will also be a 300 DPI image. Both the units refer to the number of units within a square inch of an image.



Credit: ZinaidaSopina / Shutterstock

FIGURE 6.11 Quality of an image depends on DPI

Figure 6.11 clarifies that we see a pixel in greater detail when we have a greater number of pixels in an inch.

6.10 Convolutional Neural Networks (CNN)

Convolutional neural networks are a special type of neural network that roughly imitates human vision. Initially, they were used to recognize handwritten digits and were especially used in the postal sectors to read zip codes, pin codes, etc. as shown in **Fig. 6.12**. A CNN requires a large amount of data and computing resources to train.

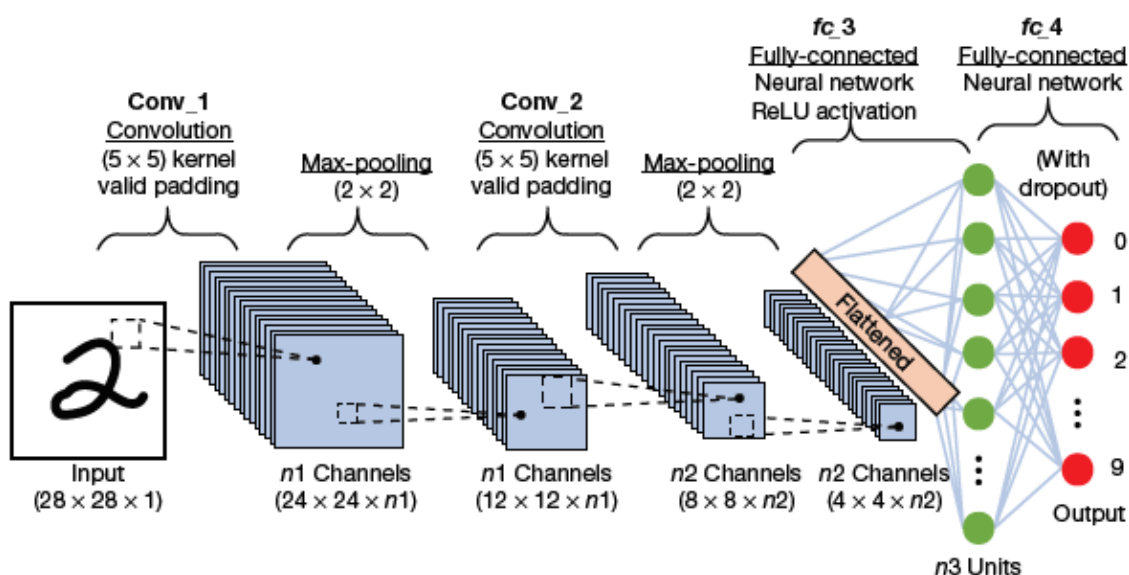
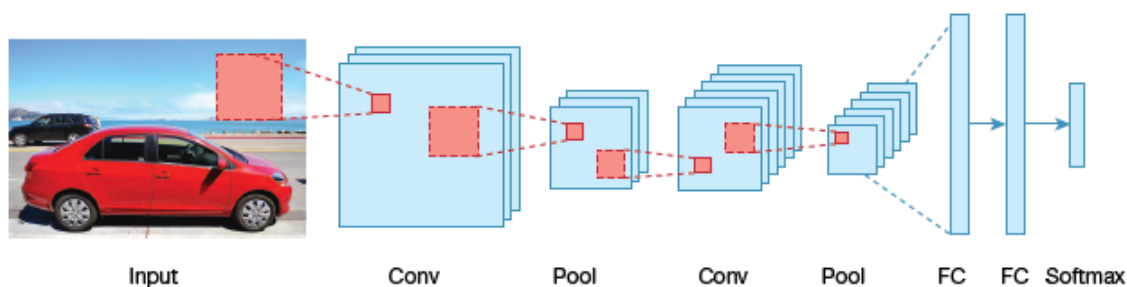


FIGURE 6.12 Classic computer vision

CNN is a branch of deep learning that uses multi-layered neural networks to analyse visual imagery. For this, CNN uses a special technique called convolution. Convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.



Credit: NERAMIT SISA / Shutterstock

FIGURE 6.13 Layers in CNN

We will not discuss the mathematics part required to understand how a CNN works. Just understand that a CNN is a composition of three main types of layers (refer **Fig. 6.13**).

Convolution layer, which is the primary building block and performs computational tasks based on convolution function.

Pooling layer, that is arranged next to convolution layer and is used to reduce the size of inputs by removing unnecessary information to speedup computations *without losing features that are critical for getting a good prediction*.

Fully connected layer, arranged next to series of convolution and pooling layer to classify input into various categories.

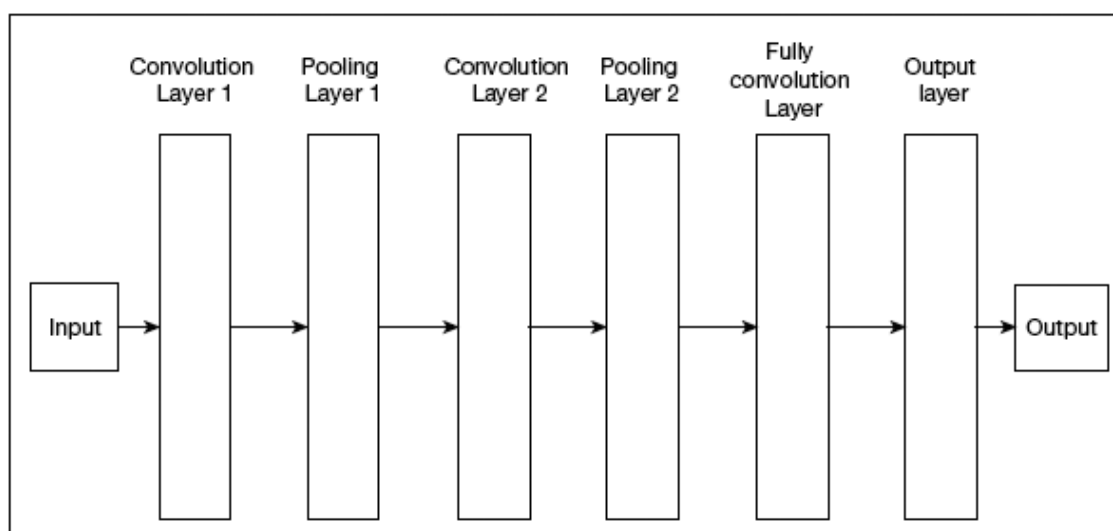


FIGURE 6.14 Convolution neural network

In **Fig. 6.14**, there are two series of convolution and pooling layer. The convolution layer receives and process the input to produce the output.

6.10.1 Understanding the Working of a Simple CNN

Convolution is a simple mathematical operation that can be applied on image processing operators. Convolution can be done by multiplying two arrays (or lists) of numbers, to produce a third array (or list) of numbers of the same dimensionality.

An image **convolution** is an **element-wise multiplication of image arrays and another array called the kernel followed by sum**.

For simplicity, let us consider a grayscale image that is represented as a pixel of values.

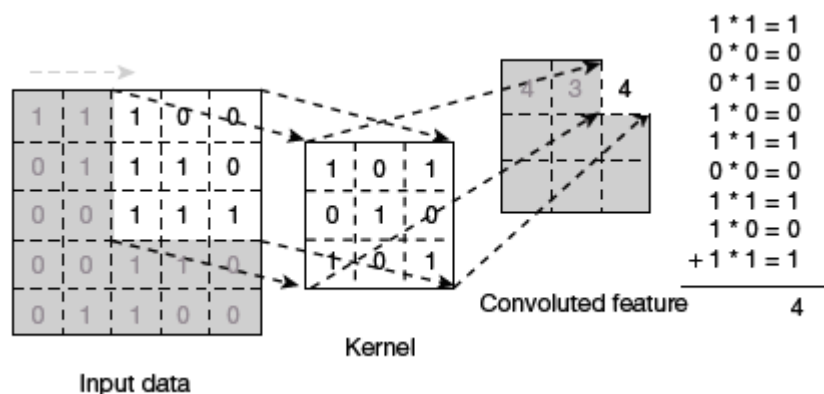


FIGURE 6.15 Kernel Layer for Filtering or Manipulating Images

In **Fig. 6.15**, we have taken a filter by using a kernel layer comprising of 3×3 matrix. **The kernel layer is applied to the input image to get the convolved feature.** This convolved feature is passed on to the next layer.

Thus, we see that a CNN consists of multiple layers of artificial neurons that calculates weighted sum of multiple inputs in order to output an activation value. When an image is input in a CNN, each layer generates several activation functions that are then passed on to the next layer. For example, the first layer extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. More layers in the network can be used to identify even more complex features such as objects, faces, etc.

Activation map (also known as feature map) generated by the final convolution layer guides the classification layer to produce a set of confidence scores (values between 0 and 1) indicating how likely the image is to belong to a particular class.

Therefore, a **CNN need not be limited to only one convolutional layer**. The first convolution layer captures low-level features such as edges,

color, gradient orientation, etc. Subsequent layers can extract other high-level features that gives a wholesome understanding of images in the dataset. In the convolution layer, **several kernels are used to produce several features.**

The activation or the feature map can then be used to

- reduce the image size so that it can be processed more efficiently.
- focus on certain features of the image that can help in processing the image further. For example, to recognize a person, we may need to focus on a person's eyes, nose and mouth to recognize the person rather than focusing on the entire face.

6.10.2 THE Pooling Layer

The convolution layer is followed by a pooling layer that can be of two types—average pooling and max pooling.

Max pooling finds the maximum value of a pixel from a part of the image covered by the kernel. It discards the noisy activations thereby performing de-noising along with dimensionality reduction.

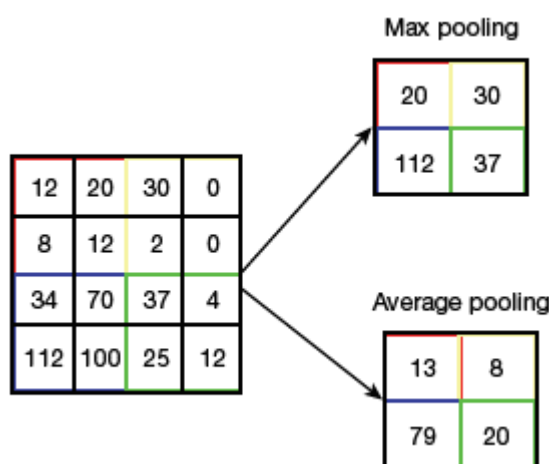


FIGURE 6.16 Pooling Techniques

Average Pooling returns the **average of all the values** from the part of the image covered by the Kernel. **Max Pooling performs better than Average Pooling.**

The two techniques of pooling can be better understood from **Fig. 6.16.**

6.10.3 Limitations

The advent of CNN has revolutionized the field of AI in a big way. A CNN gives in-depth results and recognizes even minute patterns and details that otherwise goes unnoticed to the human eye. However, even after using great amount of computing power while being trained with massive amount of data, CNN were not able to completely identify, block and remove inappropriate content from social media.

CNN also fails to detect objects when they are seen under different lighting conditions and presented from new angles. **Table 6.1** highlights the differences between a CNN and a neural network, often referred to as ANN or artificial neural network.

TABLE 6.1 Differences between CNN and ANN

CNN	ANN
• Partially connected layers • Can be used for image of any size • Considers only a part of the entire image for learning patterns (depending on filters) • Comparatively a smaller number of parameters used for learning • More efficient technique • More computational cost • Once a CNN has learnt to detect a pattern in one position, it can detect the same pattern even in different locations • More records/data for training is required • Success rate is high	• Fully connected layers • Can be used for small images • Considers the entire image for learning patterns • Large number of parameters used for learning • Less efficient technique • Less computational cost • Once an ANN has learnt to detect a pattern in one position, it cannot detect the same patter even in different locations (e.g., if the image is rotated or shifted) • Number of records/data for training depends on complexity of the task • Success rate is comparatively low.

Application Example of Image Features

In computer vision and image processing, a **feature** is a piece of information that is required to solve a computational task in context to a given application. These features may be points, edges or objects (refer **Fig.**

6.17). For example, if given six small patches of images, we can make our algorithm identify location of those image patches in the image. Once developed, we can use this algorithm to find similar images in a huge dataset of images.

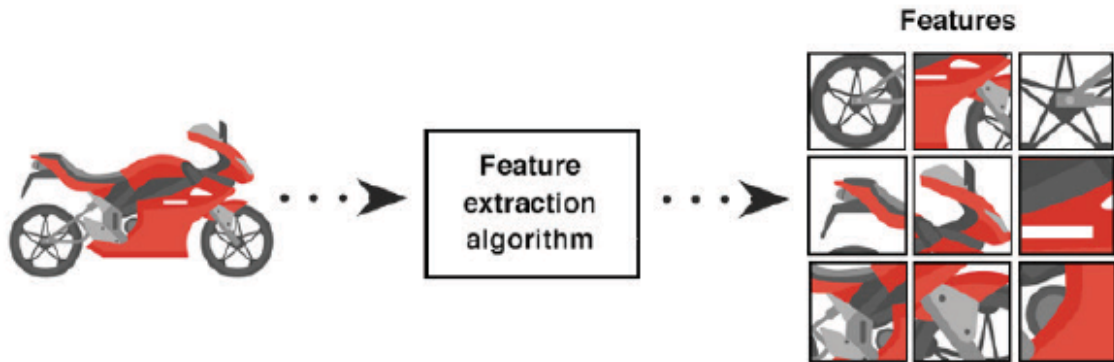


FIGURE 6.17 Extracting Features from an Image

6.11 Working with Images Using OpenCV

OpenCV is one of the most popular libraries providing functions for computer vision. The library has functions for the following tasks.

1. Reading an image
2. Extracting the RGB values of a pixel
3. Extracting the region of interest (ROI)
4. Resizing the image
5. Rotating the image
6. Drawing a rectangle
7. Displaying text

With OpenCV, users can easily do complex tasks such as identifying and recognizing faces and objects, tracking moving objects, stitching images together to generate an entire scene with a high-resolution image, etc.

To install OpenCV on windows using pip command, just write the following command in Window's Command Prompt.

```
Pip install opencv-python
```

Before using the OpenCV package, let us first discuss some important function that we will be using in this section.

The file should be in the working directory or we must give the full path to the image.

The **imread()** function accepts name of the image to open and a value 1 if the image has to be read as a coloured image. The syntax of this function can be given as follows:

```
cv2.imread(path, flag)
```

Here, the path is a string representing path of the image to be read and flag specifies how the image should be read—0 for grayscale image and 1 for a coloured image.

The imread() either returns the image loaded from the specified path file (if the image is found), else returns a matrix.

Now, type the following programming in Python IDLE to read and process images using OpenCV.

Example: Reading and Processing an image

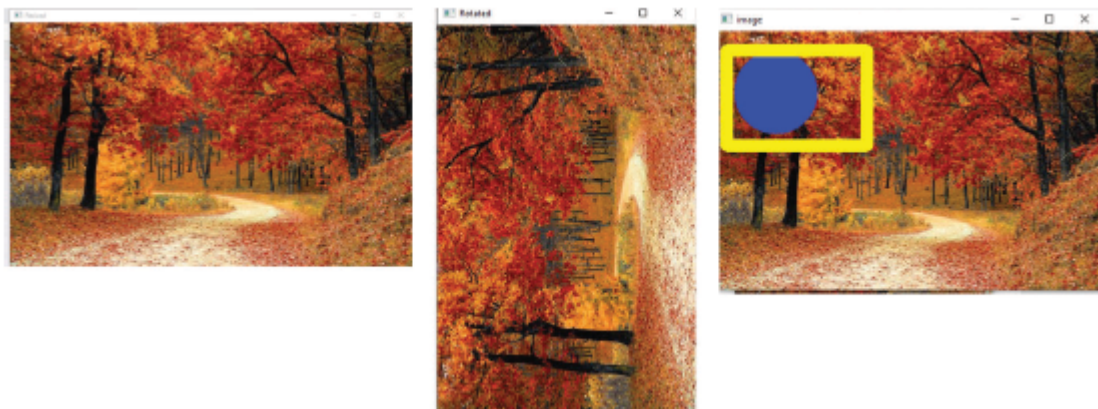

```

import cv2
import numpy as np
#importing the opencv module
img = cv2.imread('Nature.jpg',1)
print(img.shape)
# Resize image
img_resized=cv2.resize(img, (780, 540))
cv2.imshow("Resized",img_resized)
# Rotate image
image = cv2.rotate(img, cv2.ROTATE_90_COUNTERCLOCKWISE)
# Draw a circle on the image
cv2.imshow("Rotated",image)
cv2.circle(img,(80,80), 55, (255,0,0), -1)
cv2.imshow('image',img)
# Draw a rectangle on the image
cv2.rectangle(img, (15,25), (200,150), (0,255,255), 15)
cv2.imshow('image',img)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

OUTPUT

(2848, 4288, 3)



The **resize()** is used to change the size of the image. In ML algorithms, increase in number of pixels in an image increases the number of input nodes and thus the complexity of the model. The syntax of **resize()** is,

`cv2.resize(s, size)`

where,

s is the input image (required).

size is the desired size for the output image after resizing (required)

The **cv2.rotate()** method is used to rotate a 2D array in multiples of 90 degrees. The syntax of this method is,

```
cv2.rotate( src, rotateCode[, dst] )
```

where,

src is the image to be rotated.

rotateCode specifies how to rotate the array. Its values can be `cv2.ROTATE_90_CLOCKWISE`, `cv2.ROTATE_180`, `cv2.ROTATE_90_COUNTERCLOCKWISE`.

The **imwrite ()** function having syntax,

```
cv2.imwrite(filename, image)
```

is used to save an image on a storage device in the specified location. The function returns a value (or status) indicating success or failure in saving the image. The code given below reads the image in file `Nature.jpg` and saves it in `C:\-> Users -> Chikki Folder`

```
img = cv2.imread('Nature.jpg',1)
status = cv2.imwrite(r'C:\Users\Chikki\cat.png',img)
print("Image written success? : ", status)
```

If the file is successfully written then this function returns `True` otherwise returns `False`.

Basic Operation on Images

In this section, we will discuss some basic operations that can be performed on the images. These are, **Access pixel values and modify them**. **This can be done by writing**

```
img = cv2.imread('Nature.jpg',1)
px = img[50,60]          # accessing a pixel
print( px )

Output:
[157 166 200]
```

This prints a list containing 3 values that in turn specifies color of the pixel- Blue component, Green component and Red component. To modify the values, can access the pixel and then overwrite it with new value as shown below:

```
img[50,60] = [250,125,55]
print( img[50,60] )

Output:
[250 125 55]
```

Access Image properties like the size (total number of pixels in the image), number of rows, columns, and channels. While `shape()` is used to print the number of rows, columns and colors, `size()` prints the size of the image.

```
print( img.shape )
>>(350, 348, 3)
print( img.size )
>> 4248148
```

Image ROI (Region of interest) to select only a specific region of the image. For example, to detect eyes in an image, we need not search the entire image, we must concentrate only on face. Thus, here the face is our ROI.

Splitting and Merging Image Channels to split the channels from an image and then work on each channel separately or to merge the channels together.

OpenCV Drawing Functions

OpenCV has functions to allow users to draw certain shapes on an image such as circle, rectangle, ellipse, polylines, convex, etc. This is often used

to highlight any object in the input image. For example, Facebook identifies face by drawing a rectangle around it.

Drawing circle: circle() is used to draw a circle in an image. Syntax of circle() is:

```
cv2.circle(image, center_coordinates, radius, color, thickness)
```

where,

image is the input image on which a circle is to be drawn.

center_coordinates specify the X and Y coordinate values representing the center of the circle.

radius is the radius of the circle.

color specifies the color of the border line of the circle.

thickness indicates the width of the circle border line in px.

Drawing Rectangle: We use the rectangle() to draw a rectangle on the image. Syntax of rectangle() is,

```
cv2.rectangle(image, start_point, end_point, color, thickness)
```

where,

image is the input image on which rectangle is to be drawn.

start_point specifies the top left vertex X and Y coordinates of the rectangle.

end_point indicates the bottom right vertex X and Y coordinates of the rectangle.

color specifies the color of the border line of the rectangle to be drawn (in the form of BGR).

thickness gives the width of the rectangle border line in px.

Drawing Lines: The line() is used to draw lines on an image. Its syntax can be given as,

```
cv2.line(image, start_point, end_point, color, thickness)
```

where,

image is the input image on which line is to be drawn.

start_point specifies the starting coordinates (X and Y values) of the line.

end_point gives the ending coordinates (X and Y values) of the line.

Color indicates the color of the line to be drawn in the form of BGR)

thickness is the width of the line in px.

Write text on an image: Text can be written on an image by using the putText(). The syntax of this function can be given as,

```
cv2.putText(img, text, org, font,fontScale color)
```

where,

img represents the input image on which we have to write text

text is what we have to write on the image.

org denotes the Bottom-left corner of the text string on the image. Thus, org is used to set the location of text on the image

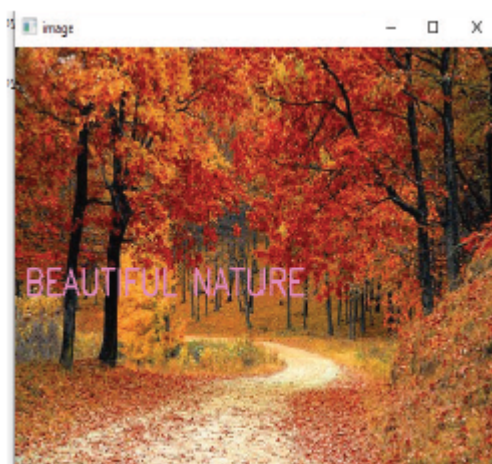
font specifies the font of text.

fontScale denotes the scale of the font by which size of the font can be increased or decreased.

color represents the color of the text in BGR form.

Example: Code given below places text on an image.

```
import cv2
import numpy as np
#importing the opencv module
font = cv2.FONT_HERSHEY_SIMPLEX
img = cv2.imread('Nature.jpg',1)
cv2.putText(img,'BEAUTIFUL NATURE',(10,200),font,1,(200,125,235),2)
#Display the image
cv2.imshow("image",img)
cv2.waitKey(0)
```



Canny Edge Detection: Edge detection is an image processing technique used to identify object's boundaries within an image. In the code given below, we have used the Canny Edge Detection algorithm. Syntax of Canny() can be given as,

```
edges = cv2.Canny(img, minVal, maxVal, apertureSize, L2gradient)
```

where,

img is the input image whose edges are to be detected.

minVal is the minimum intensity gradient

maxVal is the maximum intensity gradient

As per the syntax, edges with intensity gradient more than maxVal are definitely edges and those with intensity gradient less than minVal are sure to be non-edges and therefore discarded. Edges with values between maxVal and minVal are classified as edges or non-edges based on their

connectivity with the 'sure edges'. If the edges are connected to "sure-edges" then they are taken to be a part of the edges. Otherwise, they are treated as non-edges and discarded.

Example: Detecting Edges of an image.

```
#importing the opencv module
import cv2
# using imread('path') and 1 denotes read as color image
img = cv2.imread('Nature.jpg',1)
edges = cv2.Canny(img,200,300,True)
cv2.imshow("Edge Detected Image", edges)
cv2.imshow("Original Image", img)
cv2.waitKey(0) # waits until a key is pressed
cv2.destroyAllWindows() # destroys the window showing image
```

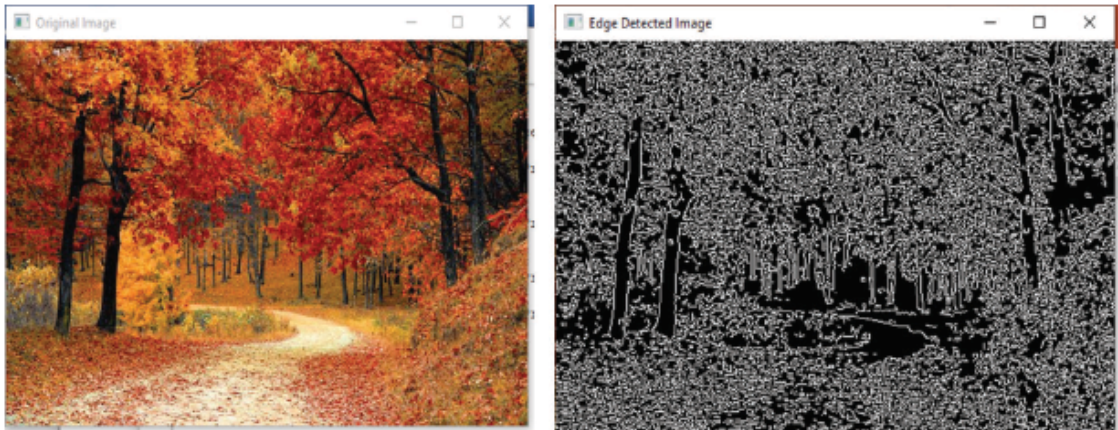


Image Smoothing is an image processing technique that removes noise from an image. For example, when blurring an image, low-intensity edges are removed. Blurring is often used to hide the details or any confidential information in an image. Different types of blurring techniques supported by OpenCV includes: averaging, median Blur, Gaussian Blur and Bilateral Filter.

In the averaging technique, the image is normalized with a box filter. Average of all the pixels in the box area is calculated. The average value then replaces the value of the pixel at the center of the box filter. The `cv2.blur()` in OpenCV is used to perform this operation. It's syntax can be given as,

```
cv2.blur(src, ksize)
```


where,

src specifies the image to be blurred.

Ksize represents the blurring box area.

```
#importing the opencv module
import cv2
# using imread('path') and 1 denotes read as color image
img = cv2.imread('Nature.jpg',1)
cv2.imshow("Original Image", img)
cv2.imshow('cv2.blur output', cv2.blur(img, (3,3)))
#cv2.imshow('cv2.GaussianBlur output', cv2.GaussianBlur(img, (5, 5),
cv2.BORDER_DEFAULT))
cv2.waitKey(0) # waits until a key is pressed
cv2.destroyAllWindows() # destroys the window showing image
```



median Blur technique calculates the median of all the pixels under the box area and replaces the value of the central pixel with this median value. `medianBlur()` provided in the OpenCV package is used to easily implement this kind of smoothing. Syntax of this function can be given as,

```
cv2.medianBlur(src, dst, ksize)
```

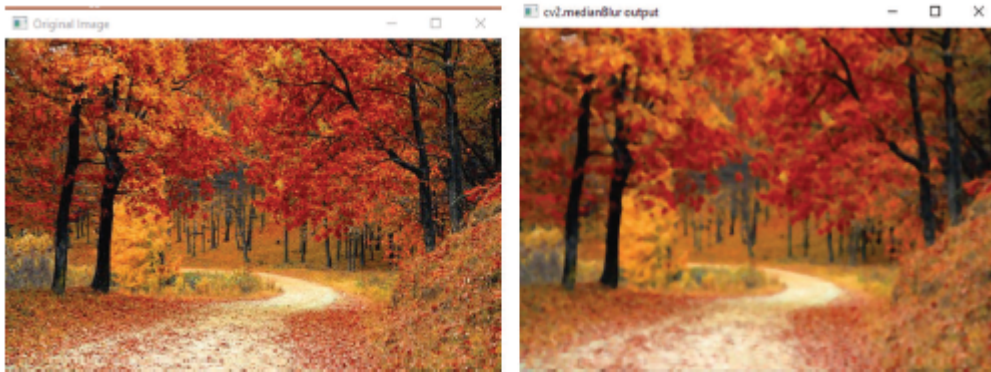
where,

src represents the input image.

dst represents the output image.

ksize represents the size of the kernel (box).

```
import cv2
# using imread('path') and 1 denotes read as color image
img = cv2.imread('Nature.jpg',1)
cv2.imshow("Original Image", img)
cv2.imshow('cv2.medianBlur output', cv2.medianBlur(img,5))
cv2.waitKey(0) # waits until a key is pressed
cv2.destroyAllWindows() # destroys the window showing image
```



Gaussian Blur technique, instead of the box filter, a Gaussian function is used to blur the image. The width and height of the kernel window should be specified and these values should be positive and odd. Standard deviation in X and Y coordinates are also specified using the `sigmaX` and `sigmaY` parameters respectively. If `sigmaX = sigmaY = 0`, then they are calculated from the kernel size. The `GaussianBlur()` is used to implement this technique. Its syntax is,

$$\text{GaussianBlur}(\text{src}, \text{dst}, \text{ksize}, \text{sigmaX}, \text{sigmaY})$$

Where,

src is the image to be blurred

dst is the output image of the same size and type as **src**.

ksize specifies the size of the kernel.

sigmaX and **sigmaY** are double values that represents standard deviation in X direction and Y direction respectively.


```
import cv2
# using imread('path') and 1 denotes read as color image
img = cv2.imread('Nature.jpg',1)
cv2.imshow("Original Image", img)
cv2.imshow('cv2.GaussianBlur output', cv2.GaussianBlur(img, (5, 5),
cv2.BORDER_DEFAULT))
cv2.waitKey(0) # waits until a key is pressed
cv2.destroyAllWindows() # destroys the window showing image
gd
```



Bilateral Filter is a highly effective technique to remove noise but is slower compared to other filters. The problem with Gaussian filter is that it blurred the edges also which was not at all desirable. Bilateral filter blurs only the pixels with similar intensities to the central pixel thereby preserving the edges (pixels on edges have large intensity variation). It is implemented using the `cv.bilateralFilter()`.

6.12 Immersive Experience

Immersive experience is any technology that extends reality or creates a new reality by leveraging the 360 space. While some types of immersive experience extend reality by overlaying digital images on a user's environment, others create a new reality by completely immersing users in a digital environment, shutting them out from the rest of the world. This gives as more engaging or satisfying experience.

The Skyview app is an example of AR in action, allowing you to see where constellations are in the sky in real-time as you move your phone around.

The simplest tool to enjoy immersive content is through smartphones or specialized headsets. On the software side, although a web app or a desktop application can be used, web app is more popular.

To better understand the concept of virtual reality, just imagine how exciting it would be if you send your digital avatar to attend a meeting on your behalf. The avatar not only attends but also jots down the minutes, and reports to you after the session is over.

6.12.1 Elements of Immersion

An immersive experience is defined as a multi-sensory experience (sight, sound, touch and scent) across a journey, or a task that is contextually relevant to create intuitive and emotional value for the user that drives a better relationship.

Sight VR headset blocks out the peripheral vision or use a wrap-around headset to enhance vision so that the attention of the wearer is focused on what is happening directly in front of him/her. On the contrary, AR uses headsets or smartphone displays to add virtual elements to the real world.

Sound VR headsets include sound-dampening headphones that force the wearer to focus on the sounds of the virtual world. AR on the other hand, provides sounds for whatever is taking place on the screen.

Touch VR accessories provide haptic feedback for the wearer. Moreover, effects like vibrations and rumbles are used to synchronize with the virtual world. AR, however, does not need touch to increase immersion.

Thus, all these elements together create an immersive experience.

True immersion depends on a suspension of disbelief and a willingness to be transported to a different world.

6.12.2 Types of Immersive Experiences

360-degree content is an interactive photo or a video recording shot in every direction at the same time. This allows users to change the viewing direction at any moment. Digital twin is a precise virtual model of a real-life object, process or system. We can even create a digital of an object that does not even exist. Such models can be used to display textual details or to monitor, control and run simulations.



Credit: insta_photos / Shutterstock

FIGURE 6.23 Immersive Experiences

Technically, 360° content integrates two or more videos in a dome-like space. These videos are shot from all possible angles, with 2D instructional content overlaid on top of them. This technology is extensively being used to walk customers through a detailed range of product features and benefits. This helps users to get a wide-eyed version of the product in the real world. Similarly, 360° content is being used in travel and tourism industry to help tourists pre-visit a particular destination.

Virtual reality (VR) is a computer simulation that immerses the user inside a virtual world and allows him to interact with the environment. VR stimulates user's senses to make him feel as if he was actually there in that environment.

PwC predicts that VR and AR could add \$1.8 trillion to the global economy by 2030.

VR shuts a user completely from the rest of the world while being surrounded by content. Through a head mounted display (HMD), whatever content the user experiences in the headset becomes the reality for him.

360 vs 360 VR: Though the two terms are same, 360 VR is a combination of 360 content with VR mode (refer **Fig. 6.23**). It allows users to view the 360 contents in a cardboard headset, or sometimes even a mobile 360 VR headset. Users can experience 360 contents without the help of an HMD and is best experienced on mobile.

Mobile VR leverages HMDs that are connected to smartphones. Examples of these are the Samsung Gear VR, Google's Daydream or even Google Cardboard.

True VR uses headsets with powerful computers or consoles. Examples include HTC Vive, Oculus Rift, Playstation VR, etc. These powerful HMDs use sensors (separate from the HMD) to keep a track of a person's movement and his surroundings. The sensors would then adjust the VR content to whatever is within a user's environment. True VR uses CGI and 3D modelling for content.

FIGURE 6.24 AR vs VR

360 VR vs In-VR: With 360 VR, users can use their device to explore content by looking in any direction but not change the content that surrounds the objects. In-VR content takes depth into consideration. This means that when a person is viewing an object, he can move closer to the

object or move further away from it. The entire content is adjusted accordingly.

Augmented reality (AR) enhances the user's perception of the real world by adding a computer-simulated layer of information on top of it (refer **Fig. 6.24**). While AR adds or hides data from the environment, VR completely replaces the user's perception of the real world.

Using AR, digital images are presented on top of the real world. This means that users who leverage AR are not completely shut off from the world. Instead, AR extends their reality.

For example, Snapchat Filters that overlay digital images onto your face is an application of AR. Another example of AR is Pokemon Go where users can walk around with their mobile phones and find Pokemons that are overlaid on the environment around the user.

Many stores like Target and Ikea have their own AR apps where users can choose a product and place it virtually on anything to see how it looks. For example, you can choose a dress and see how it looks on you. This allows consumers to 'try out' a product before even buying it.

AR also provides the ability to 'touch' the product. Some examples are as follows:

1. BMW gives its users a chance to try out 3D models of their cars.
2. Burger King used the motto 'Burn That Ad' in its advertising campaign in Brazil. In this campaign, the user is asked to scan any competitor's ad and 'burn' it by using augmented reality. The most active users are then rewarded with a free branded whopper.

TABLE 6.2 Differences between augmented reality and virtual reality

Mixed Reality (MR) brings together VR and AR into an enhanced version of AR. Mixed reality is also referred to as AR 2.0 because it achieves better immersion than AR. MR integrates virtual objects in the real world. It responds to changes in the environment and to user interaction in real-time.

Unlike VR, MR does not shut out users from the rest of the world. Instead, the Head Mounted Display (HMD) is more like a pair of glasses that overlays digital images on top of your environment (just like AR). Mixed reality is extensively being used for commercial, developmental and entertainment purposes, for the following:

1. Conducting simulation-based training especially military training without increased risks.
2. Creating an interactive environment with the full inclusion of virtual objects in reality.

For example, Disney World uses a combination of AR/MR to create a game to entertain park visitors while they walk around and wait in line for physical attractions.

Extended reality (ER) is used to enhance or completely replace reality. It is an umbrella term for every immersive technology including augmented reality, virtual reality, mixed reality, digital twins, etc. This technology extends reality by adding digital counterparts in the form of inscriptions, videos or animations.

ER is usually used for training employees, giving a walk through a live hologram of a product to understand its features.

Digital Twins: They are near-exact virtual models of real-life objects, processes or systems. This technology is usually used in manufacturing or engineering to simulate physical things either for optimizing or for studying how they behave before actually building them.

Accenture uses full-scale XR to create highly personalized experiences for its clients through sight, touch and sound.

For example, NASA uses digital twins to monitor and optimize satellites in space from the ground. Digital twins of satellites help scientists to foresee potential dangers, fuel shortage, engine power, steering power and other parameters of any spacecraft mission.

Mercedes uses this technology to optimize the performance of its F1 cars. Digital twin is also extensively used in healthcare and for digitally recreating entire cities. Therefore, a **digital twin** represents a 'carbon-copy' of a real-life component, structure or process which can be easily tracked in real-time. Digital twin technology combined with 3D-modelling techniques helps users to actively interact with the environment.

Advantages

1. Digital twins are used for quality assurance tasks.
2. It provides quick insights into a product anytime, anywhere for faster decision making.
3. It helps to visualize defects in the real-time performance of certain equipment.

4. It troubleshoots machines at remote locations.

6.12.3 Applications of Immersive Experiences

Immersive technology is not a new term for those who read Sci-Fi books or watch in Sci-Fi movies.

Business: Every business generates a number of reports using ERP software. These reports are, at times, difficult to understand. In such a case, an AR headset is used which will see only the most important bits of data in the report. The headset will show all the details when the user looks at specific parts of the report.

Using a VR headset can make remote business meetings through tele or video conferencing more authentic.

E-commerce uses 360-degree technologies to provide better visualization. It allows users to look at the product from every possible angle. Many websites have made the 360-degree content even more interactive. It can react to changes in options in real time, giving the user an immediate preview.

These websites are even using MR to show how the product will look like in a real-life environment.

Architecture and design are indeed extensively using immersive technology. With VR, we can explore building designs in a 1:1 scale. AR is used to show additional relevant information. MR allows users to edit 3D models created by them as per user's requirements. Any small changes and adjustments in the simulation are reflected instantly.

Engineering, maintenance and repair of complex equipment are easily done using immersive techniques. While repairing, engineers use AR headsets to get useful data right away. Artificial intelligence can help them by giving suggestions about the next best action that can be taken.

In the design phase, a digital twin is used by engineers to see their creations in a virtual environment. They no longer need to build prototypes.

Similarly, during testing phase, a digital twin can be used for running simulations and collecting analytical data.

Credit: Stanislav Vladimir / Shutterstock

FIGURE 6.25 AR for Shopping

Healthcare is another area that extensively uses immersive experiences. Augmented and virtual reality both can make the current state of medical training more effective. While AR provides useful information to health-care employees when learning how to use new equipment, VR can help them to practice more complex medical procedures in a virtual environment before actually performing them on a live patient.

Similarly, in radiology, AR is used during pathological tests to study medical procedures and evaluations. By virtually mapping the X-ray data of a patient, radiologists can identify abnormalities of bones and classify the disease more quickly than before.

Telemedicine uses VR or MR to break the distance barriers between patients and practitioners. Patients can now get themselves diagnosed in a more comfortable setting and having more accessible medical services. An AR headset can be used to give important information to every patient in a hospital, keeping their mind busy and divert it from all worries.

Even VR solutions are used to reduce anxiety in patients, helping them with sleep or just serve as a positive distraction. AR can be used to visualize patient data and treatment assistance. An AR headset can show only

the relevant information directly on the patient's body or give suggestions regarding the medicine to be prescribed. Moreover, during surgeries, immersive technologies can be used to outline the most optimal cutting path, give warnings about potential problems and help with additional information.

The travel industry is using immersive technologies for selling real-life experiences. For example, 360-degree content and VR provide customers with the option to try before they buy something. Customers can have a virtual tour of the place whether they intend to go, virtually walk around the hotel they intend to book. This will ensure them that they are making the right choice. AR headset can serve as a 24 × 7 personal tour guide that provides additional information in real time, making sightseeing easier and more informative, finding cafes, restaurants, hotels, inns and landmarks.

Real estate uses immersive technologies to virtually show the property that has to be taken on rent or for sale. It is very inconvenient and time consuming for the buyer as well as the seller to see every property. Buyers can check out properties remotely whenever they have time; that too without the help of a real estate agent. Later, they can personally visit only selected properties and use an MR headset to get information about the property, decoration options or furnishing suggestions.

DHL could cut 25% cost by using AR in their logistic business. Automaker Volvo uses VR headsets in the testing process and BMW uses it for work-station planning.

Augmented reality in logistic centres uses immersive technology to reduce warehouse costs by using AR smart glasses in the picking process. The smart glasses show a list of stuff to be picked (reducing lookup times) and calculate an optimal route (to lower travel time). Headset are used to read barcodes and determines if the current location is the right one. In case the location is not correct, the headset guides the workers to the correct location.

Automakers are using MR for testing automobiles in simulated extreme situations. Sensors in headsets enable precise measurement of reaction

times and other important safety factors. VR is also used in workstation planning to reduce the labour force required and assess processes completely virtually. AR headsets can be used by employees for engine assembly units training at a faster pace (approximately three times faster).

While teaching about planet Mars, teachers can use MR to give a virtual tour to mars to explore its surface.

The car industry is also using immersive technology to create virtual showrooms where consumers can look at cars, take virtual test drives and even virtually sit in actual car seats.

At **McLaren**, VR software is used to design car models. A live hologram of the car is made which designers can alter. Even Volvo engineers use VR software to simulate drive trials and customize designs accordingly.

Companies like Adidas use VR to enhance workflow by using VR as a virtual team collaboration space. They use it when different teams have to work together. VR gives employees the ability to be in the same virtual room and share 3D models and other content.

Walmart is using VR headsets for training employees. They use immersive experience to prepare their employees for different real-life situations.

Education: VR can help students to learn about a particular concept. For example, a VR headset can be used to take a virtual tour of a historical monument and learn about historical events of importance associated with it. Similarly, MR can be used to learn about human body and how each part functions by overlaying digital images on an actual person and highlighting where the different parts are located.

Gaming is among the first applications that extensively used immersive technology. The launch of Playstation VR, Pokemon Go and VR arcades are examples of application of this technology. **Table 6.3** list some games and the underlying technology they use.

TABLE 6.3 Games using Immersive Experiences

Immersive storytelling and virtual try-on: AR places users in the centre of the story, captures their attention or showcases products right on their faces. For example, cosmetic brands like L’Oreal and Maybelline allow consumers to apply virtual makeup to boost sales among Gen Z. Similarly, Turkish Airlines has created a branded filter on Instagram called ‘Turkish Airlines My Dream Destination Project’ that immerses people into virtual travels during the pandemic.

Designers combine traditional 2D CAD applications with VR tools to perfect their designs. Examples include softwares like V-Ray or SketchUp.

Aviation Management: Immersive technology allows flight engineers and ground handlers to identify, probe and analyse the situation before the aircraft takes off. This helps the crew to perform aircraft maintenance and engineering checks from a remote location. AR/VR headsets put teams on airside so that they can interact with aircraft holograms to conduct visual inspections or cargo inspections.

Supply chain and logistics: VR reduces logistic costs by providing virtual-navigation routes for the pickup of goods. This not only increases efficiency of the employees but also expedites the logistics process (refer **Fig. 6.26**).

MR devices support professionals working in supply chain by superimposing routes over truck windshields. Similarly, AR devices help by notifying teams when supplies perish.

Credit: Zapp2Photo / Shutterstock

FIGURE 6.26 Using Immersive Experiences for Handling Logistics

Entertainment sector: These days, especially after the outbreak of COVID-19, many events are conducted virtually. Immersive technology compensated for the lack of physical presence. Immersive devices help users to experience a concert or movie as if he/she are a part of it (refer **Fig. 6.27**).

Commercial businesses using immersive experience right now include the following:

DHL logistic centre uses integrated AR smart glasses in the packing process to reduce lookup time, optimize navigational routes and guide the worker to the correct location.

Credit: Roman Zaiets / Shutterstock

FIGURE 6.27 Immersive Experiences for Entertainment

BMW uses AR headsets for workstation planning to enhance productivity, assess processes and foster training of engine units.

Adidas uses AR for imparting virtual employee training. 360° content is used to prepare employees for real-life situations (e.g., a Black Friday sale simulation).

Autodesk Revit Live and Iris VR provide immersive architectural and building visualizations to help clients visualize buildings even before starting with the construction phase.

6.12.4 Virtual Reality Systems

Virtual reality is a complete artificial world that does not exist at all. Users ‘immerse themselves’ into this non-existent environment not only as an observer but also as a participant. VR devices are used to move around and manipulate objects virtually. Some common characteristics of VR systems are as follows:

Real-time simulation. The VR system provides picture sound, and other sensations (if any) in response to the actions performed, instantly, with-

out noticeable delays.

Realistic imitation of the user's environment. To give an illusion of real-world to fully immerse the user in the world of virtual reality, the system should display virtual objects with realistic effects.

Multi-user mode. VR systems facilitates collaborative work by multiple users simultaneously.

Interactivity. The virtual world must work based on user's interaction with virtual objects. Here, the user is not just an extremely active observer but can also move, manipulate objects, look in different directions and perform other actions.

VR can be experienced through mobile devices. In 2014, Google invented that allows an immersive experience through a smartphone screen.

Some companies that have successfully used VR are as follows:

1. The Teleporter 4D launched by the Marriott hotel chain in 2014 provides virtual reality booths that take the user on an introductory tour of one of the cities. The booth also simulated the climatic conditions of the selected city.
2. Lexus facilitates users to take a virtual test drive of one of its latest models. The user just needs a smartphone and Google Cardboard.
3. Samsung introduced the 'Be Fearless' programme in 2016 to help users overcome their fears. Participants took a course using Samsung Gear VR to check if they are able to deal with stressful situations in virtual reality so that if they land up in a similar situation in real life, they can handle them in a better way without any phobia.
4. McDonald's introduced a new packaging for Happy Meal, which can be transformed into virtual reality glasses.

Advantages of VR

1. VR provides diverse types of instant data.

2. It presents images from multiple viewpoints.
3. VR demonstrates non-visible data to the user (like in geochemistry).
4. Facilitates users to 'visit' the places normally inaccessible to individuals.
5. VR provides an experience which can be recreated anytime.

Limitations of Virtual Reality

1. It only creates an imaginary world that has no connection with the real world.
2. VR cannot deal with real-world objects.
3. Though it can be used for educational purposes, it is seldom used as it devalues the importance of human connections and synergy in teaching-learning process.
4. VR systems are rigid and lacks flexibility

6.12.5 Types of Augmented Reality

Augmented reality is divided into four different categories as follows:

Marker based augmented reality uses interfaces like camera or some type of visual marker (like QR code) to produce result when the marker is sensed by a reader.

Marker-less augmented reality uses a GPS, digital compass, velocity/accelerometer which is embedded in devices to provide data based on user's location.

Projection-based augmented reality projects artificial light on real world objects that senses human touch to exhibit a miracle.

Superimposition-based augmented reality partially or fully replaces the original view of an object with a newly augmented view of that same object.

Advantages of Augmented Reality

1. Increase sales: Apps such as lenskart are AR-based apps which allow users to check if the product looks good on him/her or not. This in-

- creases customer interaction, satisfaction and ultimately boosts sales.
2. Enriches content: Unlike VR, AR facilitates interaction with the real world, which enriches experience and content.
 3. For these reasons, AR is extensively being used for training and skill development applications in various industries (like military, pilot training, nuclear plant training, etc.).

Limitations of Augmented Reality

1. Sometimes, implementation of AR makes the application too ambiguous to use. When there are more layers, it becomes difficult to match with the exact environment and, therefore loses its user-friendly charm.
2. Using too much AR especially where it was not required, may obscure or narrow a user's interests or tastes.
3. Privacy control is a matter of concern with AR.

Therefore, some applications prefer to use MR rather than pure AR or VR.

6.12.6 Mixed Reality

Mixed reality is the combination of the best features of VR and AR. It ensures that limitations of both AR and VR are addressed to overcome them. MR integrates VR content with real time linear video and allows user interaction with virtual or real objects (refer **Fig. 6.28**).

Credit: Gorodenkoff / Shutterstock

FIGURE 6.28 Interacting with Objects using MR Techniques

Image Source: Microsoft

Thus, MR is a kind of hybrid system that includes both physical and virtual elements. It is a right blend between a fully physical environment with no virtual elements, and a completely virtual environment. To use MR, users need to wear a headset that helps them see through a translucent viewport or glass on which visuals are projected.

AR visualization can create 3D catalogues and interactive demos of a product to make it visually appealing to customers.

MR tends to develop virtual characters for the real world to create a deeper sense of immersion for the user than AR alone does. Since MR facilitates interaction between virtual world and real-world objects in real-time to give a parallax effect, it is also referred to as augmented Reality 2.0 or hybrid reality. People can live, work, trek, shop and survive more realistically in an MR world than in VR.

Advantages of Mixed Reality

1. MR presents a highly interactive aspect.
2. MR renders realistic projection of virtual objects on the surroundings.

3. It allows users to interact with the immersive content using their body (like, finger gestures). It is not necessary to interact only with remote controllers or phone screens.

Disadvantages of Mixed Reality

1. The cost of MR devices is high.
2. There are issues related to large file sizes and low-resolution content.
3. Personal life may get affected due to addiction
4. Business users need a sound technical team to address issues in a timely manner.

Some Exclusive Applications of Mixed Reality

1. MR has the potential to take interactive product content management (IPCM) to the next higher level.
2. MR facilitates collaborative work, especially in a global team having members working from remote locations.
3. MR supports the manufacturing sector by helping it to create mock-ups of processes.
4. In the health sector, MR has made surgeries simpler and more flexible than before. Doctors use smart glasses to concentrate on details.
5. In education, MR can be used to switch from e-learning to simulation-based learning, thereby transforming modern education system.

6.12.7 Challenges of Immersive Experiences

Though an immersive experience seems to provide an ocean of exciting opportunities, there are certain technological shortcomings, primary health and safety factors associated with it that prevent its use at large. Some other challenges that act as a roadblock in the adoption of immersive technologies include the following:

1. Immersive devices pose a big threat to security. Since applications are monitored automatically, third-party users can hack the server of a particular company to steal crucial information (about data, finances, etc.) about a company.

2. The technology can be misused to perform crimes like cyber bullying, harassment, cyber stalking and leaking data of public figures.
3. An immersive experience impacts cognitive concentration that comes naturally to humans. It can fully replace a user's sensory connection to the real world.
4. Immersive experiences work best with a 5G Internet connection which is still not available everywhere.
5. The technology comes with a high cost.
6. Not every user has the digital knowledge to operate immersive devices, making immersive experience viable only for a limited set of users.



PROGRAMMER'S ZONE

Python program to demonstrate erosion and dilation of images.

Credit: Screenshot used with permission from Microsoft

HANDLING MOUSE EVENTS

Generating image pyramid

DRAWING RECTANGLE

DRAWING CONTOURS



KEY TERMS

Computer vision: A field of artificial intelligence that helps computers gain a high level of understanding from digital images or videos.

Convolutional Neural Network (CNN): A branch of deep learning that uses multi-layered neural networks to analyse visual imagery. For this, CNN uses a special technique called convolution.

Convolution: A mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

Object Classification: The process of identifying the broad category of object in the photograph.

Object Identification: The process of classifying the type of a given object is in the photograph.

Object Verification: The process of confirming the presence of an object in the photograph.

Object Localization: The process of locating objects in the photograph. The goal of **localization** is to find the location of a single object in an image.

Object Detection: The process of repeatedly locating and classifying all objects of interest in the image.

Object Segmentation: The process of enumerating the pixels that belong to an object in the image.

Object tracking: The process of detecting an object and then discern its movements.

Video motion analysis: The process of estimating the velocity of objects in a video, or the camera itself.

Scene reconstruction: The process of creating a 3D model of a scene inputted through images or video.

Image restoration: The process of using machine learning based filters to remove noise (to remove parts of image that are blurred).

Visual Search Engines: A search engine that retrieves images meeting the specified criteria.

Noisy image: An image with incomplete/incorrect data makes it hard for computer vision to recognize objects.

Adversarial Attacks: Attacks in which an attacker intentionally attacks a machine learning model to make the machine fail. These flawed models are difficult to identify and may affect the system adversely.

Immersive Experience: A technology that extends reality or creates a new reality by leveraging the 360-degree space.

360-degree content: An interactive photo or a video recording shot in every direction at the same time. This allows users to change the viewing direction at any moment.

Virtual reality (VR): A computer simulation that immerses the user inside a virtual world and allows him to interact with the environment.

Augmented reality (AR): Technology that enhances the user's perception of the real world by adding a computer-simulated layer of information on top of it.

Digital Twins: Near-exact virtual models of real-life objects, processes or systems that is usually used in manufacturing or engineering to simulate physical things either for optimizing or for studying how they behave before actually building them.



RECAPITULATION POINTS

- A CNN complements the machine learning or deep learning model by breaking images down into pixels that are given tags or labels.
- The black-and-white image stores a '0' for a black pixel and a '1' for white pixel. This is known as a 'bitmap' image as it only requires one bit of information, 0 or 1, to describe the color

- Computer vision algorithms can be used for predictive maintenance. Images obtained through cameras (probably fitted in the eyes of robots) can be analysed to detect potential problems they occur. CV can even be helpful to reduce defects.
- CNNs are a special type of neural network that roughly imitate human vision.
- Edge detection is an image processing technique used to identify object's boundaries within an image.



REVIEW QUESTIONS

1. Discuss the applications and challenges of computer vision.
2. CNN is a composition of three main types of layers. Elaborate this statement.
3. What is the role of activation function in a neural network and in a CNN?
4. Give the different elements of immersion.
5. Write a short note on mixed reality.
6. What do you understand by the term 'computer vision'?
7. Explain the tasks involved in computer vision.
8. Computers can detect objects better than humans. Justify this statement.
9. How does computer vision work with deep learning?
10. How can we use computer vision in manufacturing units?
11. Give any four areas where CV is extensively used.
12. List any three challenges faced by computer vision systems.
13. What do you understand by CNN? Explain the different layers that makes up a CNN model.
14. Differentiate between CNN and ANN.
15. Differentiate between bitmap and a greyscale image.
16. What is image resolution? Discuss two ways in which resolution can be specified.
17. Give any two benefits of using an activation map in a CNN.
18. Why do we need multiple convolution layers?
19. Differentiate between
 1. Image size and image resolution

2. CNN and ANN

3. Augmented and virtual reality



1. _____ systems like Google Lens identify images by identifying image properties like composition, colour, texture, etc.
2. The word _____ means 'picture element' indicating one single dot.
3. _____ allows users to change the viewing direction at any moment.
4. Near-exact virtual models of real-life objects, processes or systems are known as _____.
5. Image _____ is an image processing technique that removes noise from an image.
6. Auto companies usually use _____ to perform a dummy crash test in a metaverse world.
7. A _____ neural network is used in computer vision algorithms.
8. _____ is the process of classifying the type of a given object in the photograph.
9. Object _____ is performed to detect an object and then discern its movements.
10. _____ systems recognize faces in images and videos.
11. Pokemon Go and Snapchat use _____ reality applications.
12. A _____ search engine retrieves images that meet the specified criteria.
13. _____ attacks are intentionally done on a machine learning model to make the machine fail.
14. _____ neural networks are a special type of neural network that roughly imitate human vision.
15. _____ is one of the most popular libraries providing functions for computer vision.
16. Tiny dots on a monitor are known as _____.
17. _____ refers to the detail that an image holds.

**STATE TRUE OR FALSE**

1. Image restoration uses machine learning-based filters to remove noise.
2. A 300 PPI image will also be a 300 DPI image.
3. CNN has only one convolutional layer.
4. Average pooling performs better than max pooling.
5. Computer vision makes extensive use of deep learning techniques.
6. To remove noise from an image, we use scene reconstruction technique.
7. Computer vision algorithms can detect diseases by analysing X-rays, MRI and mammography.
8. Google Lens is used to obtain detailed information from images.
9. A fully connected layer is arranged next to series of convolution and pooling layer.
10. The pooling layer is followed by a convolution layer.
11. Average pooling performs better than average pooling.
12. An image is rotated in multiples of 90 degrees.
13. Fully connected layers are present in a CNN.
14. Pooling layer is present in a CNN but not in ANN.
15. Pixel means picture element.
16. In a bitmap, every pixel stores the value of RGB components.
17. A bitmap cannot represent shades of grey.
18. Fully connected layers are present in a CNN.
19. Pooling Layer is present in a CNN but not in ANN.
20. Pixel means picture element.
21. In a bitmap, every pixel stores the value of RGB components.
22. A bitmap cannot represent shades of grey.
23. Higher the resolution of image, lesser is its clarity.
24. A 1280 X 1024 pixels camera has a higher resolution than a 3 mega pixel camera.



MULTIPLE CHOICE QUESTIONS

1. Object _____ finds and classifies a *variable number of objects* in the given image.
 1. Verification
 2. Detection
 3. Localization
 4. Classification
2. A greyscale image can represent a total of _____ different colours.
 1. 1
 2. 2
 3. 256
 4. 16 million
3. Image _____ is often measured in Dots Per Inch (DPI) and Pixels Per Inch (PPI).
 1. Hue
 2. Resolution
 3. Intensity
 4. Size
4. Which layer of convolution neural network reduces the size of inputs to speedup computations?
 1. Convolution
 2. Pooling
 3. Fully-connected
 4. All of these
5. _____ enhances the user's perception of the real world by adding a computer-simulated layer of information on top of it.
 1. Augmented reality
 2. Virtual Reality
 3. 360 -degree content
 4. Digital Twin
6. _____ reality integrates virtual objects in the real world.
 1. Augmented
 2. Virtual
 3. Mixed
 4. All of these

7. _____ is an image smoothening technique.
1. Averaging
 2. Gaussian Blur
 3. Bilateral Filter
 4. All of these
8. The size of an image is represented as $600 \times 300 \times 3$. How many channels does it have?
1. 600
 2. 300
 3. 3
 4. 0
9. The size of an image is represented as $600 \times 300 \times 3$. How many pixels does it have in a single row?
1. 600
 2. 300
 3. 3
 4. 0
10. The size of an image is represented as $600 \times 300 \times 3$. How many pixels does it have in a single column?
1. 600
 2. 300
 3. 3
 4. 0
11. The size of an image is represented as $600 \times 300 \times 3$. What would be the type of the image?
1. Bitmap
 2. Grayscale
 3. Color
 4. None of the above
12. Object _____ technique is used by news channel to blur the face of a victim or any person whose identity is not to be revealed?
1. Identification
 2. Segmentation
 3. Verification
 4. Localization
13. The process of repeatedly locating and classifying all objects of interest in the image is known as object _____.

1. Verification
 2. Detection
 3. Localization
 4. Segmentation
14. The process of creating a 3D model of a scene inputted through images or video is known as _____.
1. Scene reconstruction
 2. Object tracking
 3. Video motion analysis
 4. Image restoration
15. _____ is used to estimate the velocity of objects in a video, or the camera itself.
1. Scene reconstruction
 2. Object tracking
 3. Video motion analysis
 4. Image restoration
16. Computer vision cannot be used for _____ systems.
1. Home security
 2. Augmented reality
 3. Speech recognition
 4. Gaming
17. Video cameras placed in/on _____ produce a noisy image.
1. Dark room
 2. A road with heavy raining
 3. Bright sun
 4. All of these
18. Overlapping of image occurs when an image has _____.
1. Missing data
 2. Different lighting
 3. One object hidden below another
 4. Been taken from different angles
19. CNN is a branch of _____ learning.
1. Deep
 2. Supervised
 3. Unsupervised
 4. Reinforcement

20. _____ layer reduces the size of inputs by removing unnecessary information to speedup computations.
1. Convolution
 2. Pooling
 3. Fully connected
 4. All of these
21. _____ layer performs computational tasks.
1. Convolution
 2. Pooling
 3. Fully connected
 4. All of these
22. Which function is used to change the size of an image?
1. imread()
 2. resize()
 3. rotate()
 4. size()
23. Which function is used to print the size of an image?
1. shape()
 2. resize()
 3. rotate()
 4. size()
24. Which function is used to print the number of rows, columns and colours?
1. shape()
 2. resize()
 3. rotate()
 4. size()
25. Which technique removes noise from an image?
1. Blurring
 2. Smoothing
 3. Detecting edges
 4. Resizing
26. Which technique used to hide the details or any confidential information in an image?
1. Blurring
 2. Smoothing
 3. Detecting edges

4. Resizing
27. An ANN does not have _____ layer.
1. Input
 2. Output
 3. Convolution
 4. Hidden
28. Which of the following statements is incorrect about a CNN?
1. CNN is more efficient than ANN and has a higher success rate.
 2. CNN requires for data for training than an ANN.
 3. CNN uses a greater number of parameters for learning than an ANN.
 4. Computational cost of CNN is more than that of ANN.
29. How many different colours can a greyscale image store?
1. 2
 2. 256
 3. 20
 4. 16,535
30. A _____ image stores either of two values in a pixel.
1. Bitmap
 2. Greyscale
 3. Coloured
 4. All of these
31. An ANN does not have _____ layer.
1. Input
 2. Output
 3. Convolution
 4. Hidden
32. Image _____ is often measured in Dots Per Inch (DPI) and Pixels Per Inch (PPI).
1. Width
 2. Height
 3. Resolution
 4. Brightness
33. An image with resolution 2048 x 1536 has _____ number of pixels in a row and _____ pixels in a column.
1. 2048, 1536
 2. 1536,2048

3. 3 mega pixels

4. All of these

If the values of two neighbouring squares on the grid differ by more than 50, draw a thick line between them. (Taken from NCERT. Can add 10 to all the numbers).

If the values of two neighbouring squares on the grid differ by more than 40, draw a thick line between them.



IDENTIFY THE PROCESS

Credit:

Redchanka / _____
Shutterstock

Credit:

Zapp2Photo
/ _____
Shutterstock

Credit:

Tsomka / _____
Shutterstock



ANSWERS

Fill Up

1. Content-based image retrieval
2. 'pixel'
3. 360-degree content
4. Digital twins
5. smoothing
6. Digital twins
7. Convolutional

8. Object identification
9. tracking
10. Facial recognition
11. Augmented
12. visual
13. Adversarial
14. Convolutional
15. OpenCV
16. pixels
17. Resolution

State True or False

1. True
2. True
3. False
4. False
5. True
6. False
7. True
8. True
9. True
10. False
11. False
12. True
13. False
14. True
15. True
16. False
17. True
18. False
19. True
20. True
21. False
22. True
23. False
24. False

Multiple Choice Questions

1. b
2. c
3. b
4. b
5. a
6. c
7. d
8. c
9. a
10. b
11. c
12. b
13. c
14. a
15. c
16. c
17. d
18. c
19. a
20. b
21. a
22. b
23. d
24. a
25. b
26. a
27. c
28. c
29. b
30. a
31. c
32. c
33. a

Previous chapter

< [5. Reasoning and Learning in Artificial Intelligence](#)

Next chapter

[7. Natural Language Processing](#) >

Table of contents expanded

7

Natural Language Processing

7.1 What is Natural Language Processing?

Natural language processing is the manipulation or understanding of text or speech by a computer software. As humans interact, they understand each other views, and respond with an appropriate answer; similarly, NLP systems also interact, understand and respond to queries. The contrast between Natural Language and Computer Language can be understood from **Table 7.1**

Natural language processing, a field of artificial intelligence, studies interactions between computers and human (natural) languages. It applies **machine learning** algorithms to **text** and **speech**. For example, NLP can be used to create intelligent systems for **speech recognition, machine translation, question answering, spam detection, document summarization, predictive typing, named entity recognition** and **autocomplete**, to name a few.

Morphology studies the formation of words from stems, prefixes and suffixes.

These days, smartphones and computers support so many applications that all modern operating systems have a built-in speech recognition module. For example,

Cortana, the **virtual assistant** of Microsoft Windows, recognizes human **voice** to set up reminders, open apps, send emails, search information, play games, get latest updates on weather, etc.

Siri, the virtual assistant of the Apple Inc.’s iOS, watchOS, macOS, HomePod and tvOS helps users perform their tasks with **voice commands** like starting a call, texting someone, sending an email, clicking a picture, setting a timer, opening an app, setting an alarm, using navigation and so on.

TABLE 7.1 Differences between Natural language and Computer Language

Natural Language	Computer Language
Natural language has a very large vocabulary.	Computer language has a very limited vocabulary.
Natural language is easily understood by humans.	Computer language is easily understood by machines.
Natural language is ambiguous in nature.	Computer language is unambiguous.

Gmail, the most popular email service developed by Google, uses NLP to detect and filter spam emails.

The main reason behind the rise of NLP can be simply summarized in two points.

- 1. Large volumes of textual data:** Natural language processing helps computers communicate with humans in their own language. NLP empowers computers to read text, hear speech, interpret it, measure sentiment and determine which parts are important.
Today’s machines are supposed to interact with humans in a consistent, unbiased way without getting tired. With increasing use of machines in our day-to-day lives (from medical records to social media), enormous amount of unstructured data is being generated.
- 2. Structuring the unstructured data:** Human language is complex and diverse as humans express the same thing in infinite ways both in written and spoken form.

You will be surprised to know that there are 6500 human languages, out of which 121 are frequently spoken. Every language has its unique set of grammar and syntax rules, terms and slang. Moreover, while writ-

ing, we often misspell or abbreviate words, or omit punctuation. Similarly, while speaking, we pronounce different words differently (in regional accents). Some people mumble, stutter and borrow terms from other languages which further increases the complexity of processing natural languages.

Techniques like deep, supervised and unsupervised learning are widely used for modeling human language to resolve ambiguity in language.

How are NLP Systems useful?

NLP systems break down text/speech in human language into shorter, elemental pieces, to understand relationships that exist between the pieces for interpreting its meaning. Such systems are then used for the following tasks.

1. **Generate a summary of the document, generate keywords for efficient searching and indexing.**
2. Accurately capture the meaning and themes in text and use it for optimization and forecasting.
3. Automatically extract structured information from textual sources.
4. **Perform sentiment analysis to identify user's opinion within large amounts of text.**
5. Speech-to-text and text-to-speech conversion to transform voice commands into written text, and vice versa.
6. Automatic translation of text or speech from one language to another.

Spam emails are those emails that are sent in bulk to recipients without their permission. They are often sent for commercial purposes.

7.2 Everyday NLP Examples

Some widely used NLP applications in our everyday lives include the following:

Just check the spam folder in your email, it has already separated some emails with common words as spam. Spam filtering uses NLP to compare words in spam email to those present in valid emails to identify junk mail (refer **Fig. 7.1**).

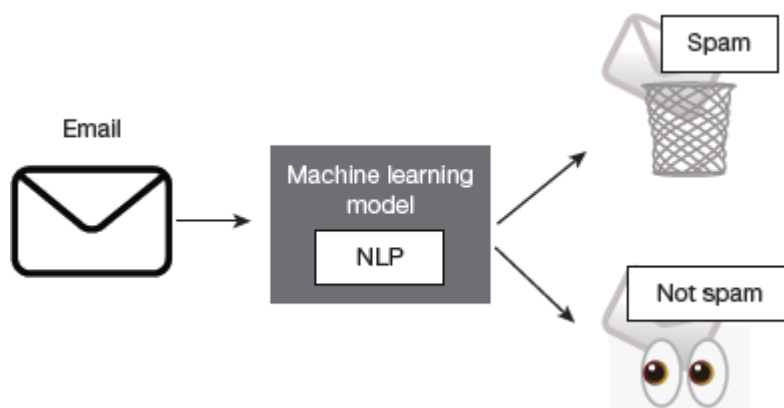


FIGURE 7.1 Classifying Email as Spam or Not a Spam

Whenever you miss a phone call, you get an automatic transcript of the voicemail in your email inbox or smartphone app. This is possible due to NLP's capability to perform speech-to-text conversion.

NLP is also the underlying technology when we do a voice search.

Speech recognition is used for converting spoken words into text. It is used in applications where the user gives voice commands to smartphones, voice biometrics device, home automation systems or dictates in software like Microsoft Word, etc. (refer **Fig. 7.2**).



Credit: Text Recognition

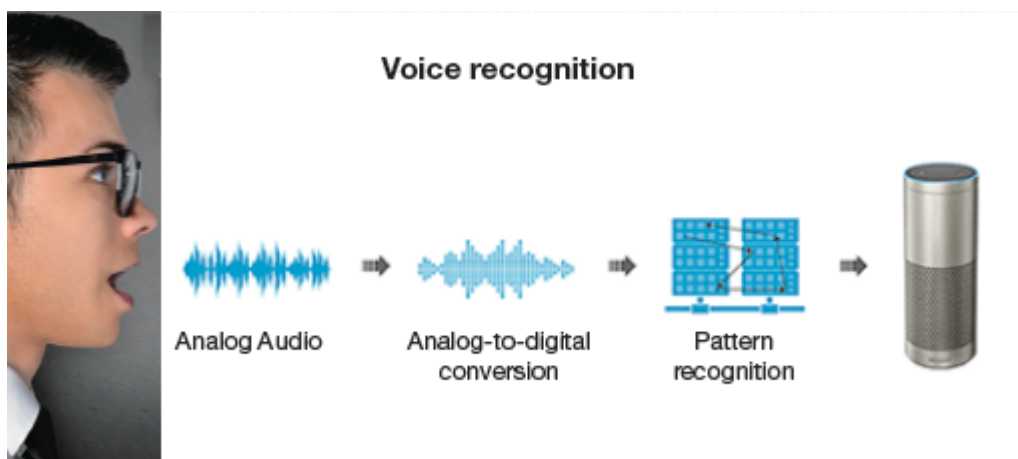
FIGURE 7.2 Performing Commands using Speech Recognition

In these systems, NLP is used to give voice commands to devices to perform a particular task (ex, play a song or search information about a topic on a search engine).

NLP systems are being used by companies to understand what their customers think and say about them on social media, emails, feedbacks and reviews.

Companies perform sentiment analysis (also known as **opinion mining**) to assess the attitude, behaviour and emotional state of customers. For this, NLP assigns values to the text (positive, negative or natural) and also identifies the mood of the context (happy, sad, angry, etc.) in which the customer has written that feedback/review.

Did you ever think what happens when we speak to our device? Actually, the microphones of the device accept our voice as input and plot the graphs of sound frequencies. Every sound has a unique frequency graph. This fact helps NLP to recognize each sound and compose an individual's words and sentences (refer **Fig. 7.3**).



Credit: fuzzbones / 123RF

FIGURE 7.3 Voice Recognition

Chatbots are NLP, AI-based systems that are specifically designed to interact with humans through text or speech. They are extensively used these days to offer 24 X 7 assistance to customers by handling their queries. Chatbots actively learn from each interaction to gain better understanding of user's intent so that they can be used by companies to au-

tomate repetitive and time-consuming tasks. This allows employees to focus on more fulfilling tasks.

Spelling correction is a very important tool provided by Microsoft Corporation in its application software like MS-Word and PowerPoint. The NLP program embedded in this software automatically checks the spelling in the text. If the spelling is incorrect, it notifies the user and also gives suggestion on what could be the correct spelling.

Topic classification uses NLP to organize unstructured text into categories. For example, companies use NLP to analyse hundreds of open-ended responses to surveys to gain insights from customer feedback. Topic classifier tags all keywords in seconds and then automatically routes them to the right person for speedy response.

NLP is used to translate text/speech written in one language to another; for **example:** Google Translator.

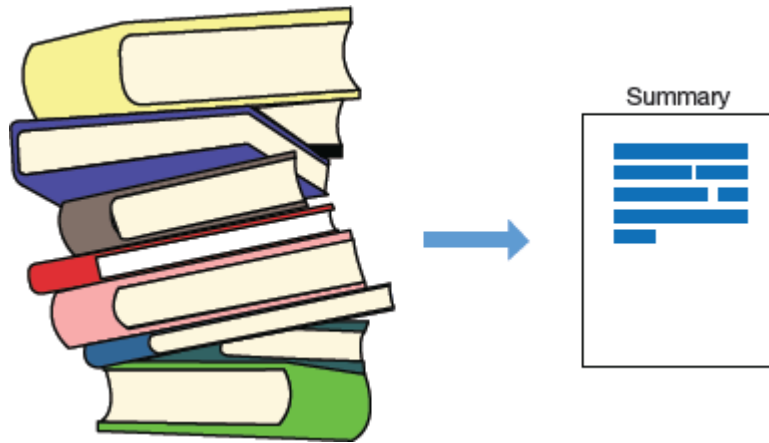
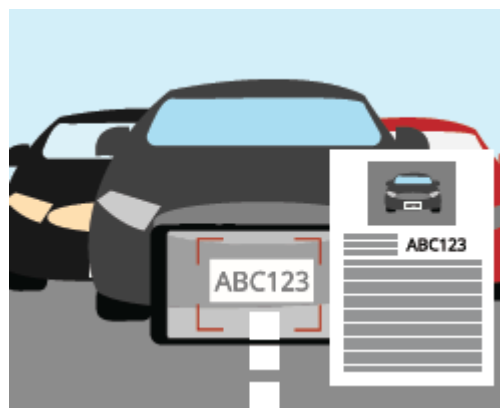


FIGURE 7.4 Text Summarization



Credit: Piscine26 / Shutterstock

FIGURE 7.5 Text Recognition

Text Recognition (in an image or video): These days, cameras are installed at major crossings. Have you ever noticed that if someone skips stopping during a red signal then automatically a challan comes to his/her house in a day or two? Did you try to think about the technology being used? Basically, the camera clicks the image of the vehicle along with its number plate (refer **Fig. 7.5**). NLP tool is then used to extract number from the number plate. Obviously, once we get the exact vehicle number, we can easily retrieve information about the owner such as name, age, address, phone number, date of purchase, registration number, etc. In this application, NLP is being used to recognize text from an image.

Text Summarization: NLP tools can easily read, understand and summarize long paragraphs of text or even full article into a shorter narrative without changing its inherent meaning (refer **Fig. 7.4**). Summarization is

very helpful in various applications. For example, researchers need to read several papers and articles published before finalizing their work. It is a very tedious job to read those documents line-by-line and make notes. A summarization tool is of great essence in this domain.

Lawyers and judges have to read through large volumes of documents to develop an understanding of the ongoing case. Even in this case, the summarization tool can read the case files to create a short abstract that can be used by legal practitioners, thereby saving their time and expediting the pending cases resolution. Text summarization is done in two ways.

First, key phrases are extracted from the document and combined to form a summary. This is known as **extraction-based summarization**.

The second option is **abstraction-based summarization** in which the source document is shortened. We create new sentences from the original content in this step. This is in contrast to extraction-based technique, in which we only utilized the phrases that were present. It is possible that the phrases formed by abstractive summarization are not present in the original text. Therefore, although abstraction models are more complex to build, they are more efficient than extraction.

The process of constructing a concise, cohesive and fluent summary of a lengthier text document, which includes highlighting the text's important points, is known as text summarization.

Information Extraction: Information extraction is the process of extracting information from unstructured textual sources to identify entities so that they can be efficiently stored in a database. Real-world example of information extraction is using email extracts to add relevant in your Calendar.

Information extraction extracts relevant information from unstructured text data and presents it in a structured format.

Information extraction techniques are widely used in healthcare, judiciary and to understand social media interactions and streams, online news, government documents, corporate reports and more to automate certain tasks (like smart content classification); integrate search, management and delivery tasks; identify patterns and trends to uncover hidden relationships. For example, go through the cricket news article given below.

Indian captain Virat Kohli was dismissed cheaply for just 2 in Wellington on Friday by debutant Kyle Jamieson extending a rare lull in the batsman's stellar career. Throughout the ongoing New Zealand tour, Kohli has managed to score just a single fifty across 8 innings in all 3 international formats.

From this unstructured text, we can extract the following information.

- Country – India, Captain – Virat Kohli
- Batsman – Virat Kohli, Runs – 2
- Bowler – Kyle Jamieson
- Match venue – Wellington
- Match series – New Zealand
- Series highlight – single fifty, 8 innings, 3 formats

Structured data helps us to take full advantage of powerful query tools like SQL for further analysis.

7.3 Chatbot

Chatbot (refer **Fig. 7.6**) is an artificial intelligence (AI) program that simulates conversation (or a chat) with a user in natural language through messaging applications, websites, mobile applications or by phone. You must have seen that many websites offer a real-time chat service that helps you to find relevant information on that website. Such an application provides a pleasant user experience and faster and simpler customer service interaction. At the same time, it also allows companies to improve their operational efficiency by reducing the typical cost of customer ser-

vice. Employees that were dedicatedly providing customer support can now be utilized to do more fruitful jobs. Companies are now able to save thousands of rupees per month by deploying a chatbot instead of a human employee dedicatedly working to provide customer support.

The best part of a chatbot is that it allows users to pick up a conversation from the point where they last left. This has been possible because chatbots have a memory that implicitly learns about a use and stores key information about it for future use. The advantages of chatbots are summarized in **Fig. 7.7**

Users can query about multiple things one after the other without annoying the information provider.



Credit: Tero Vesalainen / Shutterstock

FIGURE 7.6 Chatbot

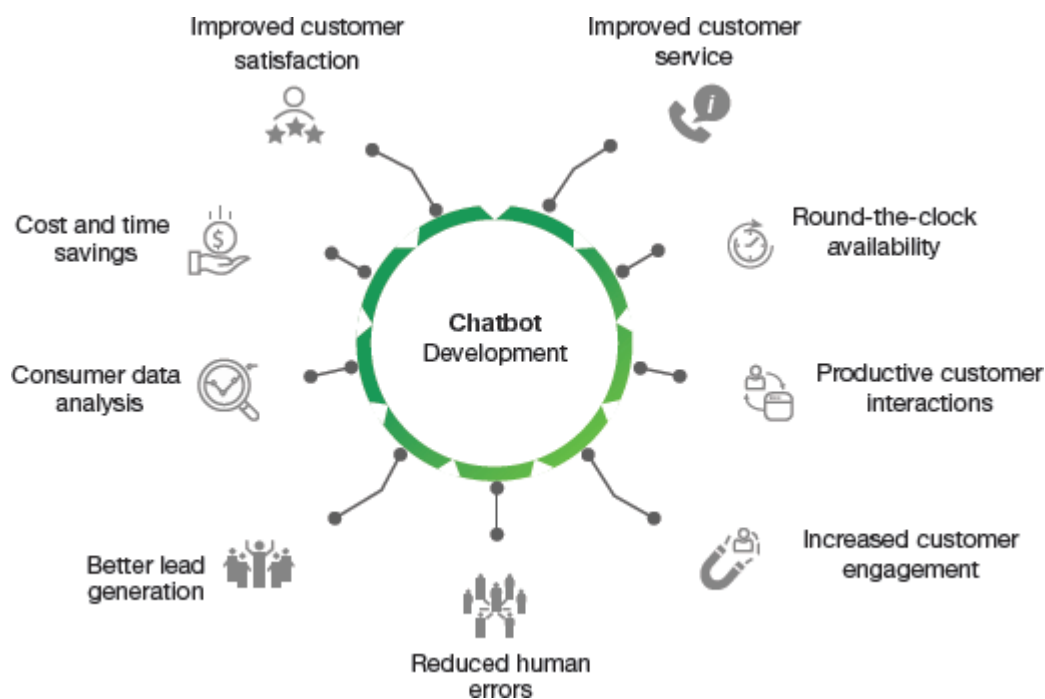


FIGURE 7.7 Advantages of Chatbots

7.3.1 Open and Closed Chatbots

A chatbot can be categorized as open or closed depending on its usage. While open chatbots use ML techniques to learn from their interactions with users, closed chatbots, on the other hand, execute a specific conversation flow.

Thus, open chatbots support a more natural way of conversing with humans. But their success very depends on how well they have been trained. Closed chatbots comparatively have a lower risk of providing poor user experience.

7.3.2 Uses of Chatbots

1. **Customer service:** Chatbots can be programmed to collect registration data, provide information, answer frequently asked questions, etc. This gives a pleasant experience to customers. Users can now interact in a simpler and faster way.
2. **Online purchases:** Chatbots assist users while shopping online to select a product of their utility and then make the payment easily. They provide information, solve problems faced by users during shopping and perform searches to identify similar products.
3. **Sending information and news about latest updates in products or about discount offers** automatically to customers and potential buyers of the product.
4. **Collect Information:** Chatbots can be designed to collect some useful information about users, such as email, phone, etc. This information can be filtered (to select most appropriate ones in the current context) and then be forwarded to sales representatives for a follow-up conversation.
5. **Make Appointments:** Chatbots can automatically make appointments by displaying availability on a calendar and allowing the user to select a suitable date.

Healthcare websites deploy chatbots in their websites to provide patient information. A patient can enter his symptoms. The NLP-based chatbot will check those symptoms with known ailments. After analysing the basic issues, the chatbot can then *assist patients in booking an appointment* (refer **Fig. 7.8**), *finding a doctor and more*.

Chatbots also help users to *find an appropriate insurance policy*. For this, the chatbot collects useful information from users and then tailor products according to the consumer's needs. These chatbots not only help users find a suitable policy but also complete the payment in a hassle-free way, file for a claim, renew policies, check their coverages, track status of policy/claim, to name a few.

Children of this era, known as Gen Z, are dealing with more mental health, body image, and loneliness issues than others. These *children can use chatbots that can gauge their emotional state based on conversational context and respond with compassion.*

6. **Builds trust and improves customer engagement:** Chatbots help companies to keep their customers satisfied, well-informed as they are assisted in everything they do. A chatbot instantly responds 24 X 7 to provide immediate answers to simultaneous users at once.

7.3.3 How a Chatbot Works?

A chatbot works on a request-response model. When a user requests information, the *chatbot* analyses the request to **identify what information the user exactly wants.**

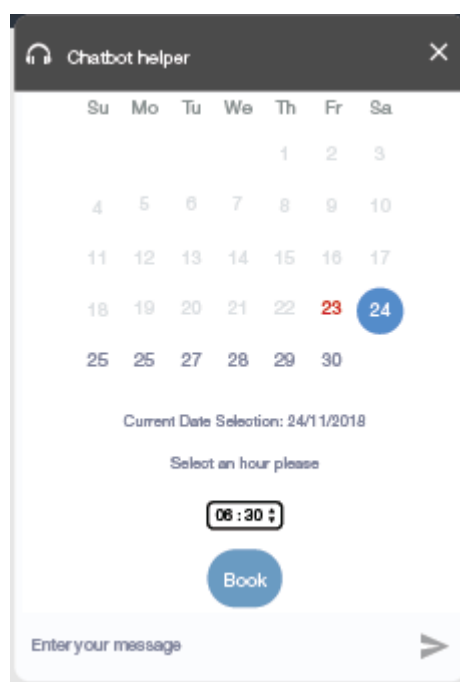
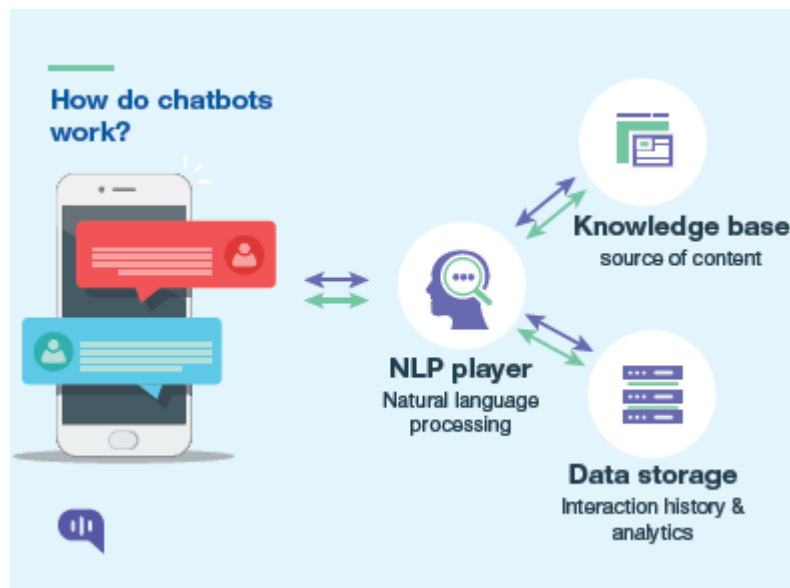


FIGURE 7.8 Chatbot for Booking Appointment

Once the intent of the user's request is correctly known, the chatbot then searches its knowledge base (refer **Fig. 7.9**) to give the most appropriate response. Many-a-time, the user asks for some information or data. In such cases, the chatbot interacts with the back-end system to retrieve the data required to satisfy the user's request.

If there is no information available in the current context or if the chatbot fails to interpret the user's request, then the chatbot may reply with a generic and predefined text.



Credit: Astrovector / Shutterstock

FIGURE 7.9 Chatbots converse using their knowledge base

7.3.4 Types of Chatbots

Chatbots apply a combination of different AI techniques including natural language processing, machine learning and semantic understanding. There are different types of chatbots (refer **Fig. 7.10**). Each one serves a different purpose and solves a different issue. Depending on the use case you want to address, some chatbot technologies are more appropriate than others.

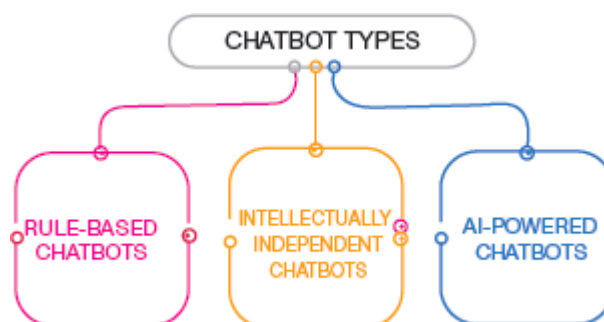


FIGURE 7.10 Types of Chatbot

1. **Rule-based chatbots:** As the name suggests, these chatbots follow pre-defined paths while conversing with the user. As the conversation progresses, the user is offered several options that correspond to their inquiry (refer **Fig. 7.11**). Based on the option selected by the user, the chatbot decides the next step in the conversation.

AI-powered chatbots get smarter over time.

Though these chatbots make the interaction simple for both the user as well as the chatbot program, it can be used only to answer basic customer inquiries (like the status of one's delivery, get some information, etc.).

2. **Menu-based chatbots:** These chatbots are a variant of rule-based chatbots that present a menu or buttons showing different options available to the user. Instead of typing text, the user has to just click on those buttons or options to interact with the system. Such chatbots lack critical understanding and flexibility. These bots can only automate situations that they have learned and are likely to arise.

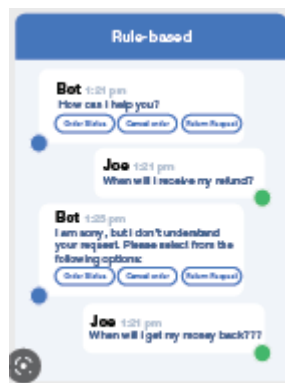


FIGURE 7.11 Rule Based Chatbot

3. **Intellectually-independent chatbots:** These chatbots use machine learning (ML) algorithms to learn from user's requests and bot's reply. During every conversation, the bots use keywords to recognize recurring patterns and common user behaviours to decide what to do when little or no human assistance is available. For example, if the user types 'Facing problems in account login', then the bot breaks the sentence into keywords like 'problem,' 'login,' and 'account,' and then provides a predefined answer based on those keywords.
4. **AI-powered chatbots:** These are hybrid bots that are a good combination of both the bots—rule-based and intellectually-independent chatbots (refer **Fig. 7.12**). These chatbots extensively use AI techniques to

simulate human intelligence, thereby making machines work and think like people. They process user's queries (using neural network) before responding to them. AI-based chatbots also learn from previous experience and use reinforced learning to keep evolving.

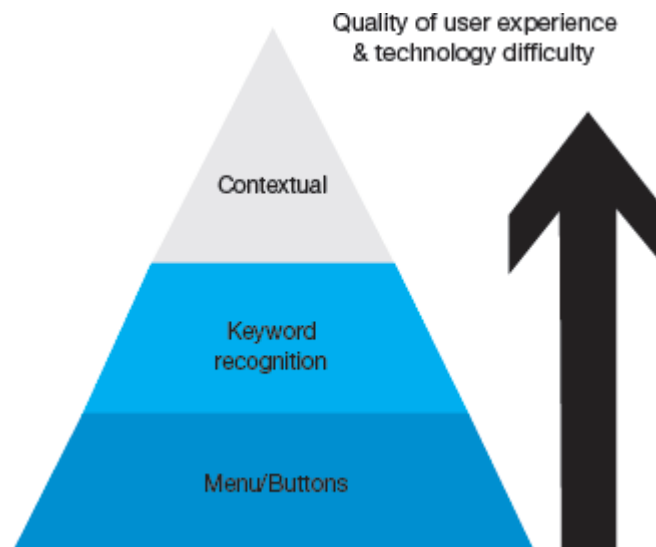


FIGURE 7.12 AI-powered chatbots

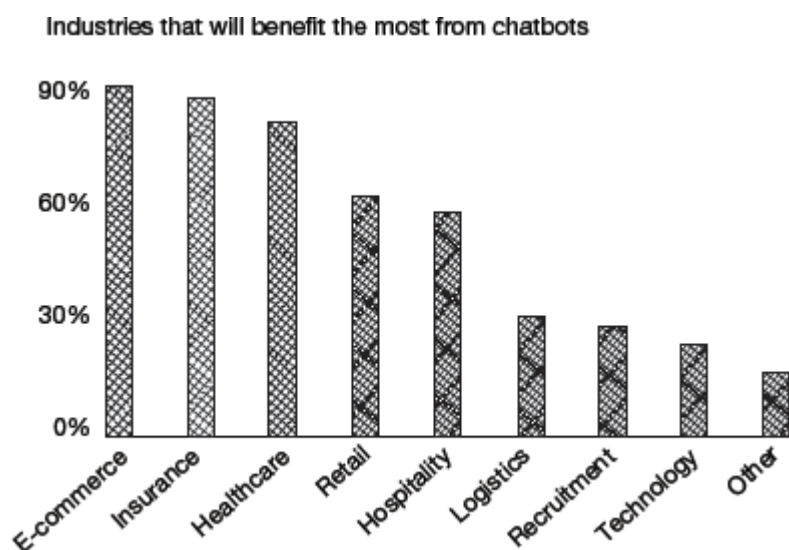


FIGURE 7.13

AI-powered chatbots use a predefined flow to ensure accurate responses to a user's inquiry. If no such path is available, they are also equipped to understand free language. For example, Shivam may give a voice command to a chatbot saying, 'Book me a flight to Chennai.' But Shubham may intend to do the same task by saying, 'I need a flight to Chennai.' Such complexities can only be handled by an AI-based or NLP-based bot

as it identifies the language, context and intent to react accordingly. A rule-based bot only understands a pre-defined set of options.

We can also classify chatbots based on where they are deployed. For example, chatbots can be implemented into messengers, standalone applications or on a website. **Figure 7.13** shows the application of chatbots in different sectors.

7.4 How Does Natural Language Processing Work?

NLP facilitates computers to understand natural language just like humans. An NLP system takes a language as input either in spoken or written form. It processes the language using AI techniques to make sense of it in a way a computer can understand.

Did you know that Facebook's translations achieved superhuman performance in 2019?

Humans have different sensors like ears to hear and eyes to see. Similarly, computers can read content through programs and listen to audio through microphones. While humans process inputs in their brain, computers process in the CPU by following program instructions. During processing, the inputs are converted to codes that can be interpreted by a computer.

7.5 Components of NLP

NLP is a combination of NLU and NLG techniques.

Natural language understanding (NLU) deals with mapping the given input in natural language into useful representations.

Natural language generation (NLG), on the other hand, uses content in internal representation to produce meaningful sentences in a natural language. NLG is performed using the following:

1. **Text planning** retrieves relevant content from knowledge base.

- 2. **Sentence planning** selects required words to form meaningful sentences in the right tone in current context.
- 3. **Text realization** finally converts the sentence plan into sentence structure.

NLU is harder than NLG.

TABLE 7.2 Differences between NLU and NLG

NLU	NLG
Reads and interprets language.	Writes or generates language.
Generates non-linguistic outputs from natural language inputs.	Generates natural language outputs from non-linguistic inputs.

7.6 Steps in NLP

From the above discussion, we can summarize the NLP process (refer **Fig. 7.14**) in five steps.

Step 1. Perform Lexical Analysis: Lexicon of a language refers to the collection of words and phrases in that language. Therefore, in this step, the structure of words is identified and analysed by dividing the entire text into paragraphs, sentences and words.

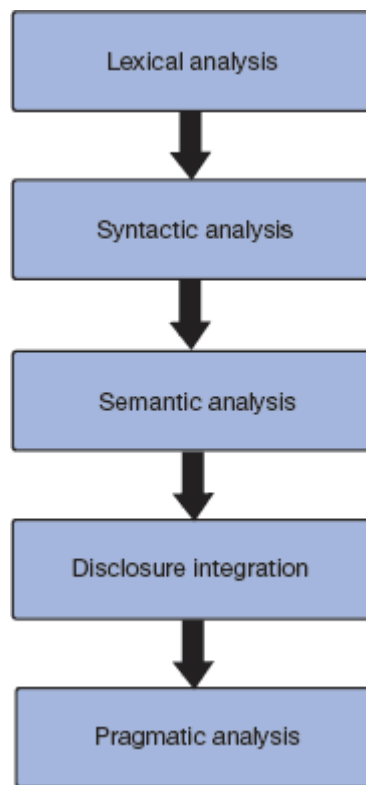


FIGURE 7.14 Steps in NLP

Step 2. Syntactic Analysis or Parsing: As discussed, all the words in a sentence are analysed as per grammatical rules of that language. Words are rearranged to study any relationship that exists between them to ensure that the sentence is syntactically correct. For example, ‘The boy was caught by the ball’ is rejected by English syntactic analyser.

Step 3. Semantic Analysis: In this step, the exact meaning or the dictionary meaning of a text is drawn. It is done by mapping syntactic structures and objects in the task domain. For example, the phrase ‘cold fire’ is rejected by the semantic analyser of English language.

Step 4. Discourse Integration: The meaning of a sentence is often studied with respect to the meaning of the sentence just before it. Also, a sentence affects the meaning of sentence immediately succeeding it. Discourse integration studies the meaning of a sentence with respect to meaning of sentence preceding as well as succeeding it in sequence.

Step 5. Pragmatic Analysis: In this step, what was said is re-interpreted on what it actually meant. This is done only in situations that require real world knowledge.

Phonology is the study of organizing sound systematically.

NLP, AI, Machine Learning: What's the Difference?

NLP, AI and ML are important terms that are often used interchangeably. But you must know that, NLP and ML are both subsets of AI.

With NLP, machines can understand what the user wrote or spoke and then respond accordingly. NLP uses ML algorithms to automate the process of delivering accurate responses.

For example, AI-powered chatbots use NLP to interpret what users say and what they intend to do, and ML to automatically deliver more accurate responses by learning from past interactions.

The untapped potential in unstructured text

People's thoughts, research, opinions, facts and feedback transfer into the digital world through social media feeds, legal case files, electronic health records, contact centre logs, warranty claims and more. Natural language processing uncovers the insights hidden in the word streams.

7.7 Phases of NLP

There are two phases in any NLP project—data pre-processing and data processing. In this section, we will learn about both of them.

Data Pre-Processing

In this phase, textual data is prepared and cleansed so that it can be efficiently and correctly analysed by algorithms. Given below are certain ways in which data is prepared for analysis.

1. **Tokenization:** This process breaks down text into smaller units called tokens. There are two types of tokenization.

Sentence tokenization, also known as **sentence segmentation**, divides a **text** written in human language **into** a sequence of individual sentences.

Word tokenization or **word segmentation** **divides a text** of written language **into** its constituent words.

2. **Stop word removal:** Here, all the commonly used words like 'is', 'the', 'and', 'are', 'to', 'on', 'in', etc. are removed from the text so that unique words that provide most of the information about the text are only studied.

Morphological segmentation divides words into smaller parts called morphemes. For example, the word untestably is broken into 'un,' 'test,' 'able' and 'ly' as morphemes.

3. **Lemmatization and stemming:** Text lemmatization and stemming are used for grammatical reasons to support documents that contain **different forms of a word**. For example, words like **run, runs, running** are related words having a similar meaning. In this case, **stemming and lemmatization can be used to reduce all these related words to a common base**.

For example, am, are and is have a common base: 'be'. Similarly, run, runs, running can be reduced to the base form: 'run'.

Stemming and lemmatization are special cases of **normalization**.

However, they are different from each other.

Lemmatization *performs its task with the help of a vocabulary and morphological analysis of words, to return the base or dictionary form of a word, which is often known as the lemma.*

Stemmer, on the other hand, operates **without knowledge of the context**. It cannot understand the difference between words having a different meaning in a different context. However, stemmers are **easier to implement** and usually **run faster**.

4. **POS Tagging** or parts of speech tagging, also known as grammatical tagging, is a process that marks words in text format for a particular part of a speech, depending on its definition and context. It reads text in a language and assigns some specific token (parts of speech) to each word. For example, a word is tagged as noun, verb, adjective, determiners, etc. Like in the sentence, 'The bird flew.' bird = noun, flew = verb.

Data Processing

Once the data is pre-processed, it is processed using a particular algorithm. Though there are many NLP algorithms, two main types are commonly used:

1. **Rules-based system:** A rule-based system uses carefully designed linguistic rules for NLP.
2. **Machine learning-based system:** These systems use statistical methods. The ML system is first trained using a training data set. During the learning phase, the system learns how to perform tasks and adjust certain parameters either to achieve higher accuracy or to process more amount of data. These days, usually, a combination of machine learning, deep learning and neural networks is used for NLP algorithms so that the system can automatically hone their own rules through repeated processing and learning.

7.8 Syntax vs Semantic Analysis

Syntax is the arrangement of words in a sentence to make grammatical sense. Syntax of a statement is assessed by checking its meaning from a language based on grammatical rules. During syntactic analysis, processes like tokenization, stemming, lemmatization, stop words removal, POS tagging are performed.

Semantics involves the use of and meaning behind words. NLP applies algorithms to understand the meaning and structure of sentences.

Semantics techniques include the following:

1. **Word sense disambiguation:** In this step, the meaning of a word is analysed based on the context in which it is used. For example, consider the sentence, 'I want to present a watch as her birthday present'. Here, the word 'present' has different meanings. An algorithm should be used here to understand that the first 'present' means 'to give' and the second 'present' means 'a gift'.
2. **Named entity recognition:** Named entity recognition or NER identifies words that can be classified in different groups. That is, semantics of the text is analysed to differentiate between entities that are vis-

ually the same but otherwise different. NER is especially useful when we have to analyse a piece of text and identify all mentions of a certain company or product. For example, in the sentence, ‘McDonald’s son, John, went to McDonald’s on his friend’s birthday.’ Here, the algorithm must distinguish between the two instances of ‘McDonald’s’, the first instance is a person and the second instance refers to a restaurant.

3. **Natural language generation:** NLG uses a database to determine semantics behind words and generate new text. For example, an NLP algorithm can automatically summarize a business meeting or an article published about the business. Another example could be to automatically generate news articles or tweets after training the algorithm using certain text.

7.9 Applications of Natural Language Processing

Some applications of NLP are discussed here.

1. **Text classification:** In text classification, tags are assigned to text for categorization in different groups. This is very important for applications like sentiment analysis, which help an NLP algorithm to determine the sentiment, or emotion behind saying/ writing a piece of text. For example, reviews can be analysed using an NLP algorithm to determine if a greater number of users are feeling positive about a new product or writing negative about it. This analysis can be very useful for detecting the intent of potential customers that can, in turn, be used to predict what the speaker or writer may do—buy the product or not, based on the text they are producing.
2. **Text extraction:** Text extraction helps to automatically summarize text and search important pieces of data. For example, keyword extraction retrieves the most relevant words from the text, which can be useful in applications such as the following:
 1. Search engine optimization
 2. Pull out the most frequently used words in the text
 3. Named entity recognition, which extracts the names of people, places and other entities from text.

Google's Director of Engineering Ray Kurzweil predicts that AIs will 'achieve human levels of intelligence' by 2029.

3. **Machine translation:** The NLP algorithm translates text from one language to another without human intervention. For example, we often use Google Translator to convert text from English to Hindi or vice versa.

Translation tools also facilitate businesses to communicate with customers in their language of choice. This helps in breaking barriers for globalization. Moreover, these translation tools are trained to understand specific terminology that are relevant for a specific industry, like finance or medicine.

4. **Natural language generation:** In this process, algorithms are used to analyse unstructured data and automatically generate content based on that data. For example, GPT3 model analyses unstructured text to generate articles.
5. **Spam detection:** Spam detection technologies use NLP's text classification capabilities to scan emails to check for words that indicates whether it is a spam email or may lead to phishing attack. For example, easy loan, free gifts, bad grammar, threatening language, urgency, misspelled company names, etc. are common points that indicate a spam email.
6. **Spelling correction:** Applications like MS Word support features for spelling correction. This has been possible just because of NLP.
7. **Search engines** use NLP to display relevant search results based on keywords provided by users. This helps even a person who is not very technology-savvy to easily find information that he/she is looking for.
8. **Chatbot:** We all have used chatbots on some website or the other. These chatbots are small AI systems that interact with humans through text or speech. They use NLP algorithms to provide the customer's chat services 24 X 7. This not only speeds up response time but also helps to handle multiple queries simultaneously, thereby freeing up humans from answering repetitive questions. Chatbots continuously learn from each interaction. This helps to improve its performance by better understanding user's intents. In case a chatbot is not able to interpret or respond to a user's query, it passes that query to a human.

For example, chatbots in healthcare, take data from patients to assess their symptoms, and determine next steps to be taken (like set up appointments with the right doctor or recommend treatments).

Technically, chatbots provide automated answers to common customer queries. They use pattern recognition systems with heuristic responses to interact with users. These days, chatbots have evolved from simple answering programs to AI-powered systems that can handle more complicated requests, thereby making conversational experiences increasingly intuitive.

9. *Question-Answer Systems: Question-answer systems are intelligent systems that are specifically designed to respond to customer queries. It is different from chatbots as it has a large knowledge base and possess a good understanding of language. Unlike chatbots, QA systems are far beyond canned answers. They can deal with textual data, audio, video and images. For example, they can answer questions like ‘When did India get independent?’ or ‘How do I get to the airport?’*

Question-answer systems are present in social media chats and tools such as Siri and IBM’s Watson. In 2011, IBM’s Watson computer competed on Jeopardy. Jeopardy is a game show during which the contestants ask questions and answers are expected. The entire IT industry was astonished to observe that when computer competed against the show’s two biggest all-time champions, it won the first place.

The functions given above are widely used across a range of real-world applications such as the following:

1. Customer feedback analysis to analyse user reviews on social media.
2. Customer service automation uses voice assistants on the other end of a customer service phone line. These assistants use speech recognition to understand customer’s query and then to re-direct the call to the appropriate department.
3. Automatic translation like Google Translate, Bing Translator and Translate Me translate text from one language to another.
4. Academic research and analysis analyses a large number of academic material and research papers to create a relevant summary.
5. Analyses and categorizes medical records to predict diseases
6. Word processors use NLP to check plagiarism and do proofreading.

7. NLP algorithms are used to analyse market history and around 10,000 documents to gain an insight into financial trading and predict value of stocks.
8. HR teams these days use NLP to summarize the CVs of applicants and recruit the best talent.
9. AI attorney uses NLP to automate routine litigation tasks.
10. In healthcare, NLP is used to convert sign language into text. This helps deaf people to communicate with people who don't understand sign language.
11. In the aircraft maintenance industry, NLP is used to find relevant information from aircraft manuals (that me 100s of pages) based on problem description given by the pilot.

7.10 Pros and Cons of Natural Language Processing

The most prominent advantage of NLP is that it facilitates efficient interaction between humans and computers. Other benefits include the following:

1. Improved accuracy
2. Enhanced efficiency of documentation
3. Automatic summarization of a large number of unstructured and complex text
4. Interpretation of spoken words by personal assistants like Alexa
5. Chatbots for customer support
6. Perform sentiment analysis
7. Once implemented, NLP solutions are less expensive
8. Save time and effort of human counterparts
9. Faster customer service response times 24 X 7, results in improved customer satisfaction
10. Pre-trained NLP solutions are easily available to facilitate different applications of NLP.

Limitations of NLP Systems

Some limitations of NLP systems are as follows:

1. Training a machine learning model is often time-consuming. Developing and training a new model to get the best results takes several weeks.
2. Remember that no machine learning model is 100% accurate in every situation. Hence, NLP solution may not be reliable. There is always a possibility of errors in predictions.

Challenges

Human languages are often ambiguous for machines as one sentence can have different meanings and several sentences may mean the same thing. Hence, we can say that this difficulty mainly persists due to different levels of ambiguity.

Lexical ambiguity exists due to confusion in different words. For example, whether the word 'board' should be treated as noun or as a verb.

Syntax level ambiguity occurs because a sentence can be parsed in different ways. For example, the sentence 'He lifted the beetle with red cap.' can be interpreted in two ways. First, did he use cap to lift the beetle? Second, did he lift a beetle that had red cap?

Referential ambiguity is common when using pronouns. For example, Ria met Gauri. She said, 'I am going to the school.' Now exactly who is going to school?

NLP is an ever-evolving field. As of now, it is somewhat ambiguous and suffers from certain limitations as follows.

1. **Lacks precision** Human languages are often ambiguous as humans speak using slang and regional dialect. Moreover, their languages are not precise and may be affected owing to social context. On the contrary, computer languages are very precise and highly structured. This variation makes text interpretation a very challenging task indeed.
2. **Tone of voice and inflection** Semantic analysis is not that easy, especially in three cases. First, when humans speak with sarcasm; second, when the meaning of a sentence changes depending on the word or syllable the speaker puts stress on; third, when the speaker changes

his/her words while speaking. The tone and inflection of speech is also different when speaking in different accents.

3. **Evolving use of language** Natural languages or human languages are continually changing. Although syntax of the language remains the same, they are subject to change over time.

7.11 Evolution of Natural Language Processing

NLP is a multi-disciplinary field that draws concepts from a number of disciplines including computer science and computational linguistics. Extensive work in NLP is being done from the mid-20th century and its evolution can be understood by analysing major milestones given below.

Augmented Transition Networks (ATN) is a finite state machine that is capable of recognizing regular languages.

1. **1950s** Alan Turing developed the Turing Test to determine whether a computer is truly intelligent or not. In this test, automated interpretation and generation of natural language is considered to be the criteria of intelligence.
2. **1950s–1990s** NLP algorithms were usually rules-based. These rules were developed by linguists to guide how computers would process language.
3. **1990s** The top-down approach to NLP shifted to a more statistical approach with computers becoming faster. Now, the new approach stressed on developing rules based on linguistic statistics without the help of a linguist to create all the rules. Data-driven NLP was the highlight of this decade. However, slowly, instead of just catering linguistics, NLP algorithms shifted from a linguistic approach to an engineer-based approach, that could attract a rich variety of scientific disciplines.
4. **2000–2020s** The field of NLP witnessed a dramatic growth with advances in computing power. The number of real-world applications making use of NLP increased manifold. Now, NLP was extensively used in both the business and consumer spheres, including chatbots, cybersecurity, search engines and big data analytics.

7.12 Handling Ambiguities

There are two main components of NLP—NLU and NLG.

1. **Natural language understanding (NLU):** As the name implies, NLU is done by analysing different aspects of the language. This information, in turn, helps the algorithm to map the input given in natural language into useful representations.
2. **Natural language generation (NLG):** As very much evident from the name, in this component, meaningful phrases and sentences in natural language are produced from some internal representation.

No doubt, NLU is more difficult than NLG. This is mainly because of two reasons:

First, natural language has an extremely rich form and structure.

Second, natural language is very ambiguous.

Different levels of ambiguity are encountered such as the following:

1. **Lexical ambiguity** It is the most primitive level of ambiguity and exists at the level of individual words. Lexical ambiguity arises when one or more words in a sentence have multiple meanings. For example, is the word “board” a noun or a verb? Moreover, consider the sentence- “I saw a bat.” Here, bat can be either the wooden material used to hit ball in cricket or a mammal. Similarly, ‘saw’ can be past tense of “see” or the tool used to cut.
2. **Syntax level ambiguity** It occurs when a sentence can be parsed in different ways. For example, if we say, “He threw the beetle with red cap.” Here, it is difficult to say whether he used a red cap to throw away the beetle or did he throw a beetle that had a red cap? Syntactic ambiguity is most commonly occurring ambiguity in English language.
3. **Semantic ambiguity** Even if meaning of each word in a sentence is clear, there may still be multiple ways in which a sentence can be interpreted. For example, consider the sentences- “The dog is chasing the cat.” And “The dog has been domesticated for 10,000 years.” In the first sentence, “The dog” means to a particular dog, whereas in the second sentence, it talks about the species “dog”. Take another example.

“Nita and Mukesh are married.”

This sentence can be interpreted in two ways.

First, Nita and Mukesh are married to each other.

Second, Nita and Mukesh, both are married but not to each other.

4. **Referential ambiguity or Anaphoric ambiguity** Referential ambiguity usually happens when pronouns are used. For example, if we say, ‘Goransh called Aarish. He said, ‘I am coming to India next week.’ Now, who is coming? It is difficult to answer.

Features like autocomplete are trained to predict the next words in a text, based on what has already been typed.

Thus, we can say that, a phrase or word refers to something previously mentioned, but there is more than one possibility while deciding what exactly it is. Moreover, sometimes certain things are assumed. For example, in the sentence given below, ‘they’ actually meant ‘hospital’, but we know it is actually ‘hospital staff’ and not ‘hospital’.

“Neon went to the hospital, and they told him that he needs to take ample rest.” (They = the hospital staff.)

Thus, referential integrity exists when, one sentence can have different meanings and when many sentences actually mean the same thing.

5. **Ellipsis**: This occurs when we omit words that are actually needed for grammatical completion of the syntax. The omitted words are easily understood by the reader/listener. Though this is very common in speech, less in writing. For example, ‘I am going to Rameshwaram. Also to Kanya Kumari.’ This sentence can be understood as ‘I am going to Rameshwaram and Kanya Kumari’.

However, both NLU and NLG need to perform the following:

Text planning to retrieve only relevant content from the knowledge base.

Sentence planning to select words to form meaningful phrases while setting the right tone of the sentence.

Text realization maps the sentence plan into a sentence structure.

7.12.1 Techniques of Ambiguity Resolution

Syntactic constraints can be resolved using the following approaches.

1. *Frequency* This is more important especially for resolving lexical ambiguity. In this approach, the most common meaning of a word is preferred. For example, in the sentence, 'I saw a bird.' Here, 'saw' is taken as past tense of see as it is more common than the other meaning 'to cut with a tool'.

We can also deduce the meaning of an ambiguous word by considering its frequency in the current context. For example, when we are talking about dinner, the word 'pitcher' means 'container of liquid' but when we are talking about baseball, then 'pitcher' means 'thrower of ball.'

2. *Using Associational frequency.* That is, by determining how often two words, or particular meanings of word, appear in text. For example, in the sentence, 'The cheetah and the deer were running very fast. They both could not be seen within a fraction of a second. However, finally it was caught.' In this sentence, 'it' is more likely to be the 'cheetah' because the word 'fast' is usually associated with 'cheetah'.

3. *Selection restrictions or semantic constraints:* This can be applied if we know that concept A can only combine with concept B in mode Z if A or B have specified features. For example, in the sentence, 'The bat ate the flying insects.' We can easily deduce that 'bat' refers to a mammal here because only living objects can eat. So, this 'bat' is not the wooden material to play cricket.

Similarly, the sentence, 'The broken bat is kept there.' Here, the adjective 'broken' usually refers to non-living things. So here, it is the wooden material to play.

4. *Recency rule:* This rule is usually used for resolving anaphora. In this approach, it is believed that in 85% of the cases, the referent is the most recently mentioned object of correct gender and number.
5. *Parallel structure.* Again, this technique is used to resolve issues related to anaphora. For example, 'Neha went with Muskan, and she drove her back.' In this sentence, she is 'Neha' and her is 'Muskan'.
6. **World knowledge** In this technique, the most plausible interpretation is used. For example, 'Cherry got down for lunch, and so did Mary.'

7. *Textual coherence: Here, the most plausible meaning is considered. For example, in the sentence,*

‘Myra said that people should vote for Kiran in the upcoming elections. She said that she is the best amongst all other candidates.’

8. *Speaker intent. In this approach, the text is interpreted in terms of speaker’s intent. For example, ‘You were to be here by 9, do you know what time it is?’. Another example, “Excuse Me! What did you say?” Note that in both the sentences, the speaker is not actually asking. The intent is to make the other person realize something. So, in computing perspective. these ambiguities can be resolved by understanding the context in which they were spoken.*

7.13 The NLP Model of Perception

According to the Oxford dictionary, perception is the ability to see, hear or become aware of something through senses. Humans perceive language not only from text, but from all our senses. For example, we visualize a situation just by listening to the tone of the speaker; we listen to the tone of the voice, we visualize the situation. Similarly, we use senses like touch and smell and even non-verbal communication like body language to interpret language.

Consider the adjective ‘blue’ that has various meanings. For example, ‘blue sea’ means the colour of the sea is blue. When we say, ‘he is feeling blue’, it means that the person is feeling sad. Similarly, when we say that a person’s face turned red, then that means the person was angry or ashamed and does not mean literally red colour. To correctly interpret these words, we need an NLP model that can identify and deal with such complexities.

The current NLP models can understand the context from the text only. This is a serious limitation and to make the NLP models perform at the human level, we need to break these barriers. Bisk et al. 2020 said that perception should be an important dimension in NLP models and it can be derived from the different types of inputs including audio recordings, images and videos. The body language or the voice tone, are currently at

the sideline, of the NLP research. But the creation of these new benchmarks is expected in the next few years within NLP research.

7.13.1 NLP Communication Model

The ‘NLP communication model’ (refer **Fig. 7.15**) is the basis for how human beings perceive reality. In fact, all advances of neuro-semantics were discovered with this model as the foundation as it provides a basic framework for mapping what happens in a person’s mind-body system in between stimulus and response.

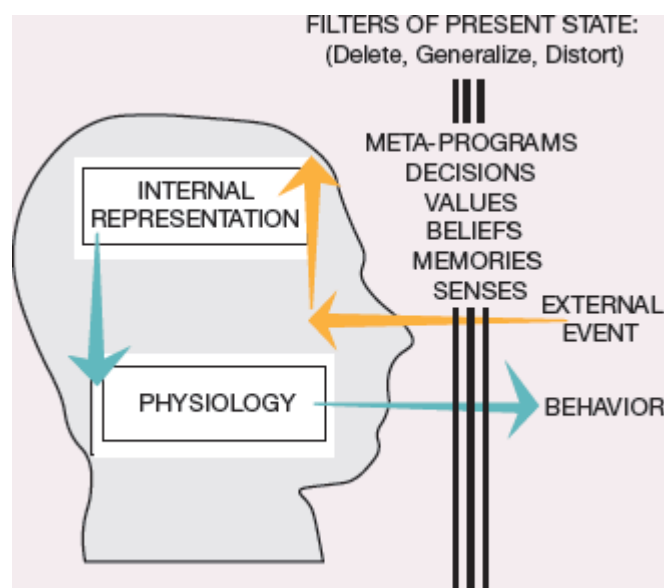


FIGURE 7.15 NLP communication model

From ‘External Event’ to ‘Internal Experience’

The moment we receive a stimulus, our initial *experience* of the ‘external world’ begins. These external events (or stimuli) are converted into neuro-linguistic (mind-body) ‘information’.

The Information is Filtered Through Our State

The information about external world (stimuli) is then filtered our present mind-body state. For example, if someone disturbs you intentionally while doing an important work, then you will respond differently if your work is going well and in a different manner if you are already under stress.

Thus, we think and perceive things differently depending on our state of mind that in turn is affected by the filters including memories, decisions, beliefs, values, identities, thinking patterns, etc.

Internal Representation

After the ‘filtering process’, we get the internal representation of the event/object in the world which is based on our personal perception of it.

State

The internal representation then feeds down into our physiology and combines with present state to form a new present state. Note that this is all done in response to the internal representation of the external event (stimulus).

Thus, the state of our mind and body **govern our behaviour** and we respond to the world by **filtering down the mental map of the world based on our available internal representations**. Therefore, we need to manage our states well that open us up to new responses to perform at our best and to unleash our highest potential.

Perceptual Positions

Perceptual position is a form of modelling in which one steps into somebody else’s shoes, and sees what the other person sees, hears and feels. This helps us to get an impression of the client’s reality.

Generally, when we imagine ourselves as someone else, stand/sit in the other person’s posture, speak with their voice, our mirror neurons get triggered and we actually pick up a lot of information about what how he/she feels thereby improving our interpersonal results.

7.13.2 The Five Perceptual Positions

We can adopt one of the five perceptual positions to fully understand another’s point of view. These positions are as follows:

1st position is where we are ourselves *associated to interpret* things from our own point of view.

2nd position is where we *dissociate* from ourselves and image ourselves in someone else's body to understand things from *their* point of view.

3rd position is where we take the role of an *independent observer* to analyse things from an outside point of view.

4th position is observing the 3rd position in relationship to the 2nd position to understand how neutral we as an observer are.

5th position is where we observe how we relate ourselves to an observer.

Note that 4th and 5th positions occasionally used.

Example: If you have an important meeting coming up with a client, then let us see these five perceptual positions.

1st position is when you imagine a 30 second movie of the meeting from **your** perspective and then describe the meeting in one word.

2nd position is when you imagine yourself **as the client** and then replay the movie to describe the meeting from the client's perspective.

3rd position is when you imagine yourself as **an independent (invisible) observer** and replay the movie to understand the behaviour of the participants and learn how it affects the atmosphere of the meeting. For example, were the participants aggressive, friendly, active/passive or respectful.

In the 4th position, you go back to the 2nd position and play the movie from the **client's perspective** to identify if *anything changed* and what could have been done to improve the meeting. Then you need to incorporate the improvements and then play the movie again in the 1st position to observe what has changed.

A Case Study: Image Analysis and Nlp In Healthcare

In healthcare, a number of medical images are created including X-rays, computational tomography (CT) scans, computed axial tomography (CAT), magnetic resonance imaging (MRI) and modern nuclear imaging modalities such as PET and SPECT.

While image analysis is the process of extracting useful information from digitally stored images using different processing techniques, NLP, on the other hand, applies computational techniques to the analysis and synthesis of natural language and speech. In the healthcare industry, patient records contain a lot of important data like medication, immunization records and lab results. But the problem is that this information is not available as structured text presented in rows and columns and extracting information from it is a difficult task.

NLP techniques are used to process, analyse and interpret clinical text that allow medical professionals to ask important questions regarding an individual's health. NLP can then be used to perform tasks like sentence boundary detection, tokenization, problem-specific segmentations, spelling or grammatical error identification and recovery, word sense disambiguation, information extraction, multi-atlas segmentation, fuzzy clustering, graph cuts, genetic algorithms, support vector machines, random forests and deep learning to improve image analysis. Then, speech-to-speech translation engines, social-media mining about health, sentiment analysis toward products and services are added advantages.

However, some of the challenges in this field are as follows:

- Need for generic image analysis techniques that can be efficiently adapted for a specific clinical task.
- Algorithms for analysing heterogeneous image data.
- Models from medical images with less user interaction.
- To understand the language's context and reasoning just like humans.
- Acquisition of the data for image analysis and NLP has lot of ethical issues. Patients have the right to decide how and when their data is used. However, the more the data is available for research, the more advances can be made in the field of image analysis and NLP.

There is a need to develop a diagnostic system that can efficiently read pathology reports by analysing images and the textual annotations that come on the reports, to generate a textual report. For example, the system can analyse the bone scan image and generate a report on the probability of how well treatment of cancer in case of bone cancer. By analysing the text from other clinical inputs such as lab reports and clinical consultation notes in addition to the bone scan image, the system can recommend further treatment for the patient.

The system can, in fact, analyse images from a large number of patients and stratify the population based on some clinical criteria, to help policy decision makers formulate strategies based on the results obtained by analysing reports.

Case Study: How to Automatically Generate Textual Descriptions for Photographs with Deep Learning

To give a caption to an image, we need to generate a human readable textual description. Though this task is easy for a human, it is not trivial for a machine as it involves application of computer vision and NLP techniques.

To describe an image (refer **Fig. 7.16**), the content of the image needs to be understood and translated to meaning in the terms of words. Let us see the tasks involved in doing this activity.

Classify Image

The computer assigns a class label to an image from one of many known classes (as shown in **Fig. 7.17**).

Describe Image

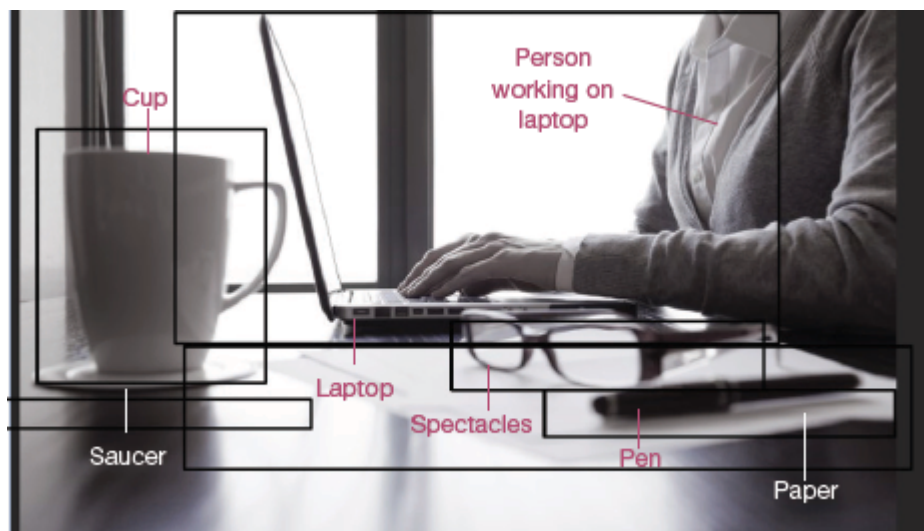
Generate a textual description of the contents image.

Annotate Image

Generate textual descriptions for specific regions on the image.



FIGURE 7.16 Images with Textual Description



Credit: KieferPix / Shutterstock

FIGURE 7.17 Classify Image

Neural Captioning Model

Neural network models are extensively used to automatically generate captions for images. This is done using feature extraction and language model (given in **Fig. 7.18**), that are discussed below.

Feature Extraction Model

The feature extraction model is a neural network-based approach that will extract the salient features from an image, which is usually a fixed-length vector. The resultant vector gives an internal representation of the image which cannot be directly interpreted by the end-users. For image analysis, deep convolutional neural network (CNN) model is used that is already trained on the images stored in the image captioning dataset.

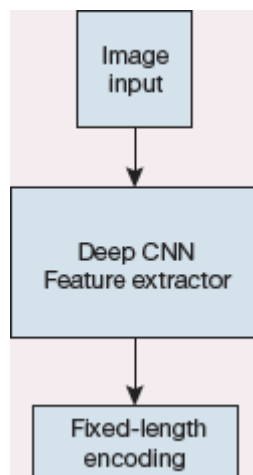


FIGURE 7.18 Feature Extraction Model

Language Model

A language model analyses the words in a sequence to predict the probability of the next word in that sequence. For the task of giving captions to an image, the language model uses the extracted features from the neural network to predict what words should appear in the description of the image. For prediction, we usually use a recurrent neural network, such as a long short-term memory network, or LSTM, as the language model. Each output generates a new word in the sequence which is then encoded and passed to the decoder as input for generating the subsequent word.

An improved form of the model gathers the probability distribution of words across the vocabulary to form the output sequence. The output sequence is then searched to generate multiple possible descriptions. These descriptions are finally rated by likelihood. For quick and accurate results, the language model is trained using pre-computed features extracted from the image dataset.

Note that a separate encoder-decoder is required for translation in a different language. So, the architecture that worked for English will not work for French.

Thus, the entire encoder-decoder architecture model is *based on a CNN that encodes an image into a compact representation, followed by a recurrent neural network (usually a stack of LSTM layers as shown in **Fig. 7.19**) that acts as a decoder and generates the corresponding sentence. The model is trained to maximize the likelihood of the sentence given the image.*

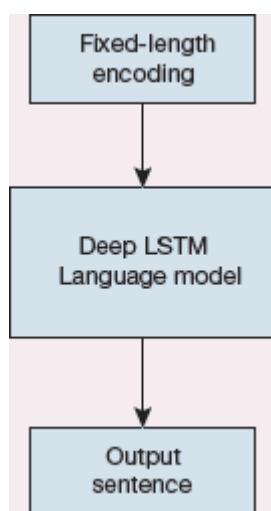


FIGURE 7.19 CNN with LSTM layers

Recent advances in image analysis, NLP and caption generation have made it possible to identify salient parts of an image while generating its caption.

7.14 Constituency Grammar

In constituency grammar, a sentence can have basic constituents as any word or a group of words (also called phrase). These constituents are organized in a sentence based on their properties. These properties can be studied either by using part of speech tagging (PoS) or by using a noun or verb phrase identification.

We know that grammar consists of certain rules. These rules have a single symbol on the left-hand side of the rewrite rules. For example, let us create grammar to parse a sentence –

‘The girl sang a song’

Articles (DET) – a | an | the

Noun Phrase (NP) – Article + Noun | Article + Adjective + Noun

= DET N | DET ADJ N

Verb Phrase (VP) – NP V | V NP

Now, we will create a parse tree that will break down the sentence into its structural parts. This will help the machine to easily understand and process the information. To help the parsing algorithm to construct this parse tree, a set of rewrite rules is used to describe what structures are legal in that language.

These rules dictate how a certain symbol in the tree can be expanded by a sequence of other symbols. Given below are some rules.

S → NP VP

NP → DET N | DET ADJ N

VP → V NP

For example, as per the first rule, a sentence (S) can be formed using a noun phrase (NP) and a verb phrase (VP) (given in **Fig. 7.20**).

The second rules say that a noun phrase is a group of words that either begins with a determiner (DET) and is followed by a noun (N) or begins with a determiner (DET), followed by an adjective (ADJ) and then a noun (N).

Similarly, the third rule states that a verb phrase (VP) is formed by combining a verb (V) and a noun phrase (NP).

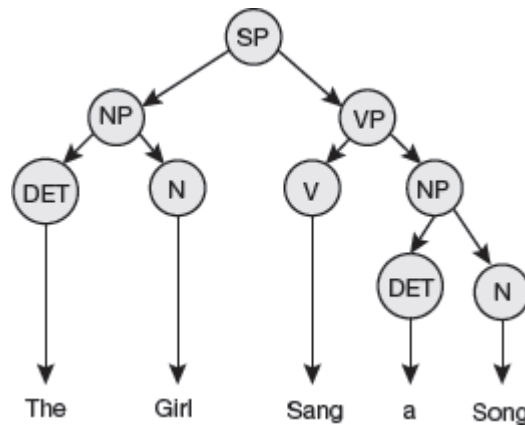


FIGURE 7.20 Sentence with Noun Phrase followed by a Verb Phrase

According to the sentence, lexicon can be formed as follows:

Nouns – girl | girls | song | songs

Verbs – sing | sang | singing

Adjectives (ADJ) – beautiful | patriotic | sad

Using the above grammar and lexicon, we can create the parse tree as given in **Fig. 7.20**. However, considering the above rewrite rules, V can be replaced by both, sing or sang. So, any sentence like ‘The bird sing a song’ will also be accepted even though it is not correct.

Merits

1. It contains the most popularly used grammar.
2. It is very simple.

Demerits

1. This grammar is not very accurate. As we have seen, the sentence, ‘The girl sing a song’ is syntactically correct according to parser, but is otherwise incorrect.
2. To enhance accuracy, multiple sets of grammar are required. While one set of rules is used to parse singular nouns/verbs, another set can parse plural forms. Similarly, a completely different set of rules can be used to parse passive sentences, and another ruleset to parse active

sentences, so and so forth. However, this can result in a huge set of rules that after a point in time becomes unmanageable.

Example: Observe the parse tree given in **Fig. 7.21** and notice how the sentence is formed.

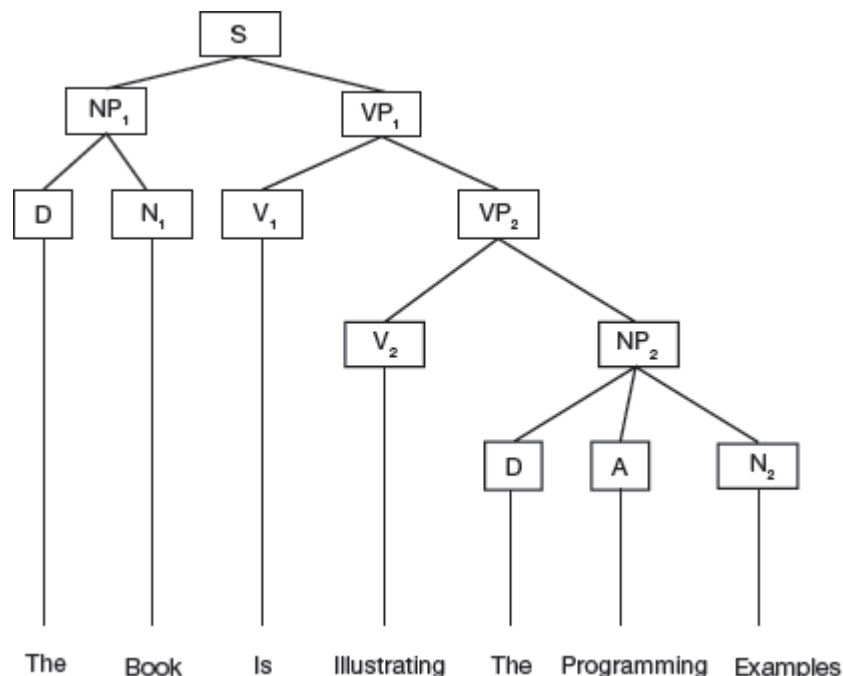


FIGURE 7.21 Parse Tree for the given sentence

7.15 Context-Free Grammar

Context-free grammar is a formal grammar that can be used to generate all possible strings in a given formal language. CFGs were developed by Noam Chomsky in the mid-1950s to help users specify rules that state how a constituent can be segmented into smaller and smaller constituents, up to the level of individual words. Its simple yet powerful formalism could describe the syntactic structure of natural languages.

Mathematically, a context-free grammar G can be defined using a 4-tuple (N, T, S, P) value, where

N or V_N is the set of non-terminal symbols, or variables.

T or Σ is the set of terminal symbols.

S refers to the Start symbol and naturally, $S \in N$

P is the set of Production rules that dictates how T and V can be combined. Every production rule consists of non-terminals, arrow and terminals.

The left side of production is non-terminals while the right side is called terminals. Such a rule is of the form $\alpha \rightarrow \beta$, where α and β are strings on $N \cup \Sigma$ and at least one symbol of α belongs to V_N .

Technically, CFG is a superset of regular grammar as shown in **Fig. 7.22**.

7.15.1 Derivations

A derivation is a sequence of rule applications that derive a terminal string $w = w_1 \dots w_n$ from S. This means that, in CFG, the start symbol derives the string by repeatedly replacing a non-terminal by the right-hand side of the production rule, until all non-terminals have been replaced by terminal symbols (refer **Fig. 7.23**).

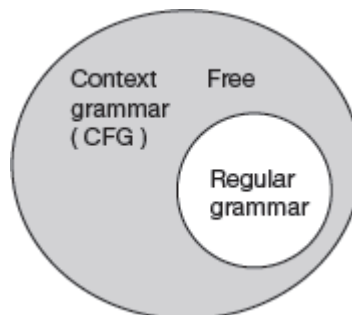


FIGURE 7.22 Relationship between CFG and Regular Grammar

Image Source: Google Images

S	Rule Applied
aS	Rule 1
aaS	Rule 1
aaaS	Rule 1
aaaaS	Rule 1
aaaaaS	Rule 1
aaaaaaS	Rule 1
aaaaaa	Rule 2

FIGURE 7.23 Deriving String from Start Symbol

Example: Given below are some production rules, derive the string “abbcbba” using the CFG.

Rule 1: $S \rightarrow aSa$

Rule 2: $S \rightarrow bSb$

Rule 3: $S \rightarrow c$

Step 1: Let's start with rule 1, $S \Rightarrow aSa$

Step 2: Using Rule 2, we can get the string $S \Rightarrow abSba$

Step 3: Again, using Rule 2, we can get the string $S \Rightarrow abbSbba$

Step 4: Now, using Rule 3, string $S \Rightarrow abbcbbba$

Example: Construct CFG for a language that permits any number of as.

Solution: We can write regular expression of such a language as, **Regular Expression= a^***

Let us now list the production rule for our regular expression.

Rule 1: $S \rightarrow aS$

Rule 2: $S \rightarrow \epsilon$

To derive a string “aaaaaa”, we will start with the start symbol and proceed as given in Table.

Thus, the regular expression, a^* is capable of generating a set of strings $\{\epsilon, a, aa, aaa, \dots\}$

Here, we have written, the null string (ϵ) because S is a start symbol and by applying Rule 2 we can straight forwardly get null string.

7.15.2 Using Nominals in CFG

Nominals contain the head and any pre- and post- modifiers of the head.

Pre-quantifiers include cardinals and ordinals (like four desks), adjectives (e.g., big data) and ordering constraints (like seven small dwarves).

Postmodifiers are of three types:

Prepositional phrases like tea from India.

Non-finite clauses like trains departing after noon.

Relative clauses like restaurants that serve Jain food.

Some general (recursive) rules to handle nominals are,

Nominal $\rightarrow$ Nominal PP

Nominal $\rightarrow$ Nominal GerundVP

Nominal $\rightarrow$ Nominal RelClause

We can even refine noun phrase (NP) as follows:

NP $\rightarrow$ DET NOM; for example, a flight

NP $\rightarrow$ Proper Noun; for example, Chennai

NOM → NOUN | NOM NOUN; for example, flight | morning flight

NP → Pronoun; for example, I

Points to remember:

1. PP stands for prepositional phrase and is equivalent to proposition NP.
For example, from Hyderabad.
2. VP stands for verb phrase and is equivalent to
VP → Verb; for example, want.
3. VP → Verb NP; for example, wants a chocolate
4. VP → Verb NP PP; for example, want a chocolate from Switzerland
5. VP → Verb PP; for example, arriving on Monday

Example: Use CFG for parsing the sentence, “I prefer a morning flight”.

Step 1: Let's start with S

Step 2: Expand S as NP VP

Step 3: Replace NP with Pro. So, now we have, PRO VP

Step 4: Replace Pro with I. We get, I VP

Step 5: Replace VP with Verb NP. Thus, it is now, I V NP

Step 6: Replace prefer with V to get, I prefer NP **Step 7:** Replace NP with Det Nom. So, I prefer DET NOM

Step 8: Replace DET with ‘a’. Therefore, we are left with, I prefer a NOM

Step 9: Expand NOM by replacing it with NOM NOUN. It becomes I prefer a NOM NOUN.

Step 10: Replace NOM with NOUN. That make the expression, I prefer a NOUN NOUN

Step 11: Replace morning with first NOUN. We get, I prefer a morning NOUN.

Step 12: Finally, we get, I prefer a morning flight.

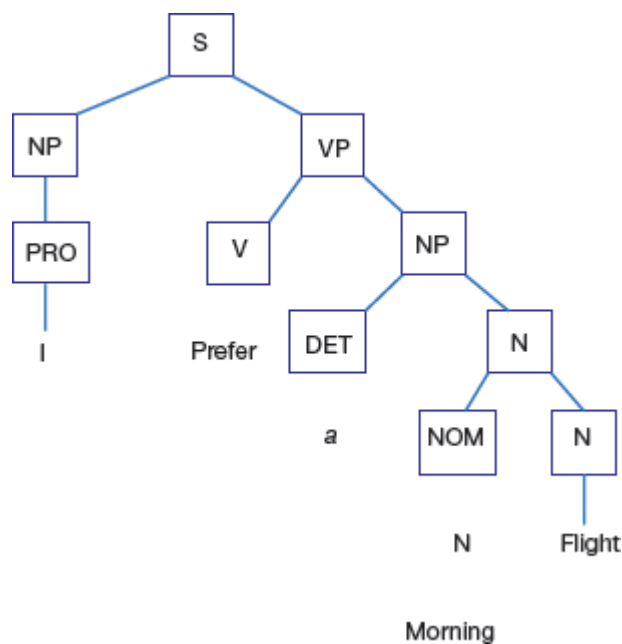


FIGURE 7.24 Parse Tree for “I prefer a morning flight”

Example: Create a parse tree for the sentence, “All the late-evening buses from Delhi to Rhishikesh leave at midnight”

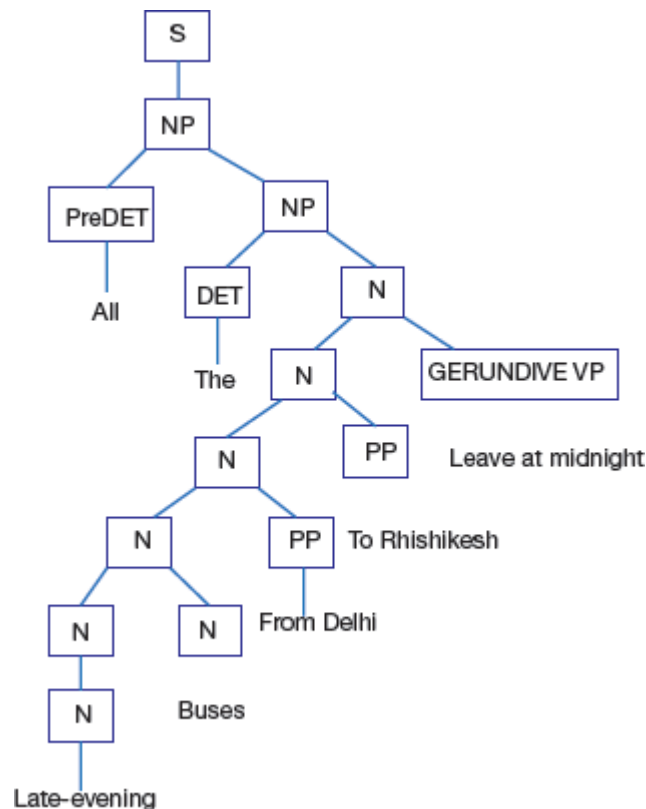


FIGURE 7.25 Parse Tree for “All the late-evening buses from Delhi to Rhishikesh leave at midnight”

7.15.3 Uses of CFG

CFGs can be used in the following scenarios:

1. Describe programming languages.
2. Automatically construct an efficient parser for a properly designed grammar.
3. With the help of information about associativity and precedence, suitable grammars can be constructed for expressions.
4. Describe nested structures like balanced parentheses by matching the starting and closing brackets, checking if-then-elses, etc.

7.15.4 Types of Derivation

We have seen that derivation specifies a sequence of production rules. During parsing, two decisions have to be taken:

First, which non-terminal has to be replaced.

Second, the production rule by which the non-terminal will be replaced.

To decide which non-terminal has to be replaced with production rule, we have two strategies.

1. **Left-most Derivation:** In this derivation, the input string is read starting from the left. At each step, the input scanned is replaced with the chosen production rule.

Example: Given the input string $a + b * c$ and production rules as given below, demonstrate left-most derivation.

Rule 1: $S = S * S$

Rule 2: $S = S + S$

Rule 3: $S = a \mid b \mid c$

Step 1: Apply Rule 1, to get, $S = S * S$

Step 2: Apply Rule 2 on left S after '=', to get, $S = S + S * S$

Step 3: Apply Rule 3, to replace leftmost S after '=' with 'a', to get, $S = a + S * S$

Step 4: Apply Rule 3, to replace leftmost S after '=' with 'b', to get, $S = a + b * S$

Step 5: Apply Rule 3, to replace leftmost S after '=' with 'c', to get, $S = a + b * c$

2. **Right-most Derivation:** This type of derivation is just the opposite of left-most derivation. Here, the input is scanned from right to left and replaced with an appropriate production rule.

Example: Given the input string $a + b * c$ and production rules as given below, demonstrate right-most derivation.

Rule 1: $S = S * S$

Rule 2: $S = S + S$

Rule 3: $S = a \mid b \mid c$

Step 1: Apply Rule 2, to get, $S = S + S$

Step 2: Apply Rule 1 on rightmost S after '=', to get, $S = S + S * S$

Step 3: Apply Rule 3, to replace rightmost S after '=' with 'c', to get, $S = S + S * c$

Step 4: Apply Rule 3, to replace rightmost S after '=' with 'b', to get, $S = S + b * c$

Step 5: Apply Rule 3, to replace rightmost S after '=' with 'a', to get, $S = a + b * c$

7.15.5 Parse Tree

We have seen that parse tree is a graphical representation of terminal as well as non-terminal symbols. While leaf nodes represent terminal nodes, interior nodes, on the other hand, are the non-terminal nodes. We can derive an entire string by parsing the tree starting from the root node. This root node contains the start symbol.

It is also interesting to note that parse tree follows the precedence of operators. The deepest sub-tree is traversed first. Therefore, an operator in the sub-tree has more precedence than an operator in the parent-tree.

Usually, the tree is parsed using in-order traversal method. In this method, first left-subtree is parsed, followed by the root node which is then followed by traversal of the right-subtree.

Example: Consider the production rules given below to generate the parse tree for the expression $a + b - c$

Rule 1: $S = S + S \mid S * S$

Rule 2: $S = a \mid b \mid c$

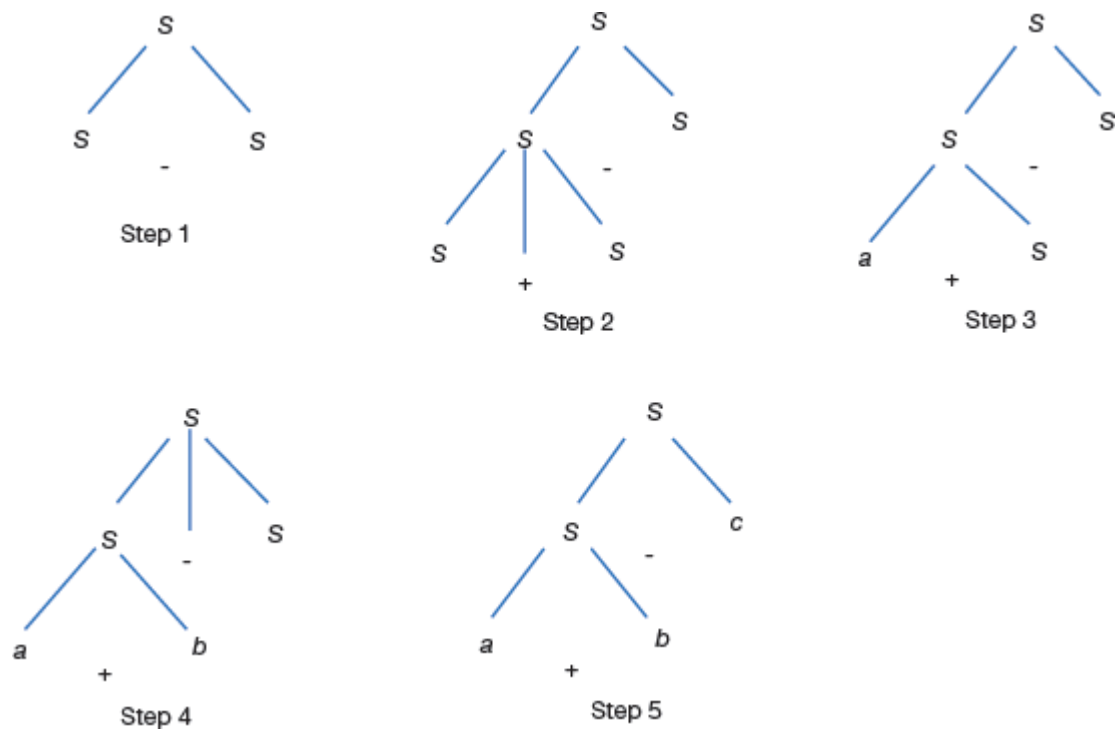


FIGURE 7.26 Parse Tree for expression, $a + b - c$

7.15.6 Ambiguity in CFG

A grammar is said to be ambiguous if there exists more than one leftmost derivation or more than one rightmost derivative or more than one parse tree for a given input string. Correspondingly, a grammar that is not ambiguous is known as unambiguous.

Example: Given the input string, $aabb$ and production rules given below. Prove that this grammar is ambiguous.

Rule 1: $S = aSb \mid SS$

Rule 2: $S = \epsilon$

For the string $aabb$, we can generate more than one parse tree. Three of them are given below in **Fig. 7.27**.

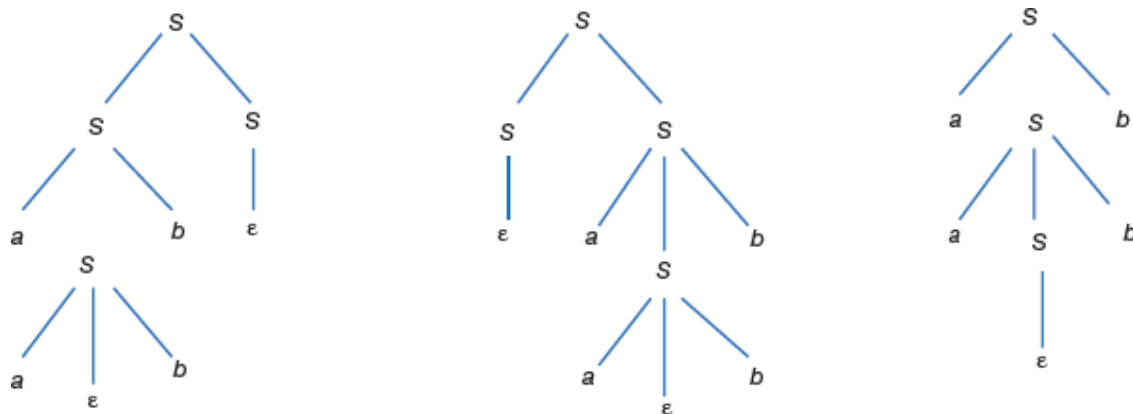


FIGURE 7.27 Parse Tree for the given grammar

Hence, this grammar is ambiguous and thus not good for a compiler construction. To remove this ambiguity, we need to re-write the grammar. There is no automatic tool to detect and remove such ambiguity.

7.15.7 Parser

Parser is a compiler that breaks the data coming from lexical analysis phase into smaller elements.

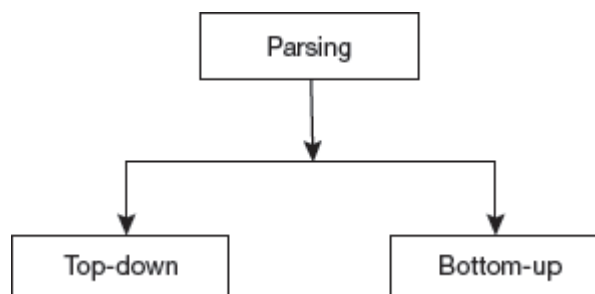


FIGURE 7.28 Techniques for Generating Parse Trees

As input, the parser accepts a sequence of tokens to generate output in the form of a parse tree. A parser can produce the parse tree either by using top-down parsing or bottom-up parsing technique (refer **Fig. 7.28**).

Top-down parsing, also known as recursive parsing or predictive parsing, starts the parsing process from the start symbol and transforms it into the input symbol. Bottom-up parsing, on the other hand, constructs the parse tree for an input string. It starts with the input symbol and constructs the parse tree up to the start symbol. For this, rightmost derivation is used to

trace the string in reverse. Bottom-up parsing is also known as shift-reduce parsing.

Example: Consider the Parse Tree representation of input string “abcd” using top-down approach.

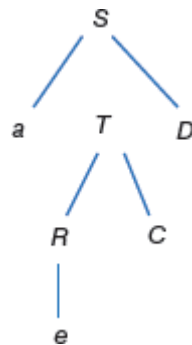


FIGURE 7.29 Parse Tree for “abcd” using top-down approach.

Example: Using the production rules given below, construct a parse tree using bottom-up parsing technique.

Rule 1: $E \rightarrow T$

Rule 2: $T \rightarrow T + F$

Rule 3: $T \rightarrow a$

Rule 4: $F \rightarrow T$

Rule 5: $F \rightarrow a$

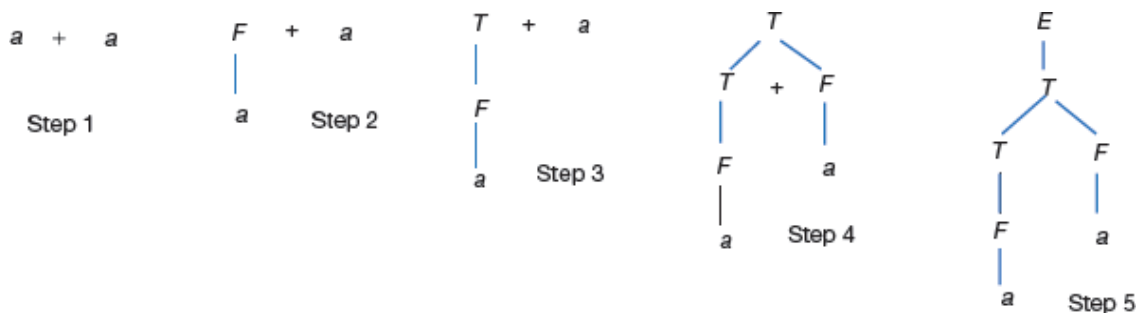


FIGURE 7.30 Parse tree using bottom-up parsing

7.16 Speech Recognition

Speech recognition, or speech-to-text, is the ability of a machine to identify words spoken aloud to convert them into readable text. Speech recognition software usually have a limited vocabulary and may identify words only when they are spoken loud and clear. A sophisticated can process words spoken in different accents and languages.

Speech recognition includes concepts from several disciplines including computer science and linguistics. These days, many modern devices that work on text-focused programs are being equipped with speech recognition functions to facilitate easier or hands-free use of the device.

Speech recognition is often thought of as voice recognition. But they are two different technologies. **Table 7.3** differentiates the two terms.

TABLE 7.3 Differences between Speech recognition and Voice recognition

Speech Recognition	Voice Recognition
• Identifies words in spoken language. • Speech recognition translates anyone’s voice. • Accuracy rate is low.	• It is a biometric technology that identifies an individual's voice. • Voice recognition understands a specific user’s voice. • Exhibits higher accuracy.

Note that a voice recognition system depends on a recorded template of a user’s voice, and is called ‘template matching.’ The software is trained to recognize a user’s voice. To understand how it works, consider the steps given below.

Step 1: The program will show printed word(s) spoken by the user and repeats several times into the system’s microphone to train the voice recognition software.

Step 2: Statistical average of multiple samples of the same word(s) are calculated.

Step 3: The average samples as a template in its data structure.

7.16.1 How Does Speech Recognition Work?

Speech recognition systems process spoken words using computer algorithms to accurately interpret them so that they can be converted into text. For example, when a software program converts the sound recorded by a microphone into written text that is readable and understandable by humans, the following steps are performed.

Step 1: Analyse the audio.

Step 2: Break the audio into smaller parts.

Step 3: Digitize the individual parts into a computer-readable format.

Step 4: Apply a suitable algorithm to match to convert into most accurate text representation.

Speaker-dependent voice recognition devices can provide personalized responses to a user.

A sophisticated speech recognition software is designed to handle the highly variable and context-specific nature of human speech. The software algorithms are trained on different speech patterns, speaking styles, languages, dialects, accents and phrasings so that it can accurately convert audio into text. The software also delineates background noise that often accompanies the audio. To meet these requirements, a speech recognition software is developed using two types of models:

1. Acoustic models to represent the relationship between linguistic units of speech and audio signals.
2. Language models to match sounds with word sequences to differentiate between words that sound similar.

7.16.2 Applications of Speech Recognition

Some of the popular applications of speech recognition systems include the following.

1. **Mobile devices.** These devices use voice commands for call routing, speech-to-text conversion, voice dialing and voice search. Now, users can respond to a text message without even looking at the devices. These days, speech recognition has become so popular that even PDF document readers and word processing applications like Microsoft Word have speech recognition as an in-built capability to facilitate users to dictate words that can be converted into text and vice versa.
2. **Virtual assistants** use speech recognition to communicate with users and perform different tasks by giving voice commands (as shown in **figure 7.31**).
3. Users can use speech recognition software to learn the correct pronunciation of a given set of words in a particular language.

Ten tasks for a virtual assistant

- Add events to a calendar
- Add items to to-do lists
- Control smart home devices
- Make and receive phone calls
- Create text messages
- Get directions
- Hear news and weather reports
- Find hotels or restaurants
- Check flight status
- Request songs



Credit: fizkes. Shutterstock

FIGURE 7.31 Ten tasks for a virtual assistant

4. Speech recognition software is used to provide 24 X 7 customer service using automated voice assistants that listen to the customer's queries to provide relevant information.
5. Doctors use real-time speech recognition software to transcribe notes that can be stored in the healthcare records. Similarly, courtroom reporting is done by such software rather than by human transcribers.
6. Speech recognition software is used by people with hearing disability to translate spoken words into text so that they can easily understand what others are saying. The software is also a boon for those who can use their hands in a very limited way to use computers by giving voice commands.
7. Speech recognition, when paired with sentiment analysis, can be used to identify the emotion(s) of the speaker about a product or service.

8. Speech recognition software is used for hands-free communication by drivers who can control phones, radios and global positioning systems without even touching them. This gives drivers a better control on their vehicle thereby reducing the risk of accidents.

5 AI technologies driving business value

From image and speech recognition systems to sentiment analysis, AI technologies in business keep adding use cases.

Here are five AI subfields and the ways in which they are being used separately and in combination by businesses.

Image recognition	Speech recognition	Chatbots and ChatOps	Natural language generation	Sentiment analysis
- Identify products on shelves - Identify people in a picture or video - Identify defects on an assembly line - Generate damage estimates in insurance - Detect customers entering a store - Count crowds at large public events - Generate models of the real world - Identify street objects for self-driving cars - Monitor for social distancing	- Record conference calls and physical meetings - Monitor call center interactions between agents and customers - Language translation for travelers - Hands-free commands for home and mobile devices and vehicles - Dictate medical reports - Train air traffic controllers - Support video game interactions - Automate closed captioning for indexing video	- Automate customer interactions - Represent the company brand on social media - Document communications within and across departments - Track key performance indicators - Automate commonly asked HR questions - Handle and triage IT help desk requests	- Generate customized product descriptions based on user interests, expertise, native language - Generate recurring content, such as earnings reports - Generate the text for what is likely to come next in an email - Generate explanations of graphs and metrics found in analytics reports	- Analyze how a product or service change affects customers - Identify and form relationships with “brand influencers” - Gauge employee morale by analyzing internal postings - Discover important trends by analyzing customer responses - Identify specific causes for brand decline, such as long wait times - Identify emotion conveyed in voices and faces

FIGURE 7.32 5 AI technologies driving business value

Besides these applications, speech recognition software is also used for the following:

Giving special attention to certain words, such as those spoken frequently or that are prominent in a conversation.

A speech recognition software with acoustic training can be used to differentiate between speaking style, pace and volume of people speaking in an office to understand the contributions of individuals in a conversation.

The software can also be used for filtering profanity that filters out undesirable words and language. Other applications of NLP and speech recog-

dition in the industry are given in **Fig. 7.32**

7.16.3 Algorithms for Speech Recognition

Speech recognition has been possible due to some powerful algorithms listed below.

Natural language processing eases and accelerates the speech recognition process.

1. **Hidden Markov model:** This is used in autonomous systems in two cases. First, the state is partially observable. Second, when the entire information required for decision-making is not immediately available to the sensor or the microphone for speech recognition.
2. **N-grams.** This technique creates a probability distribution for a sequence. Such an algorithm looks at the last few words spoken, approximates the history of the sample of speech to determine the probability of the next word that will be spoken.

Artificial intelligence, NLP and machine learning/deep learning techniques are used in advanced speech recognition software. These systems learn the grammar, structure, syntax and composition of audio and voice signals to process speech. These systems learn with each use to give better results the next time.

7.16.4 Advantages of Speech Recognition

1. Speech recognition facilitates machine-to-human communication in natural language.
2. The technology can be easily used in computers and mobile devices, making it readily accessible.
3. Even a novice user can easily use speech recognition feature that is straightforward to operate and often runs in the background.
4. Speech recognition systems that use AI techniques automatically improve with every use as more and more data is generated. So, these

systems evolve continuously to become more effective and easier to use over time.

7.16.5 Disadvantages of Speech Recognition

1. Speech recognition systems often give inconsistent performance as they may not always capture words accurately possibly due to difference in pronunciation, lack of support for a particular language and background noise. Although acoustic training does help, it is however not perfect.
2. These systems take time to deploy and master and are relatively slow.

The success of speech recognition system also depends on the recording equipment used and not just the software alone.



PROGRAMMER'S ZONE

Program for sentence tokenization on a review written for the Jruma app.

```
import nltk
text = "A very great App for students! I wish to solve more and
more quizzes every day and increase my knowledge! Thank you so
much for this App! This App should be appreciated more and more!
Congratulations!!:))"
sentences = nltk.sent_tokenize(text)
for sentence in sentences:
    print(sentence)
```

OUTPUT

Write a script to tokenize words

OUTPUT

Program to demonstrate stemming text.

OUTPUT

Remove stop words from our text

OUTPUT

Program to tag text using the `pos_tag()` method of `nltk` library.

OUTPUT

Write a code to compare the working of a lemmatizer and a stemmer.

OUTPUT

Write a code to print the two most frequently occurring words in a text.

OUTPUT

Write a code to print words in a text that have more than 7 characters.

OUTPUT

Create bigrams using a piece of quotation.

OUTPUT

Print collocations from the given text.

Hint: A collocation is a sequence of words that occur together unusually often

OUTPUT

Write a program that reads a piece of text and then returns how frequent the different lengths of word are.

OUTPUT

Example: Implementing Genetic Algorithm

OUTPUT



KEY TERMS

Natural language processing: The manipulation or understanding of text or speech by a computer software.

Chatbots: NLP, AI-based systems that are specifically designed to interact with humans through text or speech.

Lexical analysis: The process of identifying and analysing structure of words by breaking down a big chunk of text into elementary words.

Syntactic analysis: The process of parsing a sentence to analyse words in the sentence to interpret its meaning as per the grammatical rules of the language.

Parts of speech (POS) tagging: The process that marks words in text format for a particular part of a speech depending on its definition and context.

Syntax: Arrangement of words in a sentence to make grammatical sense.

Perception: The ability to see, hear or become aware of something through senses.

Context-free grammar: A formal grammar that can be used to generate all possible strings in a given formal language.

Derivation: A sequence of rule applications that derive a terminal string $w = w_1 \dots w_n$ from S .

Parse tree: A graphical representation of terminal as well as non-terminal symbols.

Parser: A compiler that breaks the data coming from lexical analysis phase into smaller elements.

Speech recognition: The ability of a machine to identify words spoken aloud to convert them into readable text.



RECAPITULATION POINTS

- NLP can be used to create intelligent systems for speech recognition, machine translation, question answering, spam detection, document summarization, predictive typing, named entity recognition and auto-complete, to name a few.

- While writing using human language, we often misspell or abbreviate words, or omit punctuation. Similarly, while speaking, we pronounce different words differently (in regional accents). Some people mumble, stutter and borrow terms from other languages which further increases the complexity of processing natural languages.
- Techniques like deep, supervised and unsupervised learning are widely used for modelling human language to resolve ambiguity in language.
- NLP systems break down text/speech in human language into shorter, elemental pieces, to understand relationships that exist between the pieces for interpreting its meaning.
- Speech recognition is used for converting spoken words into text.
- Topic classification uses NLP to organize unstructured text into categories.
- Natural language understanding (NLU) deals with mapping the given input in natural language into useful representations.
- Natural language generation (NLG) uses content in internal representation to produce meaningful sentences in a natural language.
- A chatbot works on request-response model.
- The lexicon of a language refers to the collection of words and phrases in that language.
- Discourse integration studies the meaning of a sentence with respect to the meaning of sentence preceding as well as succeeding it in sequence.
- In word sense disambiguation step, the meaning of a word is analysed based on the context in which it is used.
- Named entity recognition identifies words that can be classified in different groups.
- Chatbots provide automated answers to common customer queries. They use pattern recognition systems with heuristic responses to interact with users.
- Lexical ambiguity is the most primitive level of ambiguity that exists at the level of individual words. It arises when one or more words in a sentence has multiple meanings.
- *The recency rule believes that in 85% of the cases, the referent is the most recently mentioned object of correct gender and number.*

- A context-free grammar G can be defined using a 4-tuple (N, T, S, P) value, where, N is the set of non-terminal symbols or variables, T is the set of terminal symbols, S refers to the Start symbol and P is the set of Production rules. Every production rule consists of non-terminals, arrow and terminals.
- In a production rule, the left side of production is non-terminals while the right side is called terminals.
- *A grammar is said to be ambiguous if there exists more than one left-most derivation or more than one rightmost derivative or more than one parse tree for a given input string.*
- N-grams is an algorithm for speech recognition that creates a probability distribution for a sequence. It looks at the last few words spoken to determine the probability of the next word that will be spoken.



REVIEW QUESTIONS

1. What do you understand by the term natural language processing?
2. Give the applications of NLP.
3. Discuss some issues faced while dealing with human language.
4. How are NLP systems useful?
5. Give any three real-world applications of NLP.
6. Discuss three ambiguities faced by NLP systems.
7. Discuss the steps involved to perform NLP.
8. What is a chatbot? Give any three uses of a chatbot.
9. How does a chatbot work?
10. Enumerate the different types of chatbots.
11. What steps will you follow to create a chatbot?
12. Can you identify which is stemming and which one of the two is lemmatization?

13. Differentiate between the following.
 1. Natural language and computer language.
 2. NLU and NLG
 3. Open and closed chatbot
 4. Lemmatization and stemming
 5. Syntax versus semantic analysis
 6. Leftmost derivation and rightmost derivation
 7. Top-down and bottom-up parsing
 8. Speech and voice recognition
14. Summarize the five steps involved in any NLP system.
15. Discuss any five applications of NLP systems.
16. Highlight the problems faced when implementing NLP systems.
17. Give some techniques to resolve ambiguity.
18. What is ambiguous grammar? Draw a parse tree for such a grammar.
19. How does speech recognition work?
20. Write short notes on:
 1. Natural language processing
 2. Context-free grammar
 3. Genetic algorithms
21. Draw the tree that represents the sentence, 'This movie is revealing the exact incidence'.
22. Consider the structure given below that describes a bucket. Write some rules that you can derive from this structure.

23. Why is the sentence, 'Mary ate a salad with spinach from California for lunch on Tuesday.', syntactically ambiguous?
24. Create grammar to parse the sentence,
 1. 'The bird pecks the grains'
 2. 'The dog saw a man in the house'
25. Why is the sentence, 'John kissed his wife, and so did Sam.' semantically ambiguous?
26. Find out, who is being referred to by pronouns.
 1. 'Margaret invited Susan for a visit, but she told her she had to go to work'.
 2. 'Margaret invited Susan for a visit, and she gave her a good lunch.'
 3. 'On the train to Boston, George chatted with another passenger. The man turned out to be a professional hockey player.'
 4. 'Bill told Amy that he had decided to spend a year in Italy to study art.' 'That would be his life's work.'
27. Why is the sentence, 'Kiran went to Sunita. She said, 'I am hungry.' said to be an example of referential ambiguity?
28. 'I saw the girl with binoculars', is syntactically ambiguous. Comment.
29. 'Elsie tried to reach her aunt on the phone, but she didn't answer.' is lexically as well as syntactically ambiguous. Why? Is there any other ambiguity also involved?
30. Identify the issue- (ellipsis, associational frequency, semantic constraints):
 1. 'Mozart was born in Salzburg and Beethoven, in Bonn'.
 2. Identify the resolution 'I hammer the nail and I saw the board'.
 3. 'The sick bat lay on the ground.'

4. 'The clock is fast.' vs 'The horse is fast.' vs. 'The clothes are fast.' vs. 'The slopes are fast.' vs. 'The knot is fast.'
5. 'Manya is looking for a **match**.' is lexically ambiguous. Comment.



FILL UP

1. Google uses _____ to detect and filter spam emails.
2. In NLP, text is broken down into smaller units called _____.
3. _____ identifies words that can be classified in different groups.
4. In _____, algorithms are used to analyse unstructured data and automatically generate content based on that data.
5. _____ provide automated answers to common customer queries.
6. _____ ambiguity usually happens when pronouns are used.
7. _____ occurs when we omit words that are actually needed for grammatical completion of the syntax.
8. _____ uses the most plausible interpretation is used.
9. _____ is a superset of regular grammar.
10. _____ is a compiler that breaks the data coming from lexical analysis phase into smaller elements.
11. NLP applies machine learning algorithms to _____ and _____.
12. _____ uses NLP to detect and filter spam emails.
13. _____ language is ambiguous in nature.
14. _____ analysis is used to identify user's opinion within large amounts of text.
15. Speech _____ is used for converting spoken words into text.
16. _____ are NLP-based systems that are specifically designed to interact with humans through text or speech.
17. _____ of a language refers to the collection of words in a language.
18. _____ analysis is done to analyse words in the sentence to interpret its meaning as per the grammatical rules of the language.
19. _____ reduces words to their base form.
20. Words like *I, they, have, like, yours* are known as _____ words.
21. _____ refers to having any sentence depends upon the meaning of the sentence just preceding it.
22. NLP and machine learning are both subsets of _____.
23. A chatbot works on _____ model.

**STATE TRUE OR FALSE**

1. An NLP system takes a language as input only in the spoken form.
2. In semantic analysis, what was said is re-interpreted on what it actually meant.
3. *Syntax* involves the use of and meaning behind words.
4. In named entity recognition step, the meaning of a word is analysed based on the context in which it is used.
5. Syntax level ambiguity arises when one or more words in a sentence have multiple meanings.
6. *Textual coherence believes that in 85% of the cases, the referent is the most recently mentioned object of correct gender and number.*
7. Every production rule consists of non-terminals, arrow and terminals.
8. In a production rule, the right side of production is non-terminals while the left side is called terminals.
9. In context-free grammar, parse tree follows the precedence of operators.
10. Voice recognition is the ability of a machine to identify words spoken aloud to convert them into readable text.
11. NLP applies machine learning algorithms to process text and speech.
12. Predictive typing uses computer vision algorithms to recommend what users should type next.
13. Siri is the virtual assistant of Google.
14. Every language has its unique set of grammar and syntax rules, terms and slang.
15. Computer language has a very large vocabulary.
16. Computer language is easily understood by humans.
17. Chatbots are used to offer 24 X 7 assistance to customers by handling their queries.
18. Chatbots are used to automate repetitive and time-consuming tasks.
19. Natural language understanding generates non-linguistic outputs from natural language inputs.
20. NLU is harder than NLG.
21. Syntax level ambiguity is common when using pronouns.
22. Stop-word removal process labels tokens as *verb, adverb, adjective, noun*, etc.

23. Word tokenization divides a string written in a language into its basic components.
24. Open chatbot executes a specific conversation flow.



MULTIPLE CHOICE QUESTIONS

1. _____ of a language refers to the collection of words and phrases in that language
 1. Syntax
 2. Semantics
 3. Lexicon
 4. All of these
2. _____ studies the meaning of a sentence with respect to meaning of sentence preceding as well as succeeding it in sequence.
 1. Lexicon analysis
 2. Syntax analysis
 3. Semantic analysis
 4. Discourse integration
3. _____ is an arrangement of words in a sentence to make grammatical sense.
 1. Syntax
 2. Semantics
 3. Lexicon
 4. All of these
4. During _____ analysis, processes like tokenization, stemming, lemmatization, stop words removal, POS tagging are performed.
 1. Lexicon
 2. Syntactic
 3. Semantic
 4. Pragmatic
5. Natural language _____ uses a database to determine semantics behind words and generate new text.
 1. Understanding
 2. Processing
 3. Generation
 4. Analysis

6. _____ helps to automatically summarize text and search important pieces of data.
 1. Text classification
 2. Text extraction
 3. Text generation
 4. Machine translation
7. _____ ambiguity occurs when a sentence can be parsed in different ways.
 1. Lexical
 2. Syntax level
 3. Semantic
 4. Referential
8. _____ maps a sentence plan into a sentence structure.
 1. Text planning
 2. Sentence planning
 3. Text realization
 4. Ellipsis
9. In parsing a tree, _____ traversal method is used.
 1. Pre-order
 2. Post-order
 3. In-order
 4. Any of these
10. A grammar is said to be ambiguous if there exists _____.
 1. more than one leftmost derivation
 2. more than one rightmost derivative
 3. more than one parse tree for a given input string
 4. any of these
11. _____ is the manipulation or understanding of text or speech by a computer software.
 1. Computer vision
 2. Natural language processing
 3. Robotics
 4. Machine learning
12. NLP systems _____ to user queries.
 1. Interact
 2. Understand
 3. Respond

4. All of these
13. _____ is the virtual assistant of Microsoft.
1. Alexa
 2. Siri
 3. Cortona
 4. Ymo
14. Sentiment analysis is used to assess the _____ of the customers.
1. Attitude
 2. Behaviour
 3. Emotional state
 4. All of these
15. NLP systems cannot be used for _____.
1. Predictive typing
 2. Spelling correction
 3. Speech translation
 4. None of these
16. _____ uses NLP to organize unstructured text into categories.
1. Document summarization
 2. Topic classification
 3. Predictive typing
 4. Speech translation
17. _____ use NLP to analyse hundreds of open-ended responses to surveys to gain insights from customer feedback.
1. Document summarization
 2. Topic classification
 3. Predictive typing
 4. Speech translation
18. _____ ambiguity exists due to confusion in different words.
1. Lexical
 2. Syntax level
 3. Referential
 4. Grammatical
19. ----- ambiguity is common when using pronouns.
1. Lexical
 2. Syntax level
 3. Referential
 4. Grammatical

20. _____ analysis is the process of identifying and analysing the structure of words by breaking down a big chunk of text into elementary words.
1. Lexical
 2. Syntactic
 3. Semantic
 4. Pragmatic
21. _____ means breaking a text into elementary words.
1. Part of speech tagging
 2. Tokenization
 3. Lemmatization and stemming
 4. Stop-word removal
22. Semantic analysis draws the exact meaning from the text.
1. Lexical
 2. Syntactic
 3. Semantic
 4. Pragmatic
23. In _____ step, text is re-interpreted based on what it actually meant in the current context.
1. Lexical
 2. Syntactic
 3. Semantic
 4. Pragmatic
24. _____ chatbots follow predefined paths while conversing with the user.
1. Rule-based
 2. Menu-based
 3. Intellectually-independent
 4. AI-powered



ANSWERS

Fill Up

1. Natural language processing
2. Tokens

3. Named entity recognition
4. Natural language generation
5. Chatbots
6. Referential
7. Ellipsis
8. *World knowledge*
9. Context-free grammar
10. Parser
11. text, speech
12. Gmail
13. Natural
14. sentiment
15. Translation
16. Chatbots
17. Lexicon
18. Syntactic
19. Lemmatization and stemming
20. stop
21. Discourse Integration
22. Artificial intelligence
23. request-response

State True or False

1. False
2. False
3. False
4. False
5. False
6. *False*
7. True
8. False
9. True
10. False
11. True
12. False
13. False

14. True
15. False
16. False
17. True
18. True
19. False
20. True
21. False
22. False
23. False
24. False

Multiple Choice Questions

1. c
2. d
3. a
4. b
5. c
6. b
7. b
8. c
9. c
10. d
11. b
12. d
13. c
14. d
15. d
16. b
17. b
18. a
19. c
20. a
21. b
22. c
23. d

24. a

Previous chapter

[< 6. Computer Vision \(CV\)](#)

Next chapter

[8. Current Trends in Artificial Intelligence](#)

>

Table of contents expanded

8

Current Trends in Artificial Intelligence

8.1 AI and Ethical Concerns

We know that ethics define the discipline that deals with moral obligations and duties of humans.

Correspondingly, ethics of AI deals with ethics specific to robots and other artificially intelligent beings. These ethics can be divided into two groups.

1. **Roboethics:** Roboethics consider moral behaviour of humans as they design, construct, use and treat artificially intelligent beings.
2. **Machine ethics:** These ethics manage the moral behaviour of artificial moral agents (AMAs) that play an important role in delivering artificial general intelligence (AGIs) within existing legal and social frameworks.

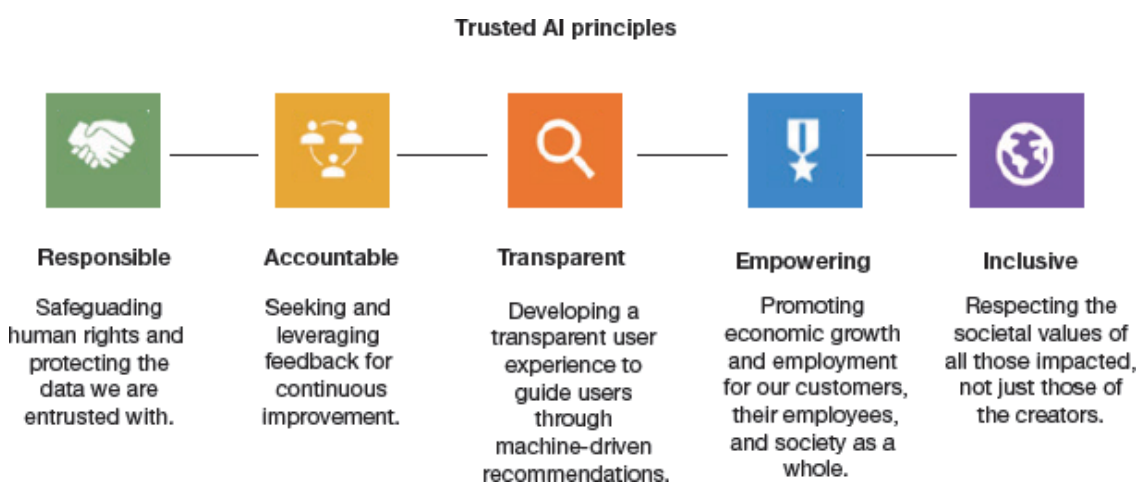


FIGURE 8.1 Trusted AI principles

The bigger concerns are as follows:

1. If AI generates human-like output, can it also make human-like decisions?
2. If AI makes human-like decisions, then are these decisions human-like?
3. If AI takes decisions of sanctioning or rejecting a bank loan, is that decision justified?
4. If AI decides whether a student should be enrolled in a college or not, is AI taking a fair decision?
5. If AI makes a human-like decision, is it human-like trustworthy?

AI is basically data + mathematical model + training based on data + predictions. What if the data provided for the training is unintentionally wrong/biased?

Issues like these are endless. Hence, getting a good understanding of them is critical to the success of this technology. Refer **Fig. 8.1** to learn about some guiding principles of a trusted AI Tool. As a citizen of the AI society, we must be aware about how AI works and the framework of AI ethics.

8.1.1 Ethical Use of Artificial Intelligence

AI systems no doubt offer a wide range of functionality for businesses, but their use also raises ethical questions.

An ideal AI system must not be biased. This is especially true when AI uses inherently unexplainable deep learning and generative adversarial network (GAN) algorithms.

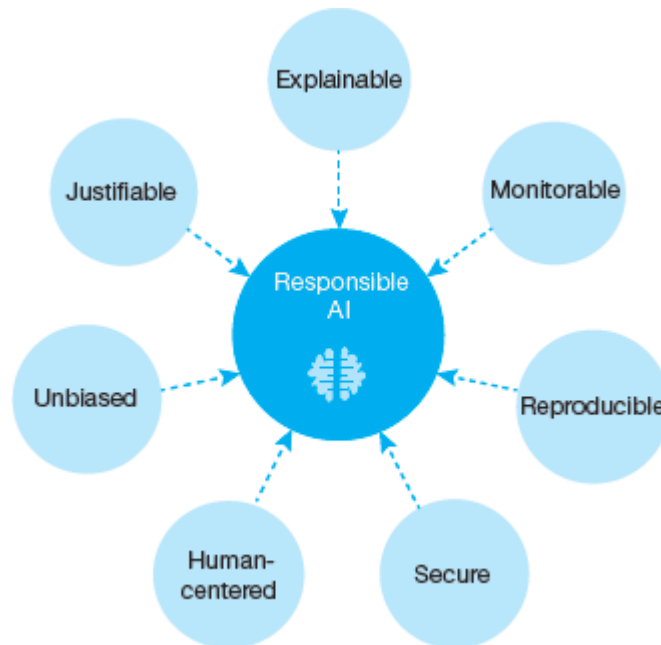


FIGURE 8.2 Features of Ethical AI System

The ability to explain is a potential stumbling block to using AI systems in industries that operate under strict regulatory compliance requirements. For example, financial institutions must explain their credit-issuing decisions. The main problem with such systems is that it will be difficult to explain how the decision was taken to refuse a credit as such decisions are made by teasing out subtle correlations between thousands of variables. In such a scenario, the AI program that is not able to explain decisions is known as black box AI.

A responsible AI system is explainable, monitorable, reproducible, secure, human-centered, unbiased and justifiable (refer **Fig. 8.2**).

Formulating laws to regulate AI is not at all easy as AI uses multiple technologies. Any strict law can hamper its progress and development. The AI technology is changing so rapidly that new technology breakthroughs sometimes make existing laws obsolete. For example, existing laws regulating the privacy of conversations and recorded conversations do not cover the challenge posed by voice assistants like Amazon's Alexa and Apple's Siri. These digital assistants collect data not to distribute them but to improve its machine learning algorithms. This is a golden chance for criminals having malicious intent.

8.1.2 Is AI Dangerous? Will Robots Take Over the World?

Rapid growth and increase in the capabilities of AI systems have made us wondering about the risks involved in using AI. Some of us think that AI will quickly takeover tasks performed by humans. But most researchers believe that super-intelligent AI is unlikely to exhibit human emotions, so there is no point thinking that AI will become malevolent. AI can become a risk only in two conditions.

AI system is specifically programmed to do something devastating.

For example, autonomous weapons are AI systems that are programmed to kill. If acquired by scrupulous people having ill intentions, such weapons could inadvertently lead to an AI war, and mass casualties, potentially even the end of humanity.

Autonomous weapons are designed to be extremely difficult to 'turn off', and once they become operational, humans could themselves rapidly lose control over them. This risk is also present with narrow AI when given autonomy.

Just imagine what would happen if an autonomous drone with facial recognition as well as a 3D-printed rifle, pistol or other gun becomes available. Or if a self-driving car, connected to the Internet is hacked to get into some serious accident.

Even in hospitals, more and more equipment are now connected to the Internet. If any or all of these are hacked, then can you imagine what the hacker can do with the patient's body?

AI could be programmed to do something beneficial, but the method used to achieve its goal can be highly destructive. This is especially true when the AI programmer asks the machine to complete a task, but does not clearly outline the goals. For example, you can instruct a self-driven car to program to take you to a particular destination as soon as possible. However, the instruction 'as soon as possible' fails to address safety, road rules, etc. The smart car may successfully complete its task, but after creating a great havoc. So, there must be some provision to con-

tinuously monitor and control the machine, its goals and process being performed to achieve that goal.

Of 9 million low-skilled services and BPO roles, 30 per cent or around 3 million will be lost by 2022, principally driven by the impact of robot process automation or RPA.

As of now, there is no liability for actions on machines. There is no clarity on what legal aspects bind machines when they become increasingly smart. The following questions may arise:

1. Do we judge AI systems the same way as we judge a human?
2. Who is responsible if the AI system becomes self-learning and autonomous to a greater extent?
3. Is there any error margin for AI machines, even if it has fatal consequences?

Apart from these two serious concerns, AI also poses some additional threats/risks that calls for attention. These threats are discussed below.



Credit: PaO_STUDIO / Shutterstock

FIGURE 8.3 Effect of AI on jobs

1. **The immediate risk posed by job automation:** As AI robots become smarter and more dexterous, they will quickly replace humans. For example, you must have heard of restaurants having robots as wait-

ers. Just imagine, where the less educated people will go if machines start taking their jobs (as seen in **Fig. 8.3**)?

It is often rightly feared that the deployment of AI systems for job automation will eventually replace certain types of jobs, especially those that are predictable and repetitive. According to a 2019 Brookings Institution study, 36 million people work in jobs that they may soon lose owing to automation as at least 70 percent of their tasks (varying from retail sales, market analysis to hospitality and warehouse labour) will be done using AI. In fact, a newer Brookings report even states that white collar jobs may be worst affected. As per McKinsey & Company report (2018), the African American workforce will be hardest hit by automation.

2. **Biased algorithms:** We know that computers work on GIGO concept which says Garbage-In-Garbage-Out. This is a very serious limitation in AI applications. If we feed our algorithms data sets that contain biased data, then the output generated from such systems will only produce biased results. This biased result, if applied to solve a real-world problem, may lead to an even bigger problem.

A dynamic advertising billboard in Utrecht was switched off because the spy software installed on these billboards aroused public outrage.

3. **Too little privacy:** When using IoT or AI systems, a lot of data is generated. It is said that approximately 2.5 quintillion bytes (or 2.5 million terabytes) of data is added each day. It is interesting to note that 90% of the digital data has been created in the last two years. A lot of data is required for the proper functioning of the smart systems. As a result, much data about us is collected thereby eroding our privacy. Once the systems collect our data, we have no means to know what data about us is used by whom and for what purpose.

These days, cameras can easily be fitted with facial recognition software to capture details about us (including our gender, age, ethnicity, gesture and state of mind). Do you know that in China, some police officers wear glasses with facial recognition technology having a database with facial pictures of thousands of 'suspects' (judged on the basis of certain behaviour)?

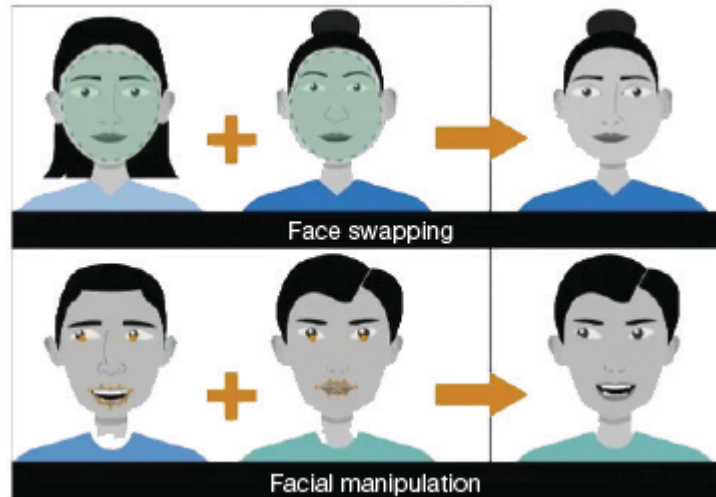


FIGURE 8.4 AI Tools can be used for Face Swapping and Face Manipulation

4. **Everything becomes unreliable:** These days, fake news and filter bubbles are not uncommon. Smart systems can create faces, compose texts, produce tweets, manipulate images, clone voices and engage in smart advertising.

An AI system can be used to depict day into night or create highly realistic faces of people who have never existed.

For example, open-source software Deepfake can easily stick pictures of faces on moving video footage. Due to such fake videos, celebrities are the worst affected. People with malicious intentions easily create pornographic videos starring them. Even normal citizens are blackmailed with morphed images and/or videos. This is often known as **Faceswap video blackmailing** (refer **Fig. 8.4**).

Even in politics, fake or manipulated videos are produced at a high speed to influence the opinions of the masses.

Did you know that the company Cambridge Analytica, was found to manage access to data from 87 million Facebook profiles of Americans to campaign for President Trump so that he could come to power?

8.1.3 Ethics in AI

The current world is quickly getting transformed by artificial intelligence and those who use this technology are living a richer and easy-going life. However, this is possible only when we talk about ethical AI, not just AI.

The term, ethical AI, is composed of two words—ethical and AI. We are now already familiar with the term AI. Ethical means one that is according to ethics. And ethics relates to moral principles that govern the behaviour and actions of a group or an individual. So, ethical AI is the theory behind developing computer systems that can perform tasks that require human intelligence. It focusses on how right, how fair and how just is the AI's output, outcome and impact.

Example: We know that AI can produce biased results if it is trained with biased data or if it does not do anything to avoid bias. So, any AI system that works on biased data is actually not ethical (or unethical). For example, Microsoft released a chatbot called Tay on Twitter in 2016, to learn by engaging people in dialogue through tweets or direct messages. However, trolls made Tay learn all negative words that spread hatred for women, a particular race and all sorts of biased data within hours. Tay was also taught to give repulsive and toxic responses. This aroused the anger of Twitter users and finally Microsoft had to silence Tay forever in even less than 24 hours.

An ethical AI system designed to solve a particular problem, uses unbiased data and is trained using the right learning model for the problem and is monitored to evaluate if the results produced by it is right and fair.

8.1.4 AI and Bias

Think about the following statements.

1. Most images that show up when you do an image search for 'doctor' are white men.
2. Images and videos of a 'doctor' show usually shows a male character and that of a 'nurse' is a woman.

3. All virtual assistants like Alexa, Siri, Google assistant, etc. work with female voice.
4. Computer vision systems frequently give an error while recognizing people of different colour.

Can you think of more such cases where we witness biasness towards a particular gender, colour, caste, religion, country, etc? Technically, AI bias occurs when an algorithm produces results that are systematically prejudiced towards a particular group of people and thus produces skewed output. AI applications may have built-in biases because ultimately, they are created by humans who have conscious or unconscious preferences that may go undiscovered until the algorithms are used at a massive scale. Let us understand three basic sources of bias in AI systems.

Data

Any AI system is just as good as the data we put into them. Naturally, adding biased or skewed data can never give fair results. Machines do not have the intelligence to differentiate between right or wrong training data.

For example, many-a-time, you must have noticed that Google or any other voice assistant replies in American accent and at times gives wrong results when we give a command in Indian accent. This happens because the AI system is trained on recorded speech from white, upper-middle class Americans, making it difficult for the technology to understand commands from people not belonging to this category.

Algorithm

Algorithm can further amplify the biases injected by skewed data. For example, if you Google Image Search 'Teacher', then the image classifier algorithm trained on the images available in public domain shows more women teaching as opposed to male teachers. AI algorithms must be designed to maximize accuracy. But in this case, the AI algorithm may decide that all people teaching are women, despite the fact the training data has some images of men in the classroom. So, this is a case of gender bias in the AI system that becomes a part of AI algorithm due biasness in data.

People

People who are developing the AI system, that is, engineers, scientists, developers, etc. are the next big source of bias. Humans design AI systems to give the most accurate results with the available data. So, data feeders that may be humans may again hit the success unknowingly. ***Therefore, it is rightly said that ethics and bias are not the problem of the machine but that of the humans behind the machine.***

8.1.5 Towards Ethical and Trustworthy AI

Today, several companies worldwide are using algorithm-based ‘emotional AI’ hiring platforms to augment and lower the financial burden of their recruitment processes. While these systems may hire in a fair and unbiased manner, there are cases reporting women applicants being disproportionately rejected based on years of biased data in a male-dominated sector.

Let us now see few ways in which world is trying to deal with this situation.

Regulating a More Ethical AI

In April 2021, the European Commission launched its first ever legal framework on AI to guarantee the safety and fundamental rights of people and businesses, while strengthening AI uptake, investment and innovation across the EU.

The new risk-based approach will not only set strict requirements for AI systems but also immediately ban AI systems which are considered to ‘be a threat to the safety, livelihood and rights of people’ including ‘systems that manipulate human behaviour, circumvent users’ free will and allow social scoring by governments’.

Later, in June 2021, Australia launched a similar AI ethics framework to guide businesses, governments and other organizations to responsibly design, develop and use AI.

Company and Organizational Engagement

Besides necessary regulation and standards, these days, organizations are building trust in AI technologies through cultural and educational programmes, risk assessments and third-party audit programmes. This would help them to move towards a prevention, detection and response framework (like anti-corruption, prevention of tax evasion frameworks).

Rights and Activist Groups

AI systems are yet to go a long way. In this journey, the relationship between people and technology will be redefined continuously. To ensure that our societies move forward on the path of ethics, it is critical that AI systems are human-centric. Civil rights and activist groups need to challenge the discourse and amplify the voices of those who are most adversely affected by new technologies.

We must all ask the question—Is the technology necessary or is there any alternative? Only if the benefits outweigh the harm, we must check from whom the AI systems work.

Ensuring Data Privacy

Companies are now taking measures that are important to keep data safe and protect it from potential misuse. Privacy protection is now becoming an integral part of any ethical AI system. Data ethics and privacy in such systems is ensured by the following:

1. Encrypting the data.
2. Ensuring that the AI algorithm cannot learn from outside its dataset.
3. Using secure computation so that even the people who developed or work on those AI systems are not able to see or access the data.
4. Restrict reverse engineering the AI model so that no user can access data used for model training.

Diversify Your Team

Since bias is an important issue and results from an AI tool will be used by people from different racial, gender and economic identities, we need

to make sure that there should be no bias. For this, build diverse teams to reduce the potential risk of bias falling through the cracks. A diverse team has data scientists, business leaders, professionals with different educational backgrounds and experiences, such as lawyers, accountants, sociologists and ethicists. This would help the company to get their views on any sort of bias so that it could be mitigated in a timely manner. The team should continually analyse data and algorithms ensure fairness. There are many tools like **Bias Analyzer** which automate this process and also show the costs and benefits associated with a variety of possible mitigation actions.

8.1.6 Why is Ethical AI Important?

Since AI is used in areas like medicine, law enforcement, recruiting, data privacy, military defense or self-driving vehicles, it makes it mandatory that the AI system must produce accurate, transparent and understandable results that is in synchronization with the ethical standards and norms of our society. To understand the need of ethical AI systems, we must first understand that a biased or incorrect output from AI system that assist in law enforcement, job recruitment, defense work, self-driving vehicles can result in eroding people's privacy (by misusing data in unintended ways), and taking decisions that are impossible for people to understand and to access liability for damages when harm is caused.

Apart from being **unbiased and accurate** (as shown in **Fig. 8.5**), the credibility of AI systems also depends how we choose to develop and use it. We can just hope that AI will be used for purposes that benefits mankind. But there is nothing that can stop a person from creating systems that can be disastrous for humanity.

In October 2019, researchers found that an algorithm used in US hospitals to predict which patients may require extra medical care heavily favoured white patients over blacks. This bias crept in because of the fact that white patients paid more bills and availed extra medical facilities for the same medical problem as compared to black.

Amazon's hiring algorithm also suffered from bias related issues. The algorithm was found to be biased against women. This was probably due to the fact that the system saw a greater number of men applicants than women.



FIGURE 8.5 An ethical system must be unbiased and accurate

Nonetheless, efforts are being made by numerous agencies, committees, coalitions and expert groups to ensure that we are on the right track to make AI systems do more to heal than harm. For example, the IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems that put out *Ethically Aligned Design*, a publication from 2016 outlines the guiding standards for developing and administering ethical AI solutions. Therefore, ethical AI aims to *ensure is that AI is doing good for the planet*.

Going further on this, the European Commission High-Level Expert Group on AI identified seven guiding principles for creating Trustworthy AI. These guidelines are as follows:

1. AI systems should support **human autonomy** and decision-making.
2. AI systems should be technically robust and **safe** to use. They must have a fallback plan that can be used when something goes wrong.
3. AI systems should ensure data privacy by protecting user's data, and providing adequate mechanisms to maintain the **quality and integrity** of the data.
4. **AI systems should not work on biased data. The system should be clear, explicable and transparent about its underlying data and the models.**
5. **AI system should be trained with diverse, non-discriminatory and fair data and models** to avoid unfair bias.
6. AI systems should be designed to **benefit** everyone, now and in future. They must focus on **societal and environmental wellbeing** to be sustainable and environmentally friendly.

7. **An AI system should have the accountability** to ensure accurate, unbiased outcomes with utmost responsibility.
8. An AI system must give due importance to data privacy and data security. A system that provides design confidentiality, transparency and security into their AI programs must ensure that data is collected, used, managed and stored safely and responsibly.
9. An AI system must be accountable for its decisions. There must be someone in the organization who will be held responsible for any problematic situation.



Have you ever wondered how much information Google has about you? In an Android phone, Google has your entire Contact List. Google knows the names and numbers of people you talk to. Google knows your location. It knows where you go every day and on holidays. Google knows what email you write to whom. Google has your photographs (Google Photos), knows details about your friends (Google Hangouts). All this is done in lieu of free services it provides. But imagine what big risk it poses to our privacy.

8.1.7 Impact of AI on Jobs

Today, AI is on the tip of everyone's tongue and one of the most in-demand areas of expertise for job seekers. The World Economic Forum has predicted that AI and ML will displace 75 million jobs but would also create 133 million new ones in a few months.

Some of our favourite sci-fi movies have shown that the advent of this technology has created a fear that AI will one day make human beings obsolete in the workforce. As the technology is advancing, many tasks that were once executed by humans have now been automated.

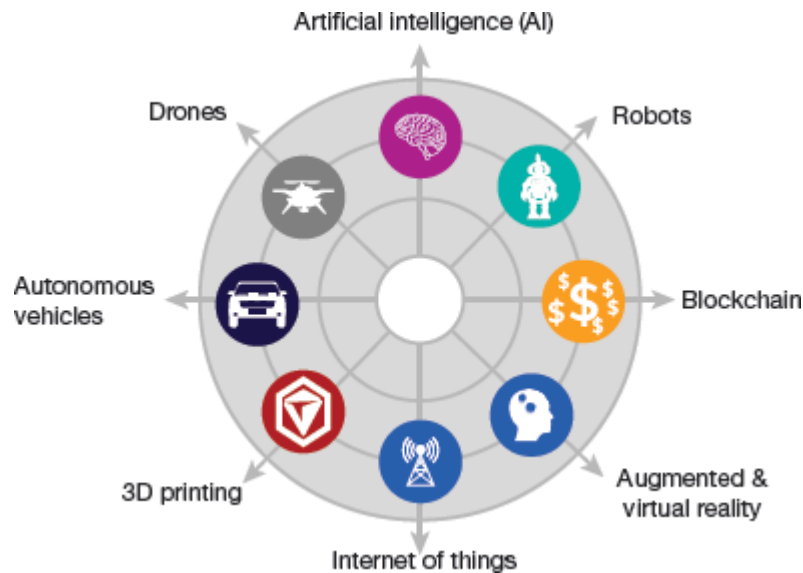


FIGURE 8.6 Emerging Areas in IT

A two-year study from McKinsey Global Institute suggests that by 2030, intelligent agents and robots could replace as much as 30% of the world's current human labour. That is, automation will displace between 400 and 800 million jobs and require approximately 375 million people to switch job categories entirely.

However, job losses due to AI is a fear far from reality. It will just introduce a paradigm shift, similar to the one which occurred after the Industrial Revolution. When the Industrial Revolution took place, many feared that people will lose jobs to machines. But now we see that revolution has generated lot more opportunities and jobs. Similarly, with AI, many professions will become obsolete and disappear, but other new emerging occupations in areas listed in **Fig. 8.6** will become popular. Learning AI skills will help you gain a significant career advantage.

Jobs which will grow with the help of AI include the following.

Creative jobs will get refined and advanced by use of AI. These jobs are usually performed by professionals like artists, doctors and scientists. AI will not only enhance efficiency but also make these jobs less complex for humans.

AI will probably not make human workers obsolete, at least not for a long time.

Management jobs cannot be replaced by artificial managers because anytime, human managers will have to control artificial managers. Managing is a very complex task that involves deep understanding of emotions, profession, ethics, people and communication. Human managers can use certain tools to become more effective at their jobs.

Tech jobs including programmers, data scientists, data analysts, big data engineers and all those who work on the creation and maintenance of AI systems will always be in high demand. However, a few tech jobs which are in demand today may become less common, while others may become more vital.

A few job titles which can be expected to appear by 2030 include Chief Bias Officer, Data detective, Man–Machine teaming manager, AI business development manager, AI assisted medical professional, AI tutor, to name a few.

In many sectors of business, AI has grown by 270% over the last four years. Rather than promoting the obsolescence of humans, AI will continue to drive massive innovation that will fuel many existing industries and could have the potential to create many new sectors for growth, ultimately leading to the creation of more jobs.

We must understand that although AI has done a lot to replicate the efficacy of human intelligence in executing certain tasks, AI programs suffer from major limitations. They can only solve one problem at a time. These systems become rigid, and are unable to think outside their prescribed programming. In contrast, humans have generalized intelligence which helps them solve any kind of problem, think abstractly to make critical judgement across all sectors.

Moreover, AI can learn only with massive amounts of relevant data that is available at the right time. Many a time, this data is protected under issues of privacy and security. In addition, data may have bias. And it is quite obvious that biased data will never give accurate results.

AI Is becoming the standard in all businesses, not just in the world of tech

Machines may also suffer because of availability of limited computation and processing power. For example, the cost of electricity required to power one supercharged language model AI was estimated around \$4.6 million.

So, AI has the potential to ultimately create more jobs, not less. The question is not 'humans or computers' but 'humans and computers' involved in complex systems that advance industry and prosperity. Nobody would want to be behind the curve when it comes to AI. In fact, 90% of leading businesses have already invested in AI technologies. More than half of businesses that have implemented some manner of AI-driven technology report experiencing greater productivity.

AI and machine learning top the skills required by companies these days and jobs in this area are expected to increase by 71% in the next five years. AI is likely to have a strong impact on certain sectors in particular. These are discussed below.

1. **Medical:** The medical industry has huge data which can be utilized to create predictive models related to healthcare and by physicians for diagnosis. And at a higher level, AI and automation will also help to eliminate disease and world poverty.
2. **Automotive:** Autonomous vehicles, autonomous navigation and manufacturing within the automotive sector have all been possible with AI technology.
3. **Cybersecurity:** During the pandemic, cyber frauds rose by 600% as hackers capitalized on people working from home, on less secure technological systems and Wi-Fi networks. AI and machine learning provide powerful tools to identify and predict threats. AI is also playing a vital role in the field of finance as it can process large amounts of data to predict and catch instances of fraud.
4. **E-commerce:** On e-commerce websites (like Amazon), AI is used to design chatbots, recommend products, use image-based targeted advertising, and assist in warehouse and inventory automation.

5. **HR applications:** An HR department gets thousands of applications every day. AI tools can be used to reject 75% of resumes that do not fit the current context. Prior to AI, HR managers had to devote considerable time to filter applications to select suitable candidates. Data from LinkedIn shows that recruiters spend up to 23 hours looking over resumes for one successful hire.
6. **Legal profession:** Support services with *document handling -classification, discovery, summarization, comparison, knowledge extraction, and management* use AI for efficiency and accuracy.

Thus, the AI revolution will lead to a new era of *prosperity, creativeness and well-being*. Humans will no longer be asked to perform routine, limited-value, full-time jobs. Rather, jobs will be *flexible and offer selective premium services*. There will be more jobs related to programming, robotics, engineering, etc. And, blue-collar and white-collar jobs will be eliminated.

8.2 AI as a Service (AIaaS)

In artificial intelligence as a service (AIAAS), companies use off-the-shelf AI tools to implement and scale AI techniques at a very low price as compared to building a complete in-house AI system.

When a software is provided as a service across the network, it usually uses cloud computing. Similarly, the idea of providing AI services using cloud computing has allowed even small business organizations to use cost effective solutions to improve their performance in a big way.

8.2.1 Factors Triggering Growth of AIaaS

In the last couple of years, AIaaS is growing fast as a business opportunity and many startups have joined to lead the AI revolution. Some major advancements in IT in the last few years have fostered the adoption of AIaaS. Some examples are as follows:

1. Availability of cloud platform with a variety of affordable options for enterprise data management.

2. Data storage technologies have improved and have become cheaper and reliable.
3. Streaming devices and IoT technologies generate massive amounts of data which when analysed can render vital information required to gain competitive advantage.
4. Availability of semi or fully automated data management, analytics and BI products changed the way business was analysed.

Moreover, some major factors that have motivated business to use AIaaS are listed below.

1. Expensive infrastructure
2. Lack of trained programmers
3. Very high charges charged by programmers for implementing AI in a particular organization
4. Many companies lacked sufficient data to analyse

8.2.2 The Growth of AIaaS

With cloud services becoming incredibly accessible, AI has seen a big boom. Now, AI can be made available to a greater number of companies and even those companies can collect and store any amount of data.

Previously, companies were hesitant to build their own clouds to develop, test and utilize their own AI systems. But now with AIaaS, every other company can take advantage of data insights without making huge investment in talent and resources. Moreover, AIaaS has been a lucrative option as business can now easily accomplish the following goals.

1. **Focus** on its core area rather than getting involved in becoming an expert in machine learning.
2. **Reduce operational and infrastructural cost** to a big extent. A complex AI system may require many parallel machines and speedy GPUs. Not many companies can afford such huge investment especially in its initial years. But with AIaaS, every small company can also harness the power of machine learning at significantly lower costs.
3. Minimize the **risk of investment**.
4. **Time** required to deploy a solution is short.

5. **Pre-built algorithms** make complex data management accessible to even small business organizations.
6. Availability of **on-demand services** empower business with a competitive edge.
7. **Increase the benefits** obtained by analysing data trends.
8. **Develop cost effective**, flexible and transparent solution. AIaaS provides a software to the organization that enables them to access machine learning capabilities using templates, pre-built models and drag-and-drop tools to assist developers in building a more customized machine learning framework. Companies no longer have to maintain a team with AI skilled professionals, train them, upgrade their knowledge and spend on areas that only partially support decision-making.
9. **Improve its strategic flexibility**. Companies can only pay for what they actually use. Though machine learning requires a lot of computing power to run, small companies may actually be using only in short amounts that too in spurts. So, in such a scenario, AIaaS is a huge saving.

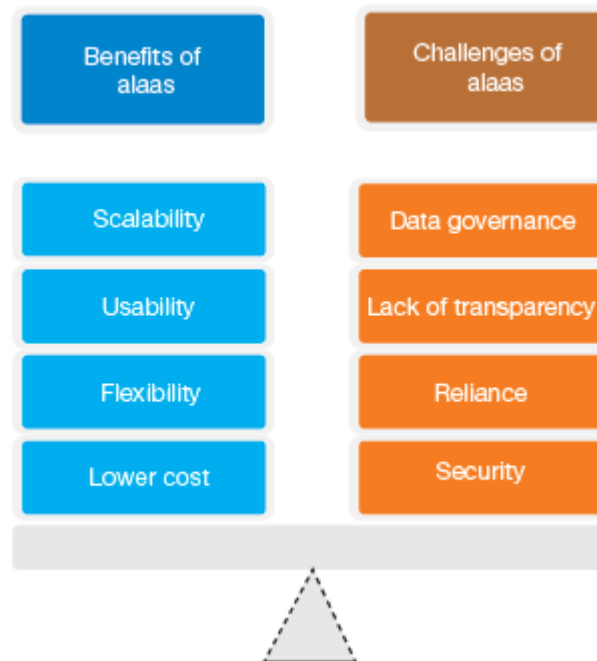


FIGURE 8.7

10. **Scalability is another big advantage of AIaaS**. Now companies can delve into AI technology with smaller projects. If they find it beneficial, then probably, they can experiment more services. At any point in time with the change in demands of the ongoing project, companies can scale up or down their demand for services.

Figure 8.7 summarizes the advantages and shortcomings of providing AI as a Service.

8.2.3 Challenges of AIaaS

1. **Reduced security:** To avail services over cloud, companies have no option but to share their data with third-party vendors. Lot of data security techniques need to be incorporated to ensure that data is safe during storage and transmission and illegal access is strictly avoided.
2. **Reliance;** When working with third parties, the organization becomes dependent on them to satisfy their information needs. This results in delay in getting response.
3. **Reduced transparency:** In AIaaS, organizations are paying to get service, but not the access. Third party vendors take the input and give the output. The entire process works like a black box. The organization has no idea about the inner workings (like algorithms used). This may result in confusion or miscommunication regarding the stability of data or the output.
4. **Data governance:** Every industry and organization has its own data policy. These policies may restrict data storage on a third-party cloud, thereby preventing a company from taking full advantage of AIaaS.

8.2.4 Vendors of AIaaS

There are multiple AI provider platforms offering various machine learning and AI services. Organizations can match their needs with the services and compare the cost at which those services can be accessed to select the offering that is best suited to their current requirements.

Cloud AI service providers allow users to use GPU-based processing for intensive workloads along with their staffing and maintenance staff as well as catering to the hardware changes for different tasks. Some major vendors of AIaaS are Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platform (GCP). Each of these vendors offer different types of bots, APIs and machine learning frameworks to cater to the organization's demands. Apart from these, countless start-ups are focusing on AIaaS. It is not uncommon for larger companies to purchase smaller companies to add the developed services to their portfolios.

An organization decides on an AI service based on its goals, business size, available budget, technical capabilities of its in-house teams and the amount of data that needs to be processed.

MonkeyLearn is an AI platform that simplifies text analysis through intuitive, no-code tools. Users can either use pre-trained model like survey analyzer to classify customer feedback by topic or build customized machine learning models to detect sentiment, keywords, and topics in the data. Once developed, these models can be integrated with apps through point-and-click integrations or via the API.

IBM Watson hosts a suite of AI tools and offers many pre-built applications, like Watson Assistant and Watson Natural Language Understand to build virtual assistants and to perform advanced text analysis tasks respectively. Similarly, IBM Watson Studio can be used to build, train and deploy machine learning models across any cloud. With these offerings even users with no expertise on machine learning or data science can make the most of their data.

Microsoft Azure is Microsoft's public cloud computing platform that provides a range of AI and machine learning solutions for developers. For example, Azure Cognitive Services can be used to add computer vision or text extraction capabilities to apps using APIs. Similarly, Azure Bot Service can be used to incorporate any type of bot in the application.

Google Cloud ML is Google's AI platform that basically facilitates data scientists and developers working with big data to create and deploy machine learning projects.

AutoML is being extensively used to train custom machine learning models for text analysis, image classification, translation and more. Data in the datasets can be visualized, 'what-if tools' can be used and metrics can be analysed to assess performance of the model.

AIPaaS or AI Platform-as-a-Service is an end-to-end solution like a cloud platform, using which business organizations can utilize required services on a pay-per-use or pay-per-service basis. With AIPaaS, users can use third-party APIs that provides a complete, intelligent data management platform.

Software-as-a-Service (SaaS) market is valued at \$133 billion and the advanced technology platforms-as-a-service market globally is forecast to reach about \$11 billion by 2023 and surpass ‘\$88,500 million by the end of 2025.’

8.3 Robotics

We have read that robot is a programmable machine that can complete a task. Correspondingly, the term robotics is the study focused on designing, developing and programming physical robots which are able to interact with the physical world and automate one or more tasks. Each robot has a different level of autonomy varying from human-controlled bots that carry out tasks to fully-autonomous bots that perform tasks without any external influences.

8.3.1 Artificially Intelligent Robot

Many people think that AI and robotics are synonymous. But this is not true. **Table 8.1** highlights the difference between an AI program and a robot system.

TABLE 8.1 Difference between AI Program and Robot

AI Programs	Robots
They operate in computer-simulated worlds. Receives input in the form of symbols and rules. To operate an AI program, general-purpose or special-purpose computers are required.	They operate in the real physical world. Receives input as analogue signal or in the form of the speech waveform. To operate a robot, special hardware with sensors and effectors are needed.

Moreover, from the Venn diagram as shown in **Fig. 8.8**, it is clear that there is one small area where the two fields overlap and, in this area, we have artificially intelligent robots (AIRs). These AIRs are the bridge between robotics and AI.

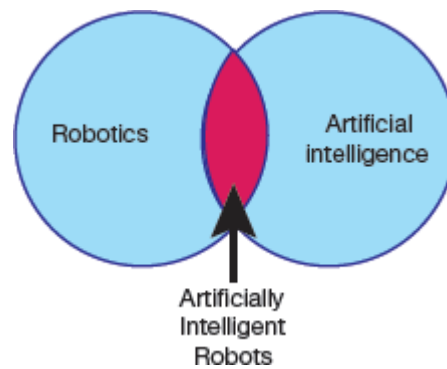


FIGURE 8.8 Venn diagram depiction relationship between AI and Robotics



Credit: PaO_STUDIO / Shutterstock

FIGURE 8.9 AI Robot

Although many AIRs (refer **Fig. 8.9**) are controlled by AI programs, they are not artificially intelligent. For example, most industrial robots are programmed to carry out a repetitive series of movements which do not require artificial intelligence. However, such non-intelligent robots have limited functionality.

To perform complex tasks, an artificially intelligent robot must implement some AI algorithm(s). For example, consider the scenarios given below to understand how AI can guide the functionality of a robot.

Case 1: A warehousing robot may use a path-finding algorithm to navigate around the warehouse.

Case 2: A drone may use autonomous navigation to return to its owner when it is about to run out of batteries.

Case 3: A self-driving car uses AI techniques to detect and avoid potential hazards on the road.

8.3.2 Characteristics of Robots

A robot is a **mechanical device** which completes its tasks in the environment for which it is designed. For example, the Mars 2020 Rover's wheels are motorized and made of titanium tubing to firmly grip the harsh terrain of the red planet.

1. Robots need **electrical components** that control and power the machinery. Usually, they draw electric current using a battery.
2. Robots work on instructions written using **computer programming**. These instructions tell the robot what to do, when to do and how to do it.

8.3.3 Types of Robots

Mechanical bots come in a variety of designs, shapes and sizes to efficiently perform the task they are designed to perform. For example, RoboBee is a 0.2-millimeter-long robot while Vindskip is a 200-metre-long robotic shipping vessel. To classify them broadly into different categories, robots can be divided based on their capabilities to perform a particular task.

Pre-Programmed Robots

They operate in a controlled environment where they perform simple, monotonous tasks. For example, a mechanical arm on an automotive assembly line to weld a door on, to insert a certain part into the engine, etc. (as shown in **Fig. 8.10**) is an example of such a robot that performs its tasks faster and more efficiently than a human.



Credit: Pop Nukoonrat / 123RF

FIGURE 8.10 Pre-Programmed Robots

Humanoid Robots

They look like or mimic human behaviour and usually perform human-like activities (like walking, carrying objects). These days, humanoids look like us. For example, Hanson Robotics' Sophia and Boston Dynamics' Atlas.

Autonomous Robots

These machines operate independently of human operators and are usually designed to perform tasks in open environments without any human supervision. Autonomous robots perceive the world around them using sensors and then use decision-making capabilities to take the optimal next step based on their data and mission. For example, Roomba vacuum cleaner uses sensors to roam freely throughout a home. Other examples of autonomous robots are lawn trimming bots, hospitality bots, autonomous drones and medical assistant bots.

Most industrial robots are non-intelligent.

We often call these robots, cobots. Cobot, or a simple collaborative robot, is a non-intelligent robot. For example, a cobot can be programmed to pick up an object and place it elsewhere. This is done by training a specialized computer vision program to identify different types of objects generally by using an AI algorithm called Template Matching. Broadly, this is an autonomous function as no human intervention is required once the cobot has been programmed to do its task efficiently. Moreover,

the task does not require any intelligence as the cobot will always be picking the object in the same way until its instructions are not changed.

Teleoperated Robots

These are semi-autonomous bots that use a wireless network to enable human control from a safe distance (as shown in **Fig. 8.11**). They are usually deployed in extreme geographical conditions, weather and circumstances. For example, drones used to detect landmines on a battlefield, robots used to fix underwater pipe leaks during oil spill are examples of teleoperated robots.



Credit: MONOPOLY919 / Shutterstock

FIGURE 8.11 Teleoperated Robots

Augmenting Robots

Also known as VR robots, they are used to enhance or replace current human capabilities. With their help, science fiction could become a reality very soon by making humans faster and stronger. For example, robotic prosthetic limbs or exoskeletons are used to lift hefty weights.

8.3.4 Types of Robots Based on Degree of Human Control

Independent Robots

They perform their work autonomously and independent of human operator control. So, they need to be intensely programmed. Such robots are deployed to replace humans for performing dangerous, mundane or otherwise impossible tasks ranging from bomb diffusion and deep-sea travel to factory automation. Though independent robots eliminate certain jobs, they also present new possibilities for growth.

Dependent Robots

These robots are non-autonomous robots as they interact with humans to perform their actions. Humans guide robots to enhance and supplement their already existing actions. For example, these days, advanced prosthetics (the branch of surgery concerned with the making and fitting of artificial body parts. For example, a piece of flexible material applied to a person's face or body to change their appearance temporarily) are controlled by the human mind.

In 2018, Johns Hopkins APL created a popular example of a dependent robot was created by in 2018 for a patient (Johnny Matheny) whose arm was amputated above the elbow. A modular prosthetic limb was fitted in Matheny that was controlled by electromyography, or signals sent from his amputated limb that controls the prosthesis. Over time, he could accurately move his arm and even play the piano. At this time, the signals sent from his amputated limb became smaller and less variable.

Chatbots

Software robotics, also called bots, are computer programs that perform tasks autonomously. For example, a chatbot is a computer program that simulates conversation both online and over the phone and is often used in customer service scenarios. While simple chatbots answer questions with an automated response, more complex digital assistants learn from user information.

Robotics involves building robots physical whereas AI involves programming intelligence.

Types of Bots

Bots, or software robots only exist on the Internet and originate within a computer. Hence, they cannot be called robots. We have learnt in the characteristics of a robot that a robot has a physical form, such as a body or a chassis. Some popularly used bots are as follows:

1. Chatbots carry out simple conversations to provide 24 X 7 customer service.
2. Spam bots collect email addresses and send spam mail.
3. Download bots automatically download software and apps.
4. Search engine crawler bots scan websites and make them visible on search engines.
5. Monitoring bots collect statistics to report speed and status of the website.

8.3.5 Components of a Robot

Robots are created to perform a variety of tasks to present a solution for a wide range of problems. For this, they need a set of specialized components that are discussed below.

Control System

Control system is a CPU that directs a robot's task at high level. It performs all computations to tell a robot how to utilize its specific components, like how our brain sends signals to other parts of the body to complete a specific task. The tasks that a robot completes vary from an invasive surgery to assembly line packing.

Sensors

Sensors are devices that detect the events or changes in the environment and send data to the computer processor. For this, they are equipped with other electronic devices to provide electrical signals that allows the robot to interact with the world. Basically, the electrical signals act as a stimulus that are processed by the controller. Based on the stimulus received, the controller instructs the robot to interact with the outside world. This helps the robot to efficiently respond to real-time data. Popularly used sensors within a robot include video cameras (as eyes), photoresistors (react to light) and microphones (as ears). All these sensors allow the robot to capture information about its surroundings and process it to deduce the most logical conclusion and give commands to other components.

Robots are programmable, autonomous or semi-autonomous machines.

It is through these sensors that the robots get the ability to see, hear, touch and move like humans. *Sensors are devices or machines which help to detect the events or changes in the environment and send data to the computer processor. These devices are usually equipped with other electronic devices.* Similar to human organs, the electrical sensor also plays a crucial role in artificial intelligence and robotics. AI algorithms control robots by sensing the environment, and provide real-time information to computer processors.

Actuators

Actuators are the motor parts that facilitate a robot's movement. We have read that a machine is a robot if it has a movable frame or body and actuators are the components that cause movement. Actuators are made up of motors that receive signals from the control system and move in tandem to carry out the movement necessary to complete the assigned task. They are made of metal or elastic.

Electric motors are the devices that convert electrical energy into mechanical energy and are required for the rotational motion of the machines.

Actuators help in moving and controlling a robot by using energy that can be electrical, hydraulic and air, etc. This means that actuators are usually operated using electricity (electrical), compressed air (pneumatic actuators) or oil (hydraulic actuators). Actuators can create linear as well as rotary motion.

Power Supply

Power supply is done to the robot by a battery. Just as we need food to do our work, robots require power to operate. The main use of power supply is to convert electrical current to power the load. Stationary robots used

in a factory work on AC power while others operate via an internal battery.

While a majority of robots work on lead-acid batteries as they are safe and have long shelf life, others use compact and expensive silver-cadmium batteries. These batteries are chosen depending on safety, weight, replaceability and lifecycle of the robot.

AI algorithms are used in Google searches, Amazon's recommendation engine and GPS route finders.

However, for future robotic development, pneumatic power from compressed gasses, solar power, hydraulic power, flywheel energy storage, organic garbage through anaerobic digestion and nuclear power are also being considered as potential sources of power.

Example of a pure AI is AlphaGo.

Pure AI is usually found in games. For example, in 1997, AI Deep Blue could easily defeat world champion. A more recent example is AlphaGo which won Lee Sedol, the world champion Go player, in 2016.

End Effectors

End effectors are the physical, typically external components of a robot that facilitate it to complete a task. For example, in factories, robots usually have interchangeable tools like paint sprayers, gripping claws or even hands for undertaking tasks like deliveries, packing, bomb diffusion, etc.

All these components are central to every robot's construction.

8.3.6 AI Technology Used in Robotics

In this section, we will list AI technologies that are used in robotics. We have already studied some of them. These technologies are discussed below.

1. **Computer Vision:** As studied earlier, Computer Vision is an important domain of artificial intelligence that extracts useful information from images, videos and visual inputs to perform appropriate action(s).
2. **Natural Language Processing (NLP):** NLP is used to give voice commands to AI robots to create a strong human-robot interaction. NLP helps the robot to understand and reproduce human language. Besides recognizing human language, robots can learn the accent, and predict how humans speak.
3. **Edge Computing:** Edge computing in robots helps in robot integration, testing, design and simulation to provide better data management, lower connectivity cost, better security practices, more reliable and uninterrupted connection.
4. **Complex Event Process:** An event is said to occur when there is a change of state and one or more events combine to define a complex event. Correspondingly, complex event processing (CEP) helps us to understand the processing of multiple events in real time. The term 'complex event process' is popularly used in various industries including healthcare, finance, security, marketing, detecting credit card fraud, stock marketing, etc. For example, the deployment of an airbag in a car is a complex event that is based on real-time data coming from multiple sensors.
5. **Transfer Learning and AI:** Transfer learning solves a problem with the help of an already solved problem. So, the knowledge gained from solving one problem is utilized to solve another problem that is related to the problem to be solved. For example, the model used to recognize a square shape can also be used to identify a circular shape. Since transfer learning reuses the pre-trained model for a related problem, the actual learning time for the new problem to be solved is relatively less and thus provides a cheaper solution. In robotics, transfer learning trains one machine using other machines.

6. Reinforcement Learning: Reinforcement learning focuses on learning based on feedback. In this technique, an AI agent learns and explores the environment, performs actions and automatically learns from experience or feedback obtained for each action. Reinforcement learning makes the agent learn to behave optimally through hit-and-trial method while interacting with the environment. It is mainly used to deduce decisions to achieve goals in an uncertain and potentially complex environment.

In the context of robotics, robots explore the environment to learn how to work optimally through hit and trial method. For every correct action, the robot gets a reward and a punishment for an incorrect action. In this way, reinforcement learning provides a framework to design and simulate sophisticated and hard-to-engineer robotic behaviours.

7. Affective computing: Affective computing focuses on developing systems that can identify, interpret, process and simulate human emotions. Therefore, this study aims to imbibe emotional intelligence in robots so that they are able to observe interpret and express themselves emotionally.

8. Mixed Reality: Mixed reality is an emerging field that focuses on programming by demonstration (PbD). PbD creates a prototype using a combination of physical and virtual objects.

8.3.7 Planning and Navigation

In this section, we will read about mobile robots that are critical to robust mobility. For this we must have some knowledge about the kinematics of locomotion, sensors for determining the robot's environmental context and techniques for localizing with respect to its map. Here, we will talk about cognitive level of robots.

Cognition refers to the capability of a system to make purposeful decision and execute them to achieve its highest-order goals. In the context of a mobile robot, cognition refers to robust mobility. Robot navigation aims to make the robot reach its goal position efficiently and reliably based on its knowledge about its environment, the goal position and values of the sensor.

Although many different approaches have been proposed for navigating a robot, there are strong similarities between all of them. The underlying difference exists in the manner in which they decompose the problem into smaller sub-units. Given a map and a goal location, path planning identifies a trajectory that will help the robot to reach the final destination.

8.3.7.1 Competencies for Planning—Planning and Reacting

Path planning is a strategic problem-solving competency (capability), as the robot must make vital decisions to achieve its goals. The second competency is tactical extreme to avoid all obstacles and collisions by modulating the trajectory of the robot. Given readings of real-time sensor readings, several approaches are proposed to avoid obstacles. We will discover a few of them.

While navigating, a robot plans and reacts. Without reacting, the planning effort(s) will be of no use as the robot never be able to reach its goal. Correspondingly, without planning, the robot will never reach its goal.

Suppose a robot R at time i in the initial belief state b_i has a map M_i to reach position p while satisfying certain some temporal constraints $\text{loc}_g(R) = p; (g \leq n)$ (like before timestep n).

Although the robot's movements have to be physical, the robot can only sense its belief state, not its physical location. We will, therefore, map the goal of reaching location p to reaching a belief state b_g . We need a plan q that specifies one or more trajectories from b_i to b_g .

However, the problem arises when either the robot's position is not quite consistent with b_i or M_i is incorrect and/or incomplete. Furthermore, the real-world environment is dynamic. Even if M_i is correct at time i , M may change over time as the robot must incorporate new information gained during plan execution.

Over a period of time, the environment changes and the robot's sensors gather new information. In this scenario, reacting becomes more important. Reacting modulates the robot's behaviour to update the planned

upon trajectory so that the robot still reaches its goal position. Moreover, unanticipated new information also requires changes to the robot's strategic plans. Therefore, in the specified time limits, the plan should consider every new piece of information in real time to produce a new plan to react to the new information appropriately. This theoretical extreme, the point at which the concept of planning and reacting merge, is called integrated planning and execution.

8.3.7.2 Key Terms in Trajectory Planning

Trajectory Planning

Trajectory planning refers to moving from point A to point B while avoiding collisions over time in a 2D or 3D space (refer **Fig. 8.12**). It is an important concept in designing autonomous vehicles. Trajectory planning also known as motion planning is mistakenly referred to as path planning. However, it is different from path planning as it is parametrized by time. Besides path planning, trajectory planning also incorporates planning based on velocity, time, and kinematics.

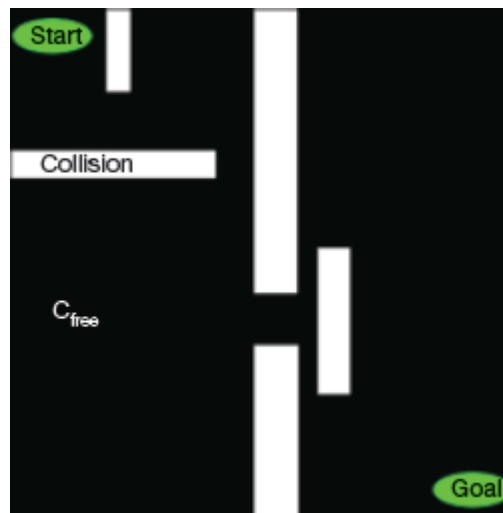


FIGURE 8.12 Trajectory planning

Configuration Space

A configuration is the pose of a robot describing its position. Correspondingly, Configuration Space C , is the set of all configurations. For example, in a 2D space, a robot's configuration is given by coordinates

(x, y) and angle θ . Similarly, in 3D space, a robot's configuration is given by coordinates (x, y, z) and angles (α, β, γ) .

The goal of path planning is to find a path from the initial position to the goal position in the physical space that avoids all collisions with the obstacles. This problem becomes particularly more difficult as k grows large. If we define the configuration space obstacle O as the subspace of C , the free space in which the robot can move safely can be computed as, $F = C - O$.

Figure 8.13 shows a picture of the physical space and a 2D configuration space for a planar robot arm with two links. The robot's goal is to move its end effector from position c_1 to c_2 .

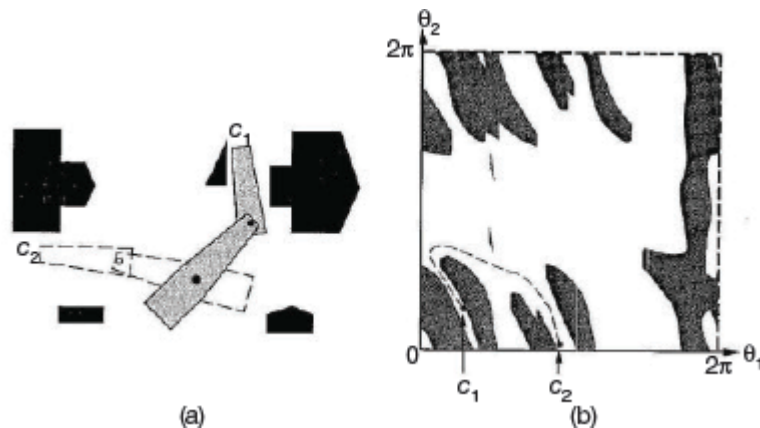


FIGURE 8.13 Physical space and a 2D configuration space

Free Space

Free space C_{free} is the set of all configurations that are collision-free. The robot should be programmed to use kinematics and collision detection from sensors to ensure if a given configuration is a collision free or not.

Target Space

Target space is a linear subspace of free space and consists of space in which we want the robot to explore. In global motion planning, target space is observable by the robot's sensors. But in local motion planning, some states in target space are not observable by robot's sensors. To solve

a problem, robots assume several observable virtual target spaces (around itself). The virtual target space is often referred to as sub-goal.

Degrees of Freedom

It specifies modes in which a mechanical device or system can move. This means that the number of degrees of freedom is equal to the total number of independent displacements or aspects of motion. For example, in a robotic arm, shoulder and wrist can move up or down, left or right but elbow can move up or down. Wrist and shoulder can also be rotated (or rolled).

Therefore, such a robot arm has five to seven degrees of freedom. And if the robot has two arms, then total number of degrees of freedom is doubled.

Multi-legged mobile robots can have more than 20 degrees of freedom. For example, Project Nao, which looks superficially like a large space-age doll, has 25 degrees of freedom.

8.3.7.3 Problem Constraints in Trajectory Planning

Trajectory planning suffers from two main problems which are discussed below.

Holonomicity

This gives the relationship between the controllable degrees of freedom and total degrees of freedom of the robot. A robot is said to be holonomic if the number of controllable degrees of freedom is greater than or equal to the total degrees of freedom. Holonomic robots are easy to work with as they facilitate many movements easier to make. Moreover, returning to a past pose is much easier.

For example, if a self-driving car is designed as a non-holonomic robot, then it will have no way to move laterally, thereby making its certain movements (like parallel parking) difficult. In contrast to this, a holo-

onomic vehicle would be one with mecanum wheels, such as the new Segway RMP.

Dynamic Environments

In dynamic environments, like the real world, objects that may cause collision are not stationary. This makes trajectory planning more difficult as objects are moving with respect to time. Many-a-time, it may not be possible for a robot to move backward in time. Moreover, many choices are completely irreversible due to terrain, such as moving off of a cliff.

8.3.7.4 Planning Algorithms

In this section, we will discuss three types of planning algorithms.

1. **Artificial Potential Field:** The algorithm places values over the map with the goal having the lowest or highest value that either increases or decreases depending on the distance from the goal (refer **Fig. 8.14**).

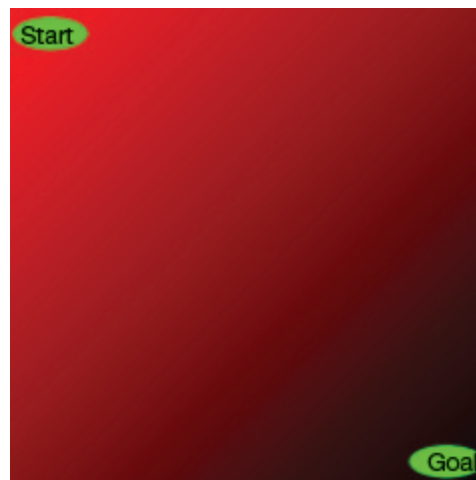


FIGURE 8.14 Artificial Potential Field

Obstacles defined can have an incredibly high or low value. The robot is programmed to move to the lowest or (highest) potential value adjacent to it, which should lead it to the goal. However, this algorithm often gets trapped in local minima.

2. **Sampling-based Planning:** Roadmap method is an example of sampling-based planning method. The algorithm first selects a sample of N configurations in C as milestones. Then, a line PQ is formed between all milestones as long as the line PQ is completely in C_{free} . Following this, any graph search algorithms can be used to find a path from start

to the goal. Though a large number of N increases computation time, it results in better solutions.

3. **Grid-based Planning:** This algorithm overlays a grid on the map (as shown in **Fig. 8.15**) and then ensures that every configuration corresponds with a grid pixel. The robot can move from one grid pixel to another adjacent grid pixel if that grid pixel is in C_{free} .



FIGURE 8.15 Grid-based Planning

Search algorithm like A^* is used to find a path to reach the goal position from the starting position. Although a lower resolution grid having bigger pixels will make the search procedure execute faster, it may however miss paths through narrow spaces of C_{free} . Moreover, as the resolution of the grid increases, memory usage also grows exponentially. In such a scenario and especially in large areas, another path planning algorithm may be necessary.

4. **Reward-based Planning:** Algorithms based on research-based planning assume that robot in each state (position) may select any action (regarding motion). However, the result of each action is not definite. In such a case, outcomes or displacement, in this context, are partly random and partly under the robot's control. On reaching the goal, the robot gets a positive reward and a negative reward on colliding with an obstacle.

Research-based planning algorithms, therefore, aim to find a path giving maximum future rewards. Markov decision processes (MDPs) is a popular technique used in many of reward-based algorithms. Though this technique generates optimal path, this path may not be smooth as it limits the robot to choose from a finite set of actions. To overcome

this drawback, usually Fuzzy Markov decision processes (FDMPs), an extension of MDPs, is used to generate a smooth path with using a fuzzy inference system.

5. **Road-map Path Planning:** Road-map path planning algorithms capture the connectivity of the robot's free space in a network of one-dimensional curves or lines, called road-maps. Once a road-map is constructed, it is then used to plan a robot's motion. The algorithm reduces the problem of path planning to connecting the initial and goal positions of the robot to the road network and then searching the roads from the initial position to the goal position of the robot. This means that, road-map uses the obstacle's geometry to decompose the robot's configuration space.

The challenge is to construct a set of roads that together helps the robot to move around anywhere in its free space, while minimizing the number of total roads. In this section, we will discuss two approaches for road mapping—visibility graph and Voronoi diagram. While visibility graph finds paths that have minimum lengths by bringing the roads closer to the obstacles, Voronoi diagram, on the other hand, keeps roads as far as possible from the obstacles.

8.3.7.5 Visibility Graph

The visibility graph for a polygonal configuration space C includes all edges (including initial and final positions) joining every pair of vertices that can see each other. The unobstructed roads depicted as straight lines in the graph are the shortest distances between two vertices. The job of the path planning algorithm is to find the shortest path from the initial position to the goal position along the roads drawn in the visibility graph (see **Fig. 8.16**).

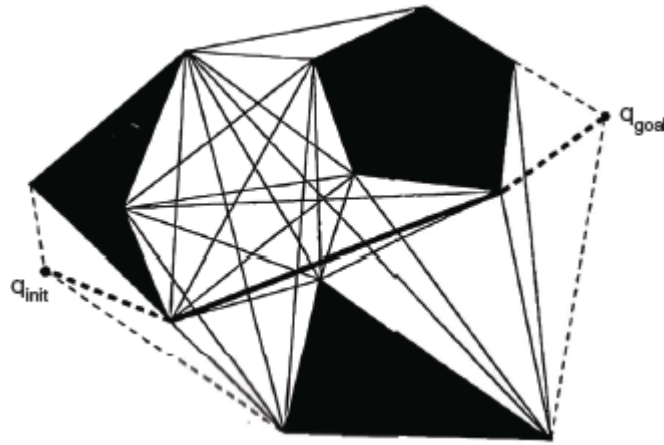


FIGURE 8.16 Visibility Graph

The visibility graph path planning algorithm is very simple and straightforward. It can be readily used when objects in the environment are described as polygons either in continuous or in discrete space. However, there are two important limitations of this algorithm.

First, with increase in the number of obstacle polygons, the number of edges and vertices increases. Therefore, this algorithm works well in sparse environments, but is very slow and inefficient when used in densely populated environments.

Second, the solution paths found by visibility graph planning algorithm takes the robot as close as possible to obstacles on the way to the goal.

Though this algorithm is optimal in terms of the length of the solution path, it does not keep the robot away from obstacles. This compromises safety. A potential solution can be either to grow the obstacle's size significantly so that it becomes more than the robot's radius, or modify the solution path after path planning to distance the path from obstacles. However, such fixes compromise the optimal-length results of the algorithm.

8.3.7.6 Voronoi Diagram

A Voronoi diagram maximizes the distance between the robot and obstacles in the map. For each point in the free space, distance from that point to the nearest obstacle is calculated and plotted as a height coming out of the page (refer to **Fig. 8.17**). As the robot moves away from the obstacle,

the height increases. Sharp ridges can be seen in the Voronoi diagram at points that are equidistant from two or more obstacles. Edges are formed by these sharp ridge points. When obstacles in the configuration space are polygons, the Voronoi diagram has straight and parabolic segments as shown in **Fig. 8.18**.

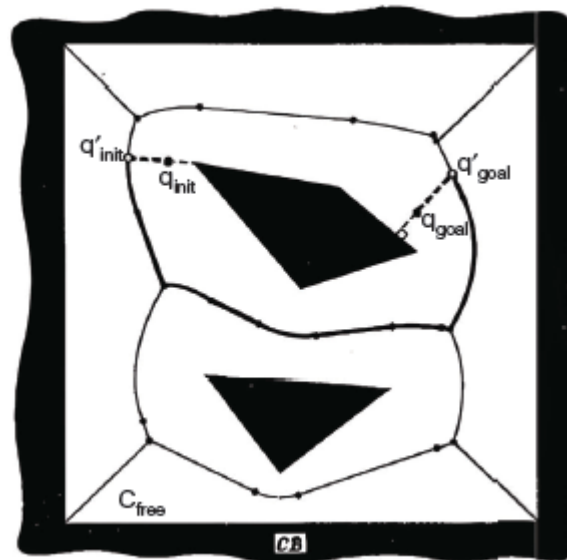


FIGURE 8.17 Voronoi Diagram

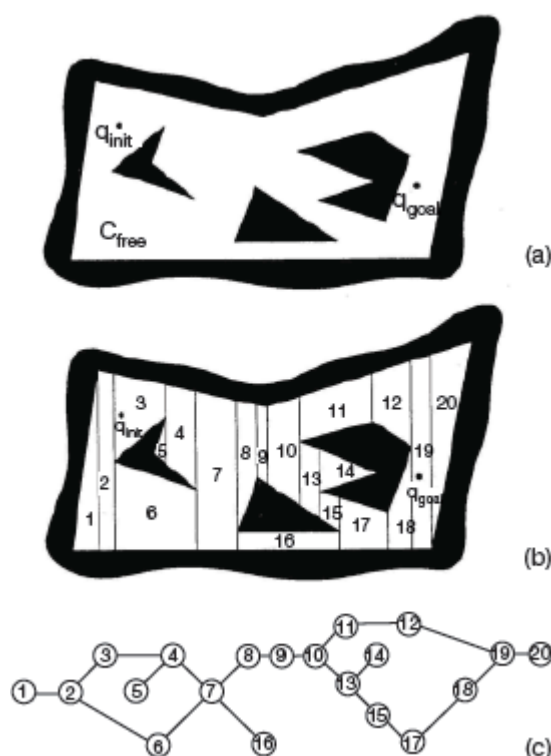


FIGURE 8.18 Forming Edges and making graph from the given Voronoi Diagram

The path in the Voronoi diagram may not be optimal in terms of total path length. Another limitation of this diagram is limited range localization sensors. Since the distance between the robot and obstacle is maximized, any short-range sensor on the robot may not be able to sense its surroundings. Hence, the chosen path may be quite poor from a localization point of view.

However, the main reason of popularity of the Voronoi diagram method is its executability. Given a planned path via Voronoi diagram planning, a robot with range sensors can follow a Voronoi edge in the physical world using simple control rules that match those used to create the Voronoi diagram. Voronoi Diagram can also be used to conduct automatic mapping of an environment by finding and moving on unknown Voronoi edges and creating a consistent Voronoi map of the environment.

8.3.7.7 Cell Decomposition Path Planning

The cell decomposition path planning aims to distinguish between geometric areas, or cells, that are free and areas that are occupied by objects. The following must be kept in mind.

1. The graph is divided into simple, connected regions called ‘cells’.
2. A connectivity graph is constructed by identifying opens cells that are adjacent.
3. Identify the cells in which the initial and goal configurations lie. Select a path in the connectivity graph that joins the initial and the goal cell.
4. From the cells identified, compute a path within each cell following motions and movements along straight lines.

The most important concept in cell decomposition method is the placement of the boundaries between cells. There are two methods which perform the task.

1. *Exact cell decomposition*: In this technique, the boundaries are placed as a function of the structure of the environment in such a way that the decomposition is lossless.
2. *Approximate cell decomposition*: This approach performs decomposition as an approximation of the actual map.

8.3.7.8 Exact Cell Decomposition

In **Fig. 8.19**, exact cell decomposition has been used. Here, the boundary of cells is based on geometric criticality. So, the resulting cells are either completely free or completely occupied resulting in a complete path. In this method, the robot's ability to traverse from each free cell to adjacent free cells is more important than the position of the robot within each cell of free space.

Advantage

In environments that are extremely sparse, the number of cells will be small.

Disadvantages

1. The number of cells and computational efficiency of the overall path planning algorithm depends upon the density and complexity of objects in the environment.
2. Due to complexities in implementation, the exact cell decomposition technique is less frequently used in mobile robot applications.

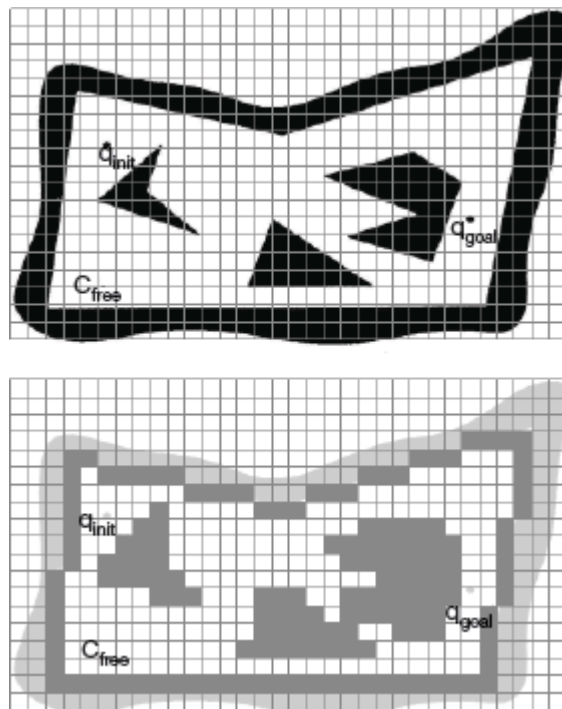


FIGURE 8.19 Exact Cell Decomposition

8.3.7.9 Approximate Cell Decomposition

Approximate cell decomposition is a popular technique for mobile robot path planning. It uses grid-based environmental representations that are themselves fixed gridsize decompositions. They are quite similar to an approximate cell decomposition of the environment.

We can see that in **Fig. 8.20**, the cell size is not dependent on the particular objects in an environment resulting in loss of narrow passageways due to the inexact nature of the tessellation. However, this is rarely a problem as the cell size used is very small, usually 5 cm on each side.

The main advantage of fixed-size cell decomposition is the low computational complexity of path planning. In **Fig. 8.20**, we see that the free space is externally bounded by a rectangle which is further decomposed into four identical rectangles. The rectangle is not decomposed further only in two conditions.

1. First, the interior of a rectangle lies completely in free space.
2. Second, the interior of a rectangle lies completely in the configuration space obstacle.

If the two conditions are not satisfied, then the rectangle is recursively decomposed into four rectangles until one of them is met. Note that the white cells lie outside the obstacles, the black inside and the grey are part of both the regions.

This technique is an efficient and simple strategy for finding routes in fixed-size cell arrays. The algorithm starts from the goal position marking for each cell its distance to the goal cell until the initial robot position is reached. At this point, distance to the goal position and a specific solution trajectory is known by linking together cells that are adjacent and closer to the goal.

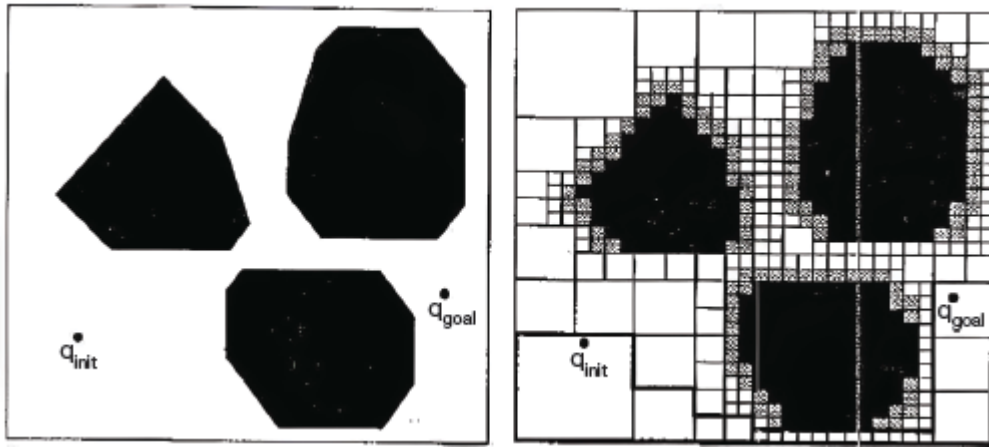


FIGURE 8.20 Approximate Cell Decomposition

If the entire array is in memory, then each cell is visited only once to deduce the shortest path from the initial position to the goal position. So, the search is linear. Therefore, the complexity of this algorithm neither depends on the sparseness and density of the environment, nor on the complexity of the objects' shapes in the environment.

The Cye robot is an example of a commercially available robot that plans path in a 2D space with 2 cm fixed-cell decomposition of the environment. Unlike exact cell decomposition method, approximate cell decomposition method does not guarantee completeness but it is mathematically less involving and easier to implement.

8.3.7.10 Potential Field Path Planning

Potential field path planning algorithm creates a field across the robot's map that directs the robot to the goal position. The potential field technique considers robot as a point under the influence of an artificial potential field $U(q)$.

Like a ball roll downhill under a gravitational field, the robot moves by following the potential field. The goal acts as an attractive force on the robot and the obstacles act as peaks, or repulsive forces. The superposition of all forces is applied to the robot. The artificial potential field thus created helps the robot to avoid known obstacles and smoothly move towards the goal. Thus, this concept goes beyond path planning.

Basically, the resulting field is a control law for the robot. A robot can always decide its next action, assuming that it can localize its position based on the map and potential field.

Knowing that the robot is attracted towards the goal and repulsed by the known obstacle, we need to update the potential field as soon as a new obstacle appear during robot motion. Considering robot as a point, the robot's orientation θ is neglected and the resulting potential field is only two dimensional (x,y).

Assuming a differentiable potential field function $U(q)$, related artificial force $F(q)$ acting at the position (x,y) is calculated as follows:

$$F(q) = -\Delta U(q),$$

where $\Delta U(q)$ is the gradient vector of U at position q .

$$\nabla U = \begin{bmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \end{bmatrix}$$

The potential field acting on the robot is then computed as the sum of the attractive field of the goal and the repulsive fields of the obstacles. Thus,

$$U(q) = U_{att}(q) + U_{rep}(q)$$

Correspondingly, the net force after considering attracting and repulsing forces is calculated as,

$$\begin{aligned} F(q) &= F_{att}(q) - F_{rep}(q) \\ &= -\nabla U_{att}(q) - \nabla U_{rep}(q) \end{aligned}$$

8.4 Recent Trends in AI

In this section, we will read about some latest trends in AI. Extensive research is being done in the terms discussed here.

8.4.1 Collaborative Systems

These days, a lot of research is being done to develop collaborative systems in which humans and machines (with AI) can become effective partners by overcoming each other's limitations and augmenting human abilities. There are many examples of machine-human collaboration as follows.

1. The computer game Foldit uses machine-human collaboration fold simulated proteins. The folding task is performed to understand how real-world proteins involved in the causes of human disease are formed. In the game, both the player as well as AI performs operations. The AI is put to use where it excels and humans' intuition and imagination is applied where they exceed machines.
2. Two amateur chess players and an AI system worked collaboratively and won against a field of supercomputers and grandmasters.
3. In 2014, Knowledge Ventures, a Japanese venture capital company, elected an AI system to its board of directors, to realize the benefits of machine-human collaboration at the highest levels of business.
4. In the military, UAV drones are being used in which a machine takes the front lines with human provides virtual support.
5. The Talos suit in the US military augments a front-line human with a power exoskeleton. The exoskeleton's integrated heating cooling, vital monitoring and heads-up display (HUD) collaborate with humans much similar to that seen in the Ironman movies.

While many people believe that humans and machines can be good partners, others, however, fear that relying too much on AI will result in a massive job loss and making common man people dumb and lazy.

However, the good news is that research involving 1,500 companies says that many companies that have used AI to automate processes and displaced employees will see only short-term productivity gains. Better performance is expected only when humans and machines work together to actively enhance each other's complementary strengths- leadership, teamwork, creativity, social skills of humans and speed, scalability and quantitative capabilities of the machines.

AI can boost our analytic and decision-making abilities and heighten creativity.

Humans Assisting Machines

In a human-machine collaborative system, humans need to perform three important roles. First, they must *train* machines to perform a particular task; *explain* the outcome of that task (especially when results are counterintuitive or controversial) and *sustain* the responsible use of machines so that machines do not harm mankind. For example, the European Union's new General Data Protection Regulation (GDPR) empowers consumers with the right to get an explanation for any algorithm-based decision (like reason for acceptance/rejection of a loan). This feature will not be a big step in delivering responsible AI but will also create about 75,000 new jobs to administer the GDPR requirements.

Last but not the least, companies also need employees who can continuously monitor to ensure that AI systems are functioning properly, safely and responsibly. For example, a self-driving car must not hit objects and any living being. Similarly, tech companies including Apple use AI to collect personal details about users to improve the user experience, but collecting too much data can compromise privacy, anger customers and even go against the law. Therefore, employees make sure that AI use data only for analysis and the privacy of individual users is not violated.

Points to remember:

Humans must also train AI systems to interact with humans in the best possible way. For example, all virtual assistants were given extensive training to develop to react as a confident, caring and helpful personality without being bossy. Such training requires inputs from a team of experts (even including a poet, a novelist and a playwright).

These days, AI assistants are also being trained to display sympathetic feelings. For example, the start-up Koko, an offshoot of the MIT Media Lab, is developing an empathetic assistant. For example, if a user had a bad day, then Koko would feel sorry for it, ask for more information and offer advice to help the person see his issues in a different light.

8.4.2 Machines Assisting Humans

Smart machines are helping humans enhance their abilities in three ways—amplify, interact and embody. AI systems boost human’s creativity, analytic and decision-making abilities by providing the right information at the right time. They can also interact with company’s employees and customers in more effective ways. Even at homes, digital assistants are assisting humans to perform their instructions.

SEB, a Swedish bank, uses Aida, a virtual assistant to interact with millions of customers and assist them in opening a bank account, making cross-border payments and interpreting their voice tone (satisfied or frustrated) to provide better service later. In 30% of the cases when the system is not able to resolve an issue, it directs the call to a human customer-service representative and then monitors that interaction to learn how to resolve similar problems in the future.

Case study Question:

Collect information about how Autodesk’s Dreamcatcher AI enhances the imagination of even exceptional designers.

Embodying

AI systems are embodied in machines (like robots) to augment the capabilities of a human worker. Using their sophisticated sensors, motors and actuators, these machines can recognize people/objects and work safely in factories, warehouses and laboratories. Such robots also known as cobots handle repetitive actions that require heavy lifting or some dangerous tasks. In such a collaborative environment, humans are free to do other complementary tasks that require dexterity.

For example, Hyundai has extended the concept of cobot with exoskeletons.

Exoskeletons are wearable robotic devices, which adapt to an industrial worker’s location in real time to enable him to perform his job with superhuman endurance and strength. Similarly, Mercedes-Benz, uses cobot

arms to customize cars as per user's expectations according to the real-time choices consumers make at dealer's showroom.

8.4.3 Algorithmic Game Theory and Computational Social Choice

These days, (especially enhanced due to pandemic) with increased Internet connectivity and speed, more people prefer to play games like chess, checker, poker and solitaire. These games are played using a clear set of defined rules. So, there was a need to embed algorithms in machines that allowed machines to play ethically according to pre-defined rules. This led to the onset of game theory. In a multi-agent situation (when more than one person is involved in solving a logical problem), game theory is used to choose from a set of options knowing that *our choice would affect the choices of the opponent* and their decision would affect our choices.

Von Newman is credited for the invention of game theory.

We can understand the concept of game theory by studying the five categories.

1. **Cooperative vs Non-cooperative Games:** In cooperative games, participants can establish alliances to increase their chances of winning the game. Correspondingly, in a non-cooperative game, participants cannot form any alliance.
2. **Symmetric vs Asymmetric Games:** In a symmetric game, each participant has the same goal(s). However, their strategy to win may be different. In contrast to this, in an asymmetric game, the participants have different (and at times conflicting) goals.
3. **Perfect vs Imperfect Information Games:** In perfect information games like chess, all the players can see what moves the other player(s) are making. But, in an imperfect information game (like card game), other players' moves are hidden.
4. **Simultaneous vs Sequential Games:** In simultaneous games, multiple players can take actions concurrently. In a sequential game like in

a board game, each player is aware of the other players' previous actions.

5. **Zero-sum vs Non-zero Sum Games:** In zero sum games, a player gains something that causes a loss to other players but in a non-zero sum game, multiple players get the advantages when another player gains.

When using AI techniques in game theory, two concepts are very important—Nash equilibrium and inverse game theory.

Nash Equilibrium

The Nash equilibrium is a condition in which all the players playing the game accept that there is no better solution to the game than the current situation. No player would gain anything after changing their current strategy. So, any new decision made by the players will not result in anything better.

Example: A typical example of Nash equilibrium is the Prisoner's Dilemma. If two criminals get arrested and are kept in confinement without being allowed to communicate with each other, then look at **Fig. 8.21** to understand the possibilities.

- If either of the two prisoners confess that the other committed a crime, the first will be set free and the other will be sent to prison for 10 years.
- If none of them confess, then they will be sent to prison only for 1 year.
- If both confess, then both will be sent to prison for 5 years.

		PLAYER B	
		COOPERATE	DEFECT
PLAYER A	COOPERATE	A: 1 year jail B: 1 year jail	A: 10 years jail B: 0 years jail
	DEFECT	A: 0 years jail B: 10 years jail	A: 5 years jail B: 5 years jail

FIGURE 8.21 Nash equilibrium

In this case, the Nash equilibrium is reached when both criminals betray each other.

Inverse Game Theory

Game theory is used to understand the dynamics of a game to optimize the possible outcome of its players. In contrast to this, inverse game theory designs a game based on players' strategies and aims. It is key to designing AI agents environments.

Practical Examples

A generative adversarial network (**GAN**) is a machine learning (ML) model in which two neural networks compete with each other to become more accurate in their predictions.

Nash equilibrium can be attained easily in symmetric than asymmetric games. But, in real world, asymmetric games are more common.

GANs contains two models—a *generative model* and a *discriminative model* (**Fig. 8.22**).

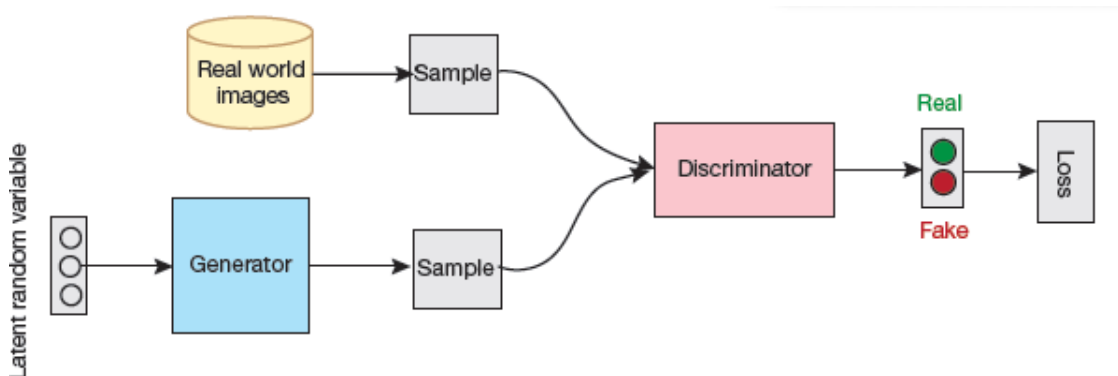


FIGURE 8.22 GAN Architecture [3]

Generative models accept some features as input to examine their distributions and understand how they have been produced.

Discriminative models take the input features to predict the class to which our sample belong.

For example, in a GAN, the generative model uses input features to create new samples that resemble the main features of the original samples.

The original samples and the newly generated samples are then passed to the discriminative model to identify which samples are genuine and which are fake. A common application of GANs is to generate images (as given in **Fig. 8.23**) and then distinguish between real and fake ones.



Credit: doglikehorse / Shutterstock

FIGURE 8.23 Generate Images using GAN Models

Even in game playing, GANs are used extensively. The players (the two models) challenge each other. While one player creates fake samples to confuse the other, the other tries to identify the right samples. The game is played iteratively until Nash equilibrium is reached. In each iteration, the learning parameters are updated to minimize loss.

8.4.4 Multi-Agents Reinforcement Learning (MARL)

Reinforcement learning (RL) is used to make an agent (model) learn through the interaction with its environment. RL was initially developed based on Markov decision processes. This means that the agent placed in a stochastic stationary environment tries to learn a policy through a reward/punishment mechanism. The agent will finally converge to a satisfactory policy.

However, this does not hold true when multiple agents are placed in the same environment. Therefore, instead of depending only on learning through interaction between the agent and the environment, now it is also dependent on the interaction between agents.

Modelling systems with multiple agents is not a trivial task as increase in the number of agents exponentially increases the number of possible ways in which they can interact with each other. Therefore, modelling multi-agents reinforcement learning models can be done using mean field scenarios (MFS) to reduce its complexity by making the assumption a priori that all agents have similar reward functions.

Example: To improve the traffic flow in a city, AI-powered self-driving cars can perfectly interact with the external environment. But things can get complicated if these cars think as a group. In such a case, a car might get in conflict with another when both of them find it most convenient to follow a certain route.

The above situation can be easily modelled using game theory. Our cars would be different players and the Nash equilibrium would be the point between the collaboration between the different cars.

8.4.5 Neuromorphic Computing

Neuromorphic computing is a technique in which a computer (both hardware and software) is modelled based on the architecture of the human brain and its nervous system. It is a science that draws concepts from multiple disciplines including computer science, biology, mathematics, electronic engineering and physics.

The main aim of neuromorphic computing is to create artificial neural systems inspired by biological structures. Such a structure would enable the machine to behave like a human brain in the following ways.

First, to learn, retain information and even make logical decisions as the human brain makes.

Second, to acquire new information and work like a human brain.

Thus, a computer needs to be transformed into a cognition machine.

Did you know that our brain uses roughly 20 watts of power on average, which is about half that of a standard laptop?

How Does Neuromorphic Computing Work?

Traditional neural network and machine learning algorithms are best suited for existing algorithms. They either provide fast computation *or* focus on using low power and there is always a trade-off between the what to achieve. Neuromorphic systems perform fast computation while consuming less power consumption. Additionally, they also resemble human brain as they have the following features:

1. They are massively parallel, to handle multiple tasks at once.
2. They are event-driven, as they respond to events dynamically based on changing environmental conditions.
3. They are economical in terms of power used.
4. They are flexible as they are highly adaptable and are able to generalize.

5. They are strong and fault-tolerant, since they produce results even when some component(s) have failed.

Neuromorphic computing achieves this brain-like function and efficiency by building artificial neural systems that implement ‘neurons’ (nodes that process information) and ‘synapses’ (the connections between those nodes) to transfer electrical signals using analog circuitry. This structure controls the amount of electricity being passed from one node to another to mimic the varying degrees of strength that usually occurs in a human brain. Such a system of neurons and synapses that transmit these electric pulses is known as a spiking neural network (SNN). This facility was not available in traditional neural networks (that uses digital signals).

Neuromorphic systems deviate from the Von Neumann architecture and introduce a new chip architecture that collocates memory and processing together on each individual neuron. Hence, there is no need to have separate memory unit (MU), central processing unit (CPU) and data paths. Having separate parts requires information to be repeatedly moved back and forth between different components to complete a given task, creating a bottleneck for time and energy efficiency. This is known as the von Neumann bottleneck.

By bringing memory on the neuromorphic chip, information can be processed much more efficiently, thereby making the chip more powerful and efficient.

Neuromorphic Computing and Artificial General Intelligence (AGI)

We have seen that the term ‘artificial general intelligence’ (AGI) exhibits intelligence equal to that of humans has not yet been achieved and may not even be reached with the traditional computers. But with neuromorphic computing, researchers are feeling optimistic about it.

Example: The Human Brain Project, featuring the neuromorphic super-computer SpiNNaker produces a functioning simulation of the human brain and is one of the hottest research projects in AGI.

A machine is said to achieve AGI when the machine can do the following:

- Reason and make judgments in an uncertain situation
- Plan
- Learn
- Communicate using natural language
- Represent knowledge
- Integrate the above skills to fulfil a goal

An AGI may even have the ability to imagine, gain from experience and be self-aware. To confirm whether a system has developed AGI or not, it can be made to pass the Turing Test or the Robot College Student Test, in which a machine enrolls in classes and obtains a degree like a human would.

Debatable Issue

Even if a machine develops human intelligence, then a question rises that whether it should be handled ethically and legally. While some researchers argue that it should be treated as a nonhuman animal in the eyes of the law, others say that still a lot has to be done to inculcate consciousness in machines.

Comparing Human Brain with Neuromorphic Computing

Brains need very less energy than most supercomputers. While a human brain uses about 20 watts, the Fugaku supercomputer needs 28 megawatts (about 0.00007% of Fugaku's) power supply.

Supercomputers need elaborate cooling systems but our brain sits at 37°C in between bones.

Supercomputers can make complex millions and billions of complex calculations in second(s) but the human brain is very adaptable. A single brain can write poetry, identify a familiar face out of a crowd, drive a car, learn a new language, instantly take decisions, respond to events and do much more.

While traditional computers designed on von Neumann architecture are largely serial, brains use massively parallel computing. Brains are also

more fault-tolerant than computers. Therefore, harnessing techniques used by a human brain will be the key to making more powerful neuromorphic computers in the future.

So How Can You Make a Computer That Works Like the Human Brain?

To understand how neuromorphic computers work, let us first recapitulate how the brain works.

When we feel hot or cold, neurons (a type of nerve cell) carry messages to the brain. To transfer messages, several neurons release chemicals across a gap called a synapse. This happens until the message reaches the brain. Once the brain identifies the sensation, it sends an appropriate message to the point which felt the sensation using neurons.

An action potential is triggered either through sending multiple inputs at once (spatial), or by building multiple inputs over time (temporal). The brain transfers information quickly and efficiently through the huge interconnectivity of synapses where one synapse might be connected to 10,000 other synapses.

Neuromorphic computers work like human brains using spiking neural networks (SNN). While transistors used in a conventional computing can be either on (1) or off (0), SNN can produce more than one outputs and transfers information in temporal and spatial way as the human brain does.

Neuromorphic systems can be either digital or analogue. Synapses are controlled either by software or memristors. Memristors can store a range of values instead of storing just one and zero. Like the human brain, they can vary the strength of a connection between two synapses to allow the brain-based systems to learn.

Uses of Neuromorphic Systems

As of now, to perform heavy computational tasks, devices like smartphones hand off processing to a cloud-based system. The cloud processes the query and sends the output to the device. This scenario will change

with neuromorphic systems. In such a system, instead of passing queries and responses back and forth, computations would be done within the device itself.

The current generation AI is rules-based. It is extensively trained on datasets to generate a particular outcome. However, the human brain does not work like this. Researchers are trying hard to make the next generation of artificial intelligence deal with more brain-like problems. For example, constraint satisfaction, in which a system has to find the optimum solution to a problem with a lot of restrictions.

Neuromorphic systems also perform well with noisy and uncertain data.

Examples of Neuromorphic Computer Systems Used Today

64 Intel's neuromorphic chips named Loihi are used to make an 8 million synapse system called Pohoiki Beach, consisting of 8 million neurons (may soon reach 100 million neurons in the near future). They are being used to create artificial skin and also develop powered prosthetic limbs.

IBM's neuromorphic system, TrueNorth, launched in 2014, consists of 64 million neurons and 16 billion synapses. IBM in a partnership with the US Air Force Research Laboratory is trying to create a 'neuromorphic super-computer' known as Blue Raven for creating smarter, lighter, less energy-demanding drones.

The EU-funded Human Brain Project (HBP), a 10-year project that started in 2013, has led to two major neuromorphic initiatives, SpiNNaker and BrainScaleS. In 2018, the largest neuromorphic supercomputer of that time, a million-core SpiNNaker system went live. It is hoped that this model would soon scale up to one million neurons. BrainScaleS also has a similar goal and its architecture is now on its second generation, BrainScaleS-2.

Challenges to Using Neuromorphic Systems

Shifting from Von Neumann to neuromorphic computing is not that easy and straight-forward. For example, when analysing visual input, tradi-

tional systems treat them as a series of individual frames, but a neuromorphic processor would encode this information as changes in a visual field over time.

New programming languages need to be developed to utilize the change in the underlying architecture. New kind of memory, storage and sensor technology need to be created to take full advantage of neuromorphic devices. This may even change the entire process of creating and integrating hardware and software components.



Lobe is a free, easy-to-use Microsoft desktop application that allow users to build, manage and use custom machine learning models. Once created, these models can be directly uploaded to AI Builder for use with Power Apps and Power Automate.

Step 1: Download and Install Microsoft Lobe.

Step 2: Add and Label Images in Lobe. For this, click the New Project button when Lobe opens. Enter a name for your project. Then add and label some images to create machine learning model.

You can either import images from an existing dataset or capture them from a webcam. While adding images, make sure that you add at least five images of the same subject. Ideally, these should have variations in background, lighting or position to help Lobe identify the important parts.

Now, enter a descriptive label for the first image. Lobe will save this label, so that it can be used for other images in the dataset.

Next, import another set of images of a different but related subject. Label the first image with a descriptive tag, and then use that label to the other images.

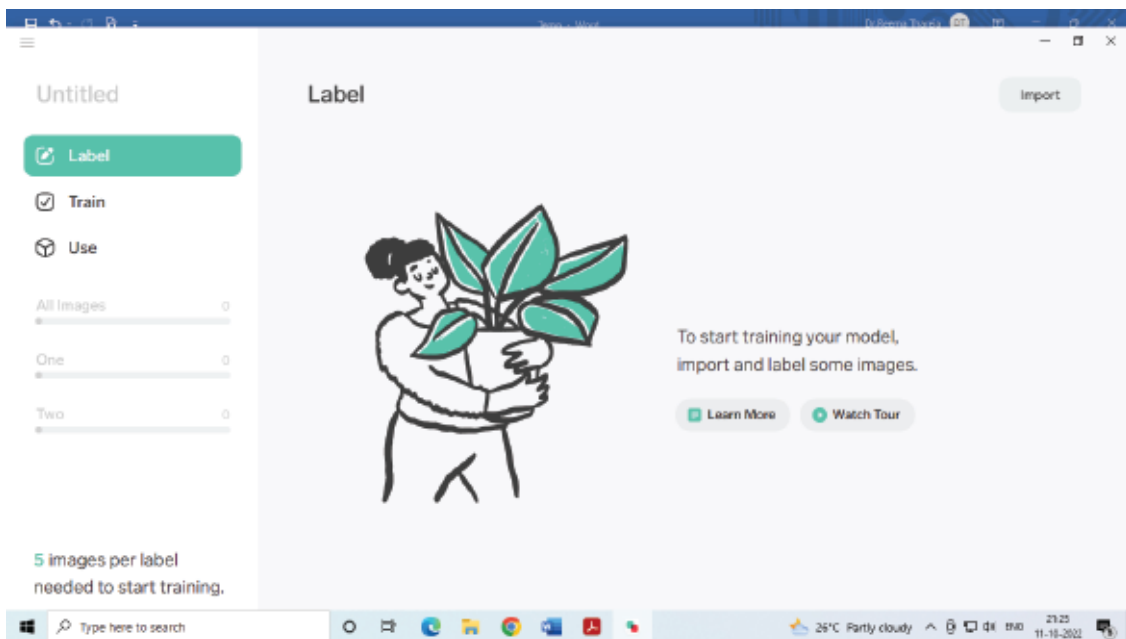
You may add more sets of images. However, to create a model, we need two different sets at least.

Step 3: Train Your machine learning model. You can test if the model works accurately or not by importing another image. See if Lobe predicts the correct label. To check the prediction, click on Play button and then either drag and drop a picture into the app, or click Import. If Lobe correctly predicts the label for the imported image, click the green tick button; if not, click the red button.

You may add any number of images to train the AI model to recognize different variations of the subject. Microsoft suggests that you should add between 100 and 1,000 images per label, depending on the complexity of the task.

Step 4: Optimize the machine learning model. If Lobe makes incorrect predictions, you will have to make it more reliable. You may add more variations of mislabelled images to make future predictions more accurate.

Step 5: Export the Lobe Model to be used in an App.



Credit: Screenshot of Lobe, used with permission from Microsoft



Black box AI: The AI program that is not able to explain decisions.

Responsible AI: An AI system that is explainable, monitorable, reproducible, secure, human-cantered, unbiased and justifiable.

Faceswap video blackmailing: Blackmailing people with morphed images and/or videos.

Ethical AI system: An AI system that solves a particular problem, uses unbiased data and is trained using the right learning model for the problem and is monitored to evaluate if the results produced by it are right and fair.

Robotics: The study focused on designing, developing and programming physical robots which are able to interact with the physical world and automate one or more tasks.

Humanoid Robots: Robots that look like or mimic human behaviour and usually perform human-like activities (like walking, carrying objects). Example, Hanson Robotics' Sophia and Boston Dynamics' Atlas.

Autonomous Robots: Robots that operate independently of human operators and are usually designed to perform tasks in open environments without any human supervision.

Teleoperated Robots: Semi-autonomous bots that use a wireless network to enable human control from a safe distance. They are usually deployed in extreme geographical conditions, weather and circumstances.

Augmenting Robots: Also known as VR robots, they are used to enhance or replace current human capabilities.

Independent robots: Robots that are deployed to replace humans for performing dangerous, mundane or otherwise impossible tasks.

Dependent Robots: Robots that are non-autonomous as they interact with humans to perform their actions.

Bots: Also known as software robotics, they are computer programs that perform tasks autonomously.

Chatbot: A computer program that simulates conversation both online and over the phone and is often used in customer service scenarios.

Sensors: Devices that detect the events or changes in the environment and send data to the computer processor.

Actuators: The motor parts that facilitates a robot's movement.

End Effectors: The physical, typically external components of a robot that facilitates it to complete a task. For example, in factories, robots have interchangeable tools like paint sprayers, gripping claws or even hands for undertaking tasks.

Cognition: The capability of a system to make purposeful decision and execute them to achieve its highest-order goals.

Trajectory planning: Moving from point A to point B while avoiding collisions over time in a 2D or 3D space.

Generative adversarial network (GAN): A machine learning (ML) model in which two neural networks compete with each other to become more accurate in their predictions.

Neuromorphic computing: A technique in which a computer (both hardware and software) is modelled based on the architecture of the human brain and its nervous system.



RECAPITULATION POINTS

- An ideal AI system must not be biased.

- As of now, there is no liability for actions on machines. There is no clarity on what legal aspects bind machines when they become increasingly smart.
- In artificial intelligence as a service, companies use off-the-shelf AI tools to implement and scale AI techniques at a very low price as compared to building a complete in-house AI system.
- Stationary robots used in a factory work on AC power while others operate via an internal battery.
- Reinforcement learning focuses on learning based on feedback.
- Affective computing focuses on developing systems that can identify, interpret, process and simulate human emotions.
- Programming by Demonstration (PbD) creates a prototype using a combination of physical and virtual objects.
- Robot navigation aims to make the robot reach its goal position efficiently and reliably based on its knowledge about its environment, the goal position and values of the sensor.
- Path planning is a strategic problem-solving competency (capability), as the robot must make vital decisions to achieve its goals. The second competency is tactical extreme to avoid all obstacles and collisions by modulating the trajectory of the robot.
- Trajectory planning is different from path planning as it is parametrized by time, velocity and kinematics.
- *Degrees of freedom* specify modes in which a mechanical device or system can move.
- A robot is said to be holonomic If the number of controllable degrees of freedom are greater than or equal to the total degrees of freedom.
- Artificial potential field algorithm places values over the map with the goal having the lowest or highest value that either increases or decreases, depending on the distance from the goal.
- Visibility graph finds paths that have minimum lengths by bringing the roads closer to the obstacles; Voronoi diagram, on the other hand, keeps roads as far as possible from the obstacles.
- In collaborative systems, humans and machines (with AI) become effective partners by overcoming each other's limitations and augmenting human abilities.
- AI systems boost human's creativity, analytic and decision-making abilities by providing the right information at the right time.

- The Nash equilibrium is a condition in which all the players playing the game accept that there is no better solution to the game than the current situation. No player would gain anything after changing their current strategy.
- Game theory is used to understand the dynamics of a game to optimize the possible outcome of its players. In contrast to this, inverse game theory designs a game based on the players' strategies and aims.
- In a GAN, the generative model uses input features to create new samples that resemble the main features of the original samples. The original samples and the newly generated samples are then passed to the discriminative model to identify which samples are genuine and which are fake.
- Neuromorphic systems deviate from the Von Neumann architecture and introduce a new chip architecture that collocates memory and processing together on each individual neuron.



REVIEW QUESTIONS

1. List some concerns in using an AI system.
2. How can AI become a risk?
3. List the guidelines for making an ethical AI system.
4. Why is AIaaS gaining importance these days?
5. Explain the different types of robots.
6. Explain the different types of bots.
7. Discuss the planning algorithms used by a robot.
8. What do you understand by the term 'degree of freedom'? When is a robot said to be holonomic?
9. Differentiate between two commonly used techniques in road-planning algorithms.
10. Compare the exact cell decomposition with approximate cell decomposition.
11. What are the two types of generative adversarial network?
12. When is a machine said to achieve AGI?
13. Compare the human brain with neuromorphic systems and traditional computers.

14. Give the applications of edge computing.
15. Write a short note on the following:
 1. Collaborative systems.
 2. Algorithmic game theory
 3. Nash equilibrium
 4. Neuromorphic systems
16. Differentiate between:
 1. Human-controlled and fully-autonomous bots
 2. AI programs and robots
 3. Dependent and independent robots
 4. Game theory and inverse game theory
 5. Human brain and neuromorphic systems
 6. Domain adaptation and domain confusion



1. AI program that is not able to explain decisions is known as _____.
2. _____ and _____ are examples of voice assistants.
3. Some major vendors of AIaaS are _____, _____ and _____.
4. _____ is Google's AI platform used to create and deploy machine learning projects.
5. Software robotics is also called _____.
6. _____ is a computer program that simulates conversation both on-line and over the phone and is often used in customer service scenarios.
7. _____ are the devices that detect the events or changes in the environment and send data to the computer processor.
8. Robots get power supply from a _____.
9. _____ is used to give voice commands to AI robots.
10. _____ planning refers to moving from point A to point B while avoiding collisions over time in a 2D or 3D space.
11. _____ is the pose of a robot describing its position.
12. _____ cell decomposition technique is less frequently used in mobile robot applications.

13. In _____ game of chess, all players can see what moves the other player(s) are making.
14. The _____ is a condition in which all players playing the game accept that there is no better solution to the game than the current situation.
15. _____ is a machine learning model in which two neural networks compete with each other to become more accurate in their predictions.
16. The huge availability of address space in _____ has led to the development of the Internet of Things.
17. _____ learning systems comprise multiple layers and each layer learns different features.
18. In an object detection task, initial lower layers of the network learn _____ features and the higher layers learn _____ features.



STATE TRUE OR FALSE

1. A black-box AI system is explainable, monitorable, reproducible, secure, human-centred, unbiased and justifiable.
2. Super-intelligent AI is unlikely to exhibit human emotions.
3. Cobot is an intelligent robot.
4. Monitoring Bots collect statistics to report speed and status of the website.
5. Actuators can create linear as well as rotary motion.
6. Feedback learning focuses on learning based on feedbacks.
7. Cognition refers to the capability of a system to make purposeful decisions and execute them to achieve its highest-order goals.
8. Holonomic robots are easy to work with.
9. The challenge of road path planning algorithm is to maximize the number of total roads.
10. Path in the Voronoi diagram may not be optimal in terms of total path length.
11. Potential field path planning algorithm creates a field across the robot's map that directs the robot to the goal position.
12. In Non-zero sum games, a player gains something that causes a loss to the other players.

13. Nash equilibrium can be attained easily in asymmetric than symmetric games.
14. Game theory is used to understand the dynamics of a game to optimize the possible outcome of its players.
15. Neuromorphic systems deviate collocates memory and processing together on each individual neuron.
16. The more you want to inherit features from a pre-trained model, the more you have to freeze layers.
17. Transductive transfer learning is used when the domains of the source and target tasks are not exactly the same but are interrelated.
18. *In symmetric feature-representation transfer approaches*, the source features are transformed to match the target ones.
19. Homogeneous transfer learning methods are preferred over heterogeneous ones.
20. In deep neural networks, initial layers freeze and higher layers are fine-tuned.
21. Domain adaptation is done by matching the distributions of the target and source domains to ensure that samples are indistinguishable to the classifier.



MULTIPLE CHOICE QUESTIONS

1. Which type of robot is known as a cobot?
 1. Humanoid
 2. Autonomous
 3. Pre-programmed
 4. Tele-operated
2. _____ are the motor parts that facilitate a robot's movement.
 1. Sensors
 2. Actuators
 3. End effectors
 4. Power supply
3. Actuators are operated using _____.
 1. electricity
 2. compressed air

3. oil
4. All of these
4. Identify the incorrect statement.
 1. An ideal AI system must not be biased.
 2. Teleoperated robots are semi-autonomous bots that use a wireless network to enable human control from a safe distance.
 3. VR robots are used to enhance or replace current human capabilities.
 4. None of these
5. Learning solves a problem with the help of an already solved problem.
 1. Machine
 2. Deep
 3. Transfer
 4. Reinforcement
6. In _____ learning, for every correct action, the robot gets a reward and a punishment for an incorrect action.
 1. Machine
 2. Deep
 3. Transfer
 4. Reinforcement
7. _____ computing imbibes emotional intelligence in robots.
 1. Cognitive
 2. Affective
 3. Edge
 4. Machine
8. Markov decision processes is based on _____ planning algorithm.
 1. Sampling-based
 2. Grid-based
 3. Reward-based
 4. Road-map path
9. In a _____ game, every participant has the same goal(s).
 1. Symmetric
 2. Simultaneous
 3. Zero sum
 4. Cooperative
10. _____ designs a game based on the players' strategies and aims.
 1. Game theory

2. Inverse game theory
 3. Nash equilibrium
 4. Collaborative systems
11. _____ computing is a technique in which a computer is modelled based on the architecture of the human brain and its nervous system.
1. Neuromorphic
 2. Cognitive
 3. Affective
 4. Edge
12. Identify the correct statement.
1. Neuromorphic systems can be either digital or analogue.
 2. Memristors can store a range of values.
 3. Neuromorphic systems also perform well with noise.
 4. Neuromorphic systems work with fixed data and are rule-driven.
13. _____ sensors detect noise and measure distance between two objects.
1. Speed
 2. Ultrasonic
 3. Radiation
 4. Photo-electric



Fill Up

1. black box AI.
2. Amazon's Alexa and Apple's Siri
3. Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platform (GCP)
4. Google Cloud ML
5. Bots
6. Chatbot
7. Sensors
8. Battery
9. NLP

10. Trajectory
11. Configuration
12. exact
13. Perfect information
14. Nash equilibrium
15. Generative adversarial network (GAN)
16. IPv6
17. Deep
18. Generic, task-specific

State True or False

1. False
2. True
3. False
4. True
5. True
6. False
7. True
8. True
9. False
10. True
11. True
12. False
13. False
14. True
15. True
16. True
17. True
18. False
19. False
20. True
21. False

Multiple Choice Questions

1. b
2. b

- 3. d
- 4. d
- 5. c
- 6. a
- 7. b
- 8. c
- 9. a
- 10. b
- 11. a
- 12. d
- 13. b

Previous chapter

< [7. Natural Language Processing](#)

Next chapter

[9. Where AI Is Heading Today?](#) >

Table of contents expanded

9

Where AI Is Heading Today?

9.1 Expert System

An expert system is a computer program that solves complex problems in a particular domain, at the level of extra-ordinary human intelligence and expertise. That is, it is designed to provide decisions just like a human expert. Expert systems are the computer applications developed to solve complex problems.

Expert systems are designed for a specific domain, such as **medicine, science**, etc.

To perform its task, the expert system analyses user queries to extract knowledge from its knowledge base using the reasoning and inference rules.

The first expert system was developed in 1970. This system makes decisions for complex problems using **both facts and heuristics like a human expert**. The performance of an expert system depends on the knowledge stored in its knowledge base. More the amount of knowledge stored in the KB, better is the performance of the expert system. ***The suggestion of spelling errors while typing in the Google search box is a perfect example of expert systems used these days by all users.***

An expert system is not used to replace the human experts; instead, it is used to assist the human in making a complex decision.

The working of an expert system can be understood through the block diagram given in **Fig. 9.1**. The main components of an expert system include user interface, knowledge base and inference engine.

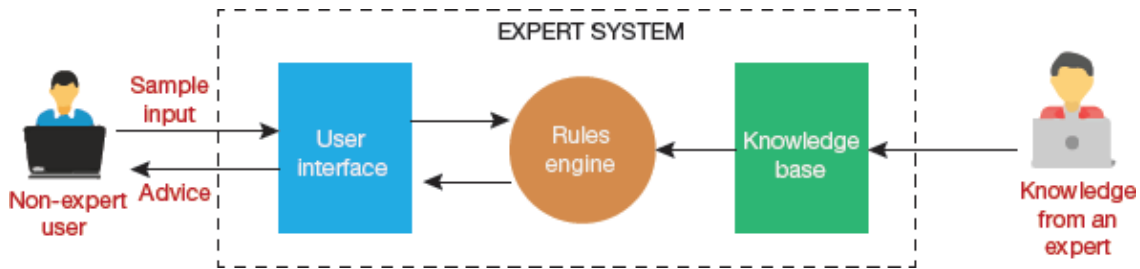


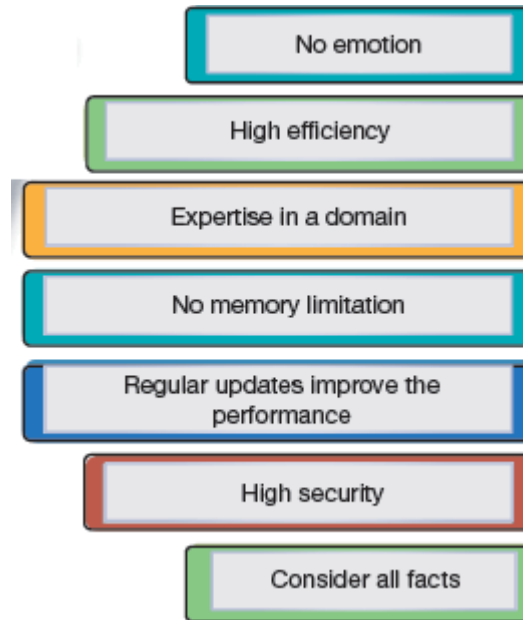
FIGURE 9.1 Working of an expert system

9.1.1 Popular Examples of the Expert System

1. **DENDRAL**: It is a chemical analysis expert system. DENDRAL was used in organic chemistry to detect unknown organic molecules with the help of their mass spectra and knowledge base of chemistry. The expert system studied a substance's spectrographic data to predict its molecular structure.
2. **MYCIN**: It is one of the earliest backward chaining expert systems that was designed to diagnose blood clotting diseases and find the bacteria causing infections like bacteraemia and meningitis. MYCIN could also recommend antibiotics and drugs based on the person's physical conditions.
3. **PXDES**: It is an expert system that determines the type and level of lung cancer by analysing picture(s) from the upper body.
4. **CaDeT**: CaDeT is an expert system that works like a diagnostic support system to detect cancer at early stages.
5. **R1/XCON**: The expert system could select specific software to generate a customized computer system based on user's requirements.
6. **DXplain**: It is a clinical support system that could suggest the presence of one or more diseases based on the findings of the doctor.

9.1.2 Characteristics of an Expert System

Expert systems are among the first truly successful forms of artificial intelligence (AI) software. Let us read about their distinguishing characteristics (refer **Fig. 9.2**).

**FIGURE 9.2**

1. **High performance:** An expert system provides high performance for solving any type of complex problem of a specific domain with high efficiency and accuracy. The success of an expert system depends on its knowledge base. So, to get accurate knowledge continuously, the knowledge base is updated regularly.
2. **Understandable:** The results of an expert system are easily understood by users. The system takes input in human language and provides the output in the same language.
3. **Reliable:** An expert system generates efficient and accurate output. This makes it a reliable system.
4. **Highly responsive:** An expert system generates results for any complex query within a very short period of time, making it highly responsive. The total time by an expert system should be less than that taken by a human expert.
5. **No memory limitations:** An expert system can store and exploit huge amounts of data.
6. **Expertise in a domain:** There are many human experts in a particular domain and each of them specialize in a different skill and has different experiences. When we put the knowledge gained from these human experts into the expert system, the results are provided by utilizing all the facts and knowledge

7. **Not affected by emotions:** An expert system is not affected by human emotions such as fatigue, anger, depression, anxiety, etc. This helps them to give a consistent performance every time they are used.
8. **Not biased:** To respond to any query, the expert systems check and consider all the available facts.
9. **Reduce cost:** Human experts charge a heavy amount for examining, diagnosing and treating a patient. But with an expert system, multiple experts can be consulted at much cheaper rates.
10. **Non-perishable:** Human experts are perishable as they have a limited life span but an expert system is permanent. Once developed, it can be used without any limitations of expiry.
11. **Intelligent:** These systems use knowledge base and inference engine to solve complex problems. This is done by deducing new facts from existing facts that are represented as if-then rules.

9.1.3 Components of an Expert System

An expert system mainly consists of three components—**user interface**, **inference engine** and **knowledge base** (as shown in **Fig. 9.3**).

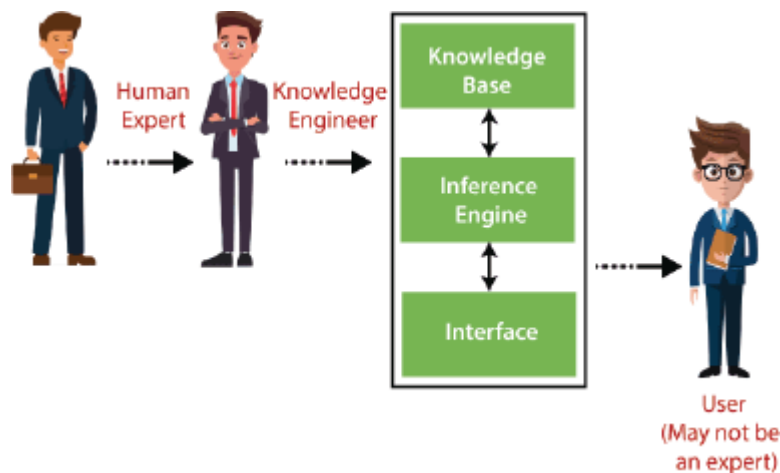


FIGURE 9.3 Components of an Expert System

User Interface

The user interacts with expert system through the user interface. It is through this interface that the user enters queries in a readable format. The interface then passes it to the inference engine. After getting response from the inference engine, it displays the output to the user. The

user interface is designed to be very simple and intuitive so that even a **non-expert user can communicate with the expert system to find solution to a complex problem.**

The user interface also explains how the expert system derived a particular conclusion. This explanation may be text displayed on screen, verbal narrations in natural language or a listing of rules on the screen. Through this, the user interface makes it easy to trace the credibility of the conclusions. An effective user interface must have the following features:

1. Help users to accomplish their goals in shortest possible way.
2. Be easy to customize as per user's requirements.
3. Make efficient use of user input.

Inference Engine (Rules of Engine)

The inference engine is the main processing unit and hence known as brain of the expert system. The engine derives conclusion or deduces new information by applying inference rules to the knowledge base. There are two types of inference engine:

Deterministic Inference Engine

This engine uses facts and rules to draw conclusions. These conclusions are thus assumed to be true.

Probabilistic Inference Engine

This inference engine contains uncertainty in conclusions as it is based on the probability.

Efficient procedures and rules used by the inference engine helps to formulate correct, flawless solutions. In case there is a conflict when multiple rules are applicable to a particular case, the inference engine resolves the problem by selecting the best rule that is applicable in the current context. Inference engines can also include explanation and debugging abilities. The explanation module, if incorporated, can be used by the expert system to give the user an explanation about how it reached a particular conclusion.

The inference engine generally uses two strategies for acquiring knowledge from the knowledge base and deriving solutions. These are as follows:

1. **Forward Chaining:** As discussed earlier, forward chaining starts from the known facts and rules (refer **Fig. 9.4**). After applying the inference rules, conclusions are drawn and then added to the known facts. Forward chaining strategy facilitates an expert system to answer the question, **‘What can happen next?’** For example, forward chaining can be used to predict stock prices after changes in interest rates.

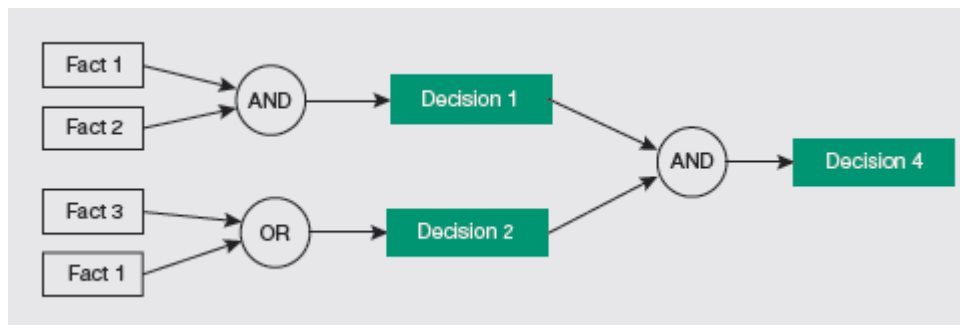


FIGURE 9.4 Forward Chaining

2. **Backward Chaining:** Backward reasoning, on the other hand, starts from the goal and works backward to prove the known facts (as shown in **Fig. 9.5**). This strategy enables an expert system to answer the question, **‘Why this happened?’** If an event has already taken place, then the inference engine can be used to find out which conditions occurred in the past for this result. Thus, backward chaining strategy is used for discovering cause or reason. For example, diagnosis of blood cancer in humans.

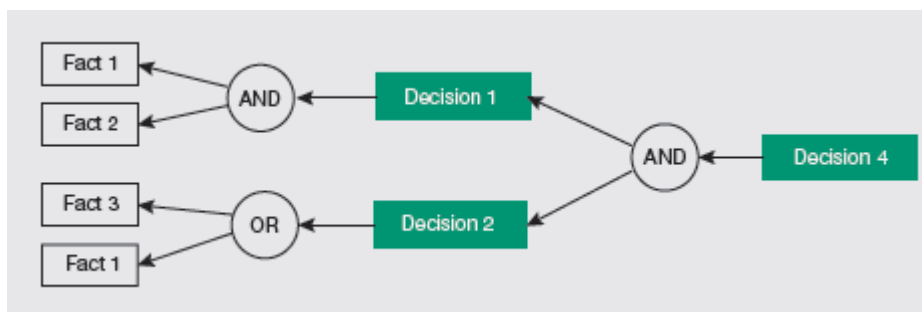


FIGURE 9.5 Backward Chaining

Knowledge Base

Before going to knowledge base, let us first understand what knowledge is. From junior classes, we are taught that data is a collection of facts. Data, when processed, gives us information. Data, information and past experience combined together is known as knowledge.

We have read earlier that the knowledge base stores knowledge acquired from different experts of a particular domain. More the knowledge in the knowledge base, more precise is the result from the expert system.

The knowledge base contains information and rules of a particular domain or subject. The two main components of a knowledge base are factual knowledge and heuristic knowledge.

Factual knowledge is based on facts and accepted by knowledge engineers in that domain.

Heuristic knowledge is based on practice, probability, evaluation and experiences.

Knowledge in a knowledge base is usually represented using the If-else rules and is acquired by extracting, organizing and structuring the domain knowledge under the guidance of human experts. This knowledge is domain-specific and of high-quality.

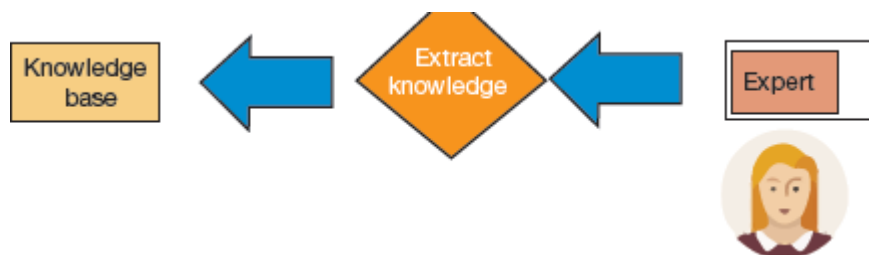


FIGURE 9.6 Creation of Knowledge Base

CASE STUDY: Working of MYCIN as an expert system.

Step 1: Knowledge base is created with expert knowledge about bacterial infection—their causes, symptoms, treatment, etc. (refer **Fig. 9.6**).

Step 2: The knowledge base is regularly updated. In order to test it, the doctor gives the expert system a new problem of identifying the presence of the bacteria by inputting the details of a patient, including the symptoms, current condition, and medical history.

Step 3: The system prompts the patient to provide general information like gender, age, etc.

Step 4: The system applies if-then rules using the inference engine and the facts stored within the knowledge base.

Step 5: The output is displayed to the patient through the user interface.

9.1.4 Participants in the Development of Expert System

In **Fig. 9,7**, we see that there are three primary participants in an expert system:

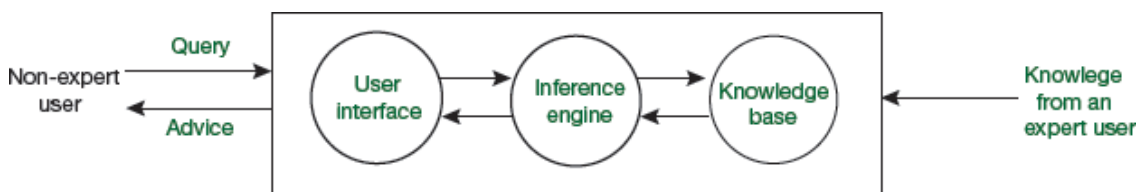


FIGURE 9.7 Participants in an expert system

Expert

The performance of any expert system depends on the knowledge fed in its knowledge base. This knowledge is provided by human experts who are specialized in that specific domain.

Knowledge Engineer

The term 'knowledge engineering' refers to the process of building an expert system and its practitioners are called knowledge engineers. The main task of a knowledge engineer is to ensure that the computer possesses all the knowledge required to solve a problem. For this, he/she gathers the knowledge from the domain experts and then represent that knowledge in the format accepted by the knowledge base.

End User

The end user who uses the expert system may or may not be well versed in that domain.

He/she just uses the expert system to find solution or advice for his queries, which are usually complex.

9.1.5 Capabilities of the Expert System

An expert system can do the following things:

1. **Advise:** The system can give its advice to users for any query—simple or complex— that fits its domain.
2. **Provide decision-making capabilities:** An expert system can be designed to make complex decisions in any field, be it finance, medical science or any other.
3. **Demonstrate a device:** It is can be used to demonstrate features, specifications, usage of any new product.
4. **Problem-solving:** It can be used to solve complex problems.
5. **Explaining a problem:** An expert system can be used to provide a detailed description of an input problem.
6. **Interpreting the input:** It can interpret the input given by the user.
7. **Predicting results:** An expert system can predict results using past data.
8. **Instructing and assisting human in decision-making**
9. **Diagnosing:** Expert systems, especially in the field of medical science, are used to diagnose any disease.
10. **Justifying the conclusion:** Expert systems not only find solutions to complex real-world problems by drawing conclusions from known facts but also justify those conclusions.
11. **Suggesting:** The best part is that an expert system also suggests alternative options to a problem.

However, expert systems cannot do the following:

1. Substitute human decision-makers
2. Possess human capabilities

3. Produce accurate output for inadequate knowledge base
4. Refining knowledge base
5. Use emotions to make decisions

9.1.6 Advantages of Expert Systems

1. Results obtained by an expert system are highly reproducible.
2. They can be used in risky places where it is not safe for humans to work.
3. They give accurate results, especially when the knowledge base is updated regularly.
4. They give consistent performance at a fast speed as these systems are not affected by emotions, tension or fatigue.
5. They are easily available due to mass production of software.
6. They are affordable as the production cost is reasonable.
7. They work at great speed and reduce human effort.
8. They are capable of explaining how the solution was obtained.
9. They improve the quality of decisions.
10. They gather scarce expertise knowledge in good amount and use it efficiently.

9.1.7 Limitations of Expert Systems

1. Results obtained from an expert system may get wrong if the knowledge base contains wrong/obsolete information.
2. Each problem is different; therefore, the solution from a human expert may be strikingly different from that of an expert system.
3. They lack creativity for different scenarios, especially in extraordinary situations.
4. They have high development and maintenance costs.
5. Knowledge acquisition is a challenging task.
6. Success depends on knowledge acquired from human experts.
7. These systems cannot learn from themselves and hence require manual updates.
8. They cannot produce correct results from less amount of knowledge.
9. They need excessive training.
10. They do not have any emotions.
11. They do not possess any common sense.

12. They are developed only for a particular domain.
13. They are not capable of making decisions in extraordinary situations.
14. They work on Garbage-in Garbage-out (GIGO) principle. This means that if there is an error in the knowledge base, the expert system will give incorrect results.
15. They have high maintenance cost.

9.1.8 Applications of Expert Systems

Expert systems are one of the prominent research domains of AI. Some common applications of expert system are as follows:

1. **Designing and manufacturing** devices such as VLSI systems, camera lenses and automobiles.
2. **Publishing** relevant knowledge content to users. An advisor and a tax advisor are two popularly used expert systems in this domain.
3. Finance industries use expert systems to detect any type of possible fraud, suspicious activity, stock market trading and advise bankers on whether they should approve the loans or not.
4. **Expert systems are extensively used for medical diagnosis and treatment. In fact, it was the first area where these systems were used.**
5. **Expert systems are used for planning and scheduling** tasks for achieving the project's goal. Companies also use expert systems for scheduling airlines, cargo customer order, computer resources and various manufacturing task.
6. Tracking the progress of a software development project.
7. **Monitoring applications** compare data continuously with observed system or with prescribed behaviour to detect leakage in long petroleum pipeline using expert systems.
8. Process control systems use expert systems to control a physical process based by continuously monitoring it.
9. The automobile and electronics industries use expert systems to identify faults in vehicles and computers manufactured.
10. Diagnosing complex electronic, electromechanical and diesel-electric locomotive system.
11. Forecasting crop damage or in general predicting results of any experiment/activity.

12. Identifying the structure of a chemical compound.
13. Assessing geologic structure from dip meter logs and space structure through satellite and robot.
14. Evaluating civil cases, product liability, employee performance, etc. using logs.
15. Detecting virus attack in a computer system.
16. Repair and maintenance projects.
17. Optimizing warehouses.
18. Configuring manufactured objects.

9.1.9 Expert System Technology

Expert systems technologies include the following.

1. **Expert System Development Environment:** The development environment of an expert system includes hardware and tools like
 1. Workstations, minicomputers, mainframes.
 2. High level symbolic programming languages (like **LIS**t Programming (LISP) and **PRO**grammation en **LOG**ique (PROLOG)).
 3. Large databases.
2. **Tools:** The cost of developing an expert system gets reduced to a large extent with the help of tools that help developers by providing,
 1. Powerful editors
 2. Debugging capabilities with multi-windows
 3. Rapid prototyping
 4. Having in-built definitions of model, knowledge representation and inference design.
3. **Shells:** A shell is an expert system without knowledge base. It facilitates developers by providing modules for knowledge acquisition, inference engine, user interface and explanation facility. Two popularly used shells are JESS and Vidwan. While Java Expert System Shell (JESS) provides Java API for creating an expert system, *Vidwan*, enables knowledge encoding in the form of IF-THEN rules.

9.1.10 Development of Expert Systems

The process of developing an expert system is iterative. The steps to develop such a system are described below.

Identify Problem Domain

The first step to develop any system is to identify the problem that it intends to solve. Ensure that this problem is solvable by an expert system. If the problem can be solved, identify the human experts in the task domain and analyse the cost-effectiveness of the system. Knowledge engineer(s) and domain experts work in coherence to define the problem and list the characteristics of the system.

Design the System

To design the expert system, first identify the technology (hardware, tools and shells) that will be used. Understand how well the system will be integrated with the other systems and databases. Prepare a framework for representing concepts in domain knowledge.

Develop the Prototype

In this step, the knowledge engineer acquires domain knowledge from the human experts and represent it in the form of If-THEN-ELSE rules. The knowledge expert also determines how heuristic knowledge could be integrated in the reasoning process and what type of explanation would be useful.

Test and Refine the Prototype

The knowledge engineer tests the prototype to identify any deficiencies in performance or results by using sample cases. Testing is also done along with end-users to unleash additional errors or discrepancies. The results of testing are then used to refine the system.

Develop and Complete the System

Once the expert system is developed and tested, it is then tested along with other elements of its environment, including end users, databases and other information systems to ensure that it interacts in the pre-scribed manner and smoothly integrates with them to render the desired functionality. At this step, the expert system is clearly documented and users are trained for its usage.

An expert system is also used for information management and help desks management.

Maintain the System

Maintenance is an ongoing process. In this phase, the knowledge base is regularly reviewed and updated. New requirements from the users or blending with new interfaces of other information systems as those systems evolve are steps that need to be done time to time.

TABLE 9.1 Traditional System vs Expert System

Conventional System	Expert System
Knowledge and processing are combined in one unit.	Knowledge database and the processing mechanism are two separate components.
The program does not make errors (unless there is an error in input data or programming logic).	The expert system may give less accurate results.
The system is operational only when it is fully developed.	The expert system is optimized on an ongoing basis and can be used even with a small number of rules.
Program is executed step-by-step according to a fixed algorithm.	Execution is done logically and heuristically.
The problem expertise is encoded in both program as well as data structures	Problem-related expertise is encoded only in data structures.
It needs full information.	It can be functional even with insufficient information.
It uses data more efficiently.	It utilizes knowledge efficiently.
It gives solution without explanation.	It gives solution and explains how it was obtained.
It does not represent knowledge symbolically for computations.	It symbolically represents knowledge for computations.

TABLE 9.2 Human expert vs Expert System

Human Expert	Artificial Expertise
Perishable	Permanent
Difficult to transfer	Transferable
Difficult to document	Easy to document
Unpredictable	Consistent
Expensive	Cost-effective system

9.2 Internet of Things

Although the concept was not named until 1999, the Internet of Things (IoT) has been in development for decades. IoT means a system of interrelated computing devices, machines, objects, animals or people that have unique identifiers for identification and also the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction (as shown in **Fig. 9.8**).



Credit: 363868 / Shutterstock

FIGURE 9.8 Interconnected Things

If the thing, in the IoT, is a person, then he can have heart monitor implant; if it is an animal, it can have a biochip transponder, if it is an automobile, it can have sensors to generate alert alarms (e.g., when the pressure in a tyre is too low) and if it is any other natural or man-made object, then it is assigned an IP address and provided with the ability to transfer data over a network.

The huge availability of address space in IPv6 has led to the development of the IoT. With more IP addresses available, more devices can be connected to exchange information with each other.

Today, IoT is being widely used for precision agriculture, building management, healthcare, energy and transportation. The first Internet appliance was a Coke machine at Carnegie Melon University in the early 1980s when the programmers succeeded in connecting the machine over the Internet to check the status of the machine and determine the availability of a cold drink. This check helped them to save their time as they did not have to go to the machine to buy Coke when it was not available in the machine.

Later, other devices like cell phones, coffee makers, washing machines, headphones, lamps, wearable devices and almost anything you can think of could be connected. It is interesting to know that in 2022, the market for the IoT is expected to grow 18% to 14.4 billion active connections and by 2025, there will be approximately 27 billion connected IoT devices. So, we can conclude that, IoT is a giant network of connected ‘things’ (including people) which will have relationships between people-people, people-things and things-things.

In simple terminology, IoT is the concept of connecting any devices over the Internet.

9.2.1 Examples of Applications of IoT

Whenever we read about a new technology, the first question that comes to our mind is how is this technology going to help us? While with IoT, anything that can be connected, will be connected, but the question is why should we want these devices to be connected? There are many examples to justify the answers to these questions.

Case 1: Imagine you are going for meeting and your car tells you the best route to take. If there is a traffic jam, then your car sends a text message to the other party, notifying them that you will be late.

Case 2: Imagine that the moment your alarm rings to wake you, a message is sent to the coffee machine to start brewing coffee for you.

Case 3: The smart watch that you wear in office tells you when and where you were most active and productive.

Case 4: You have a self-driving car with complex sensors to detect objects in their path.

In 2019, IoT devices generated an estimated 18.3 zettabytes of data, which is expected to grow to 73.1 ZB by 2025.

Case 5: A smart football that can track how far and fast it is thrown and record those statistics via an app for future training purposes.

Case 6: You have a smart refrigerator that can text message you that milk packets are over in the fridge and you need to buy them before you come home. Or, your fridge checking the expiry date of an ice-cream and notifying you that you should not eat it now.

Case 7: Imagine that you have an AC installed in your home which is connected with your smartphone. On a very hot day, you can instruct your AC to start and set the room temperature to 18 degrees before you finally reach the house.

In an article, Ashton wrote that if computers knew everything about things (using data gathered by them) without any help from us, then we would be able to track and count everything. It would also reduce waste, loss and cost. We would know exactly when things need to be replaced, repaired or recalled, and whether they were fresh or past their best.



Credit: monicaodo. Shutterstock

FIGURE 9.9 A Smart City

In this way, IoT can help organizations save a lot of money through improved process efficiency, asset utilization and enhanced productivity. With improved tracking of objects using sensors and connectivity, companies can better analyse them and make smart decisions. For example, if you own a car manufacturing company, then you can know which accessories are the most popular by using sensors to detect which areas in the showroom are the most popular, and where customers linger longest. You can even use the available sales data to identify which components are selling fastest and then automatically align sales data with supply, so that popular items never go out of stock.

The information collected by IoT devices can be used to detect patterns, make recommendations, and detect possible problems before they occur.

We have heard that the Indian government is working hard to develop smart cities. Do you know that smart cities use IoT for efficient utilization

of resources? **Figure 9.9** gives an overview of such a city. IoT has the potential to transform entire cities by solving real problems citizens which are faced by the people every day. With the proper connections and data, the IoT can solve traffic congestion issues and reduce noise, crime and pollution.

9.2.2 IoT Products

Big companies like Honeywell, Hitachi, GE, Cisco, AT&T, Apple, Google, IBM, Microsoft, Skyworks, Iridium Communications, Red Hat, Zebra Technologies, InterDigital are already playing in the market to realize the benefits of IoT. Given below are some examples.

Amazon Echo for Smart Home works through its voice assistant, Alexa. Users can talk to Alexa and give order to perform a variety of functions. For example, users can tell Alexa to play music, provide a weather report, get sports scores, order an Uber and do much more.

Fitbit One – Wearables tracks your steps, floors climbed, calories burned, and quality of sleep. The device wirelessly connects with computers and smartphones to transmit your fitness data in understandable charts to monitor your progress.

Barcelona - Smart Cities: The Barcelona city in Spain is one of the foremost smart cities in the world. It has implemented several IoT initiatives that have helped to enhance smart parking and the environment.

Amazon Web Services, Microsoft Azure, IBM's Watson, Cisco IoT Cloud Connect, Salesforce IoT Cloud, Oracle Integrated Cloud and GE Predix are popular IoT platforms.

AT&T - Connected Car: AT&T added 1.3 million cars to its network in the second quarter of 2016. With this, the count of total number of connected cars rose to 9.5 million.

Thus, we see that IoT allows for virtually endless opportunities and challenges thereby making it a hot topic for research. According to a report, nearly \$6 trillion will be spent on IoT solutions over the next five years.

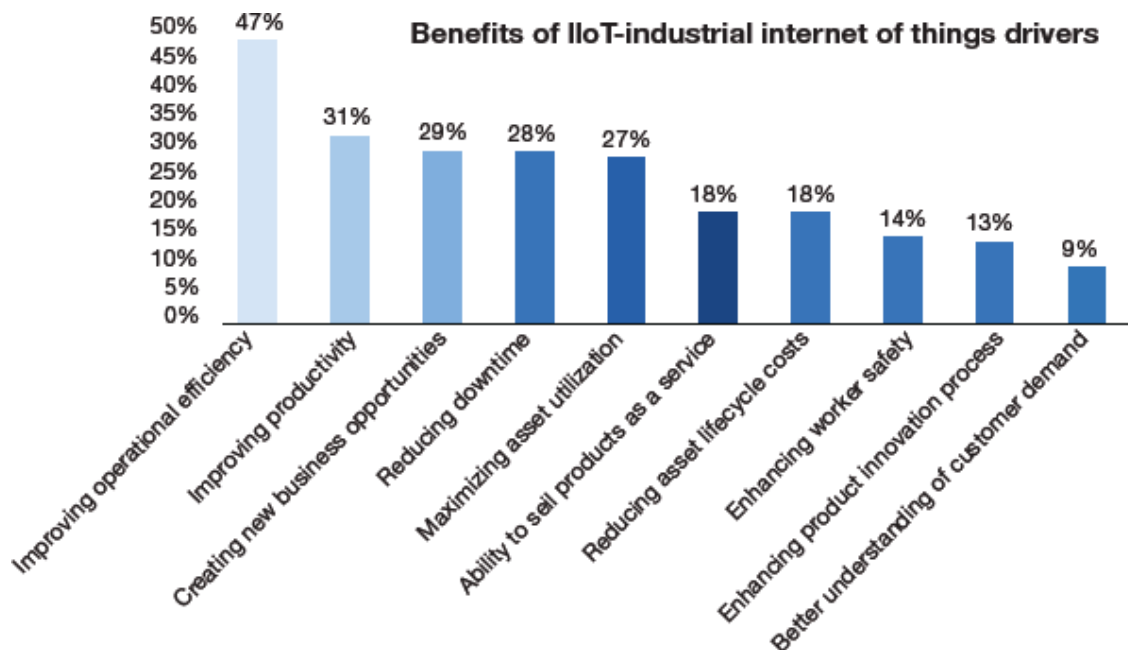


FIGURE 9.10 Benefits of IoT

9.2.3 Challenges

Security is the biggest issue that we often come across while talking about IoT. With billions of devices connected together, concerns of privacy and data sharing always haunt our minds.

In fact, the protection of sensitive data was ranked as the top concern among enterprises according to the 2016 Vormetric Data Threat Report. Hackers try to penetrate connected cars, critical infrastructure and even people's homes. Therefore, the main focus of companies is to ensure security of all the data generated by these devices. Most IoT devices do not encrypt communications if the data is transferred over a local Wi-Fi network. If the Wi-Fi network is unsecured, then it opens the gates for security threats wide open.

IoT devices used on patients as wearable devices, if left unattended (with technical errors), can be life-threatening for patient.

Another issue with IoT is that massive amounts of data is being generated by these devices. So, companies need to figure out how they would store, monitor, analyse and deduce results from this vast amount of data that is continuously being generated.

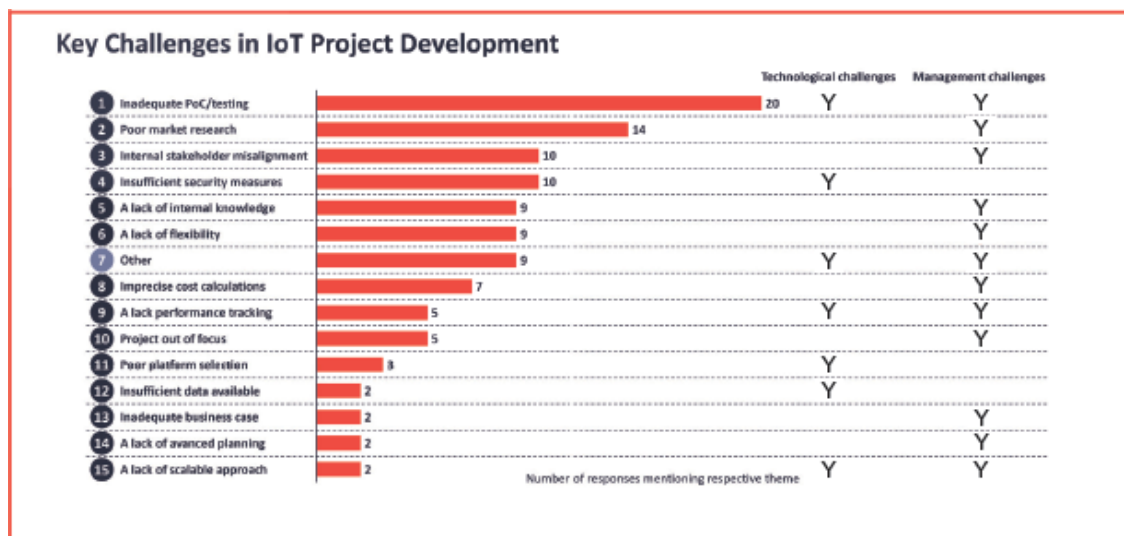


FIGURE 9.11 Key challenges in IOT project development

The graph in **Fig. 9.11** shows top challenges in the world of IoT. Besides security, other key concerns are as follows.

1. **Scalability:** When billions of Internet-enabled devices are connected, large volumes of data are generated that need to be processed. To process this huge data, devices need big data analytics and cloud storage for interpretation of useful data. Moreover, these systems need to be scalable to accommodate ever expanding data.
2. **Interoperability:** Technological standards in most areas are still fragmented and need to be converged to establish a common framework and standard for the IoT devices.
3. **Lack of standardization:** Since standardization process is still lacking, interoperability of IoT with legacy devices is critical which, in turn, prevent us to move towards the vision of truly connected interoperable smart objects.
4. **Lack of government support:** Government and regulatory bodies like the FDA should set up regulations by forming a standard committee for safety and security of devices and people.

5. **IoT Malware and ransomware:** Ransomware can lock out users from various devices and platforms and still use a user's valuable data and information. For example, a hacker can hijack a computer camera and take pictures. He may also lock a device and then demand ransom to unlock it and return the data.
6. **Connectivity:** Poor Internet connectivity is a challenge where IoT sensors are required to monitor process data and supply information.
7. **Lack of effective and informed government regulations:** IoT is a fast-developing area of technology and many legislators are either not aware or do not fully understand the workings of IoT, so they are reluctant to create or enforce regulations.
8. **No privacy policies:** There are no clear regulations for IoT devices on which information is private or public. For example, an IoT toy could collect information from a child without taking consent from the parents. The toy can then share that data with the manufacturer who could then share it with others.

IoT devices should be updated regularly. Updates should be small. Also, IoT devices should be backward compatible with older devices.

9. **Bandwidth strain:** As the number of IoT devices connected in a geographical area grows, more text/audio/video data will flow through those devices, thereby requiring higher bandwidth and placing strain on the server farms that process all that information.
10. **IoT professional skills gap:** IoT suffers from a professional skills gap. The main reason behind this is that IoT is growing rapidly and many people are yet to develop a skillset working with this technology. As of now, a few IoT experts are working in the industry, training new people even more difficult. In a survey, it was found that 38% of respondents reported that they lack in-house expertise in IoT network management, while 27% were facing a dearth of in-house experience to deploy IoT.

CASE STUDY EXAMPLE

Several high-profile IoT security breaches have been reported till date. One example is the 2015 Jeep Cherokee hack, when two hackers remotely took control of a car using vulnerabilities in the entertainment system to

access its dashboard functions. The hacker initiated a series of unexpected disturbances and finally disabled the brakes, causing the driver to swerve into a ditch.

Wireless IoT SIM cards should automatically update the system, removing the possibility of users not updating and leaving holes in system security.

CASE STUDY EXAMPLE

Inter-operability is a serious issue. Once it was seen that there was a problem when an insulin pump communicated only with a particular operating system. It makes the functionality unavailable for patients who may have a computer with a different operating system installed in it.

There are issues around the device's wireless standard. For example, if it uses Bluetooth, then the device could inadvertently connect to the wrong nearby device.

CASE STUDY EXAMPLE

A manufacturing plant where a large number of precise, high-resolution pictures are sent to 3-D printers could slow down the servers or even cause them to crash if the demands are too great.

Similarly, in a hospital, connected video cameras transfer of thousands of high-resolution radiology images and hundreds of telehealth video calls going on in a single day could significantly overload a server farm.

However, despite all odds, IoT is being used by several industries including manufacturing, defence, transportation, banks, retail, oil & gas mining, health care, connected home, banks to name a few.

9.2.4 Sensors

A sensor is a device that detects and responds to some type of input that it receives from the physical environment. This input could be heat, light, motion, pressure, moisture or any other environmental phenomena. As

the output, sensors usually generate a signal that is converted to human-readable form and then displayed at the sensor location or transmitted electronically over a network for reading or further processing.

For example, an oxygen sensor in a car's emission control system detects the gasoline/oxygen ratio. If the mixture is not optimal, the balance is readjusted.

Another example is motion sensors in home security lights, automatic doors and bathroom fixtures that send out microwaves, ultrasonic waves or light waves and detect when the flow of energy is interrupted by something entering its path.

Vision and imaging sensors detect the presence of objects or colours within their fields of view. They display a visual image of whatever it detects.

Temperature sensors detect thermal parameters and provide signals to the inputs of control and display devices. They are used to measure the thermal characteristics of gases, liquids and solids in many industrial processes.

Radiation sensors sense the presence of alpha, beta or gamma particles and provide signals to counters and display devices. They are usually used for surveys and sample counting.

Proximity sensors detect the presence of nearby objects through non-contacting means within a range of up to several millimetres. They are used in manufacturing operations to detect the presence of parts and machine components.

Pressure sensors detect forces per unit area in gases or liquids and provide signals to the inputs of control and display devices.

Position sensors sense the positions of valves, doors, throttles, etc. and supply signals to the inputs of control or display devices.

Photoelectric sensors sense objects passing within their field of detection. They can even detect colour, cleanliness and location, if required. Photo sensors are commonly used in manufacturing and material handling automation. For example, they are used for counting, robotic picking and automatic doors and gates.

Particle sensors sense dust and other airborne particulates and supply signals to the inputs of control or display devices. Particle sensors are common in bin and baghouse monitoring.

Motion sensors sense the movement or stoppage of parts, people, etc. and supply signals to the inputs of control or display devices. They are usually used for detecting the stalling of conveyors or the seizing of bearings.

Metal detectors sense the presence of metal in a variety of situations ranging from packages to people.

Level sensors are used to determine the height of gases, liquids or solids in tanks or bins. The information is then passed as signals to the inputs of control or display devices.

Leak sensors are used to identify or monitor the unwanted discharge of liquids or gases. Some leak detectors are used to measure the effectiveness of the seals in vacuum packages.

Humidity sensors measure the amount of water in the air and convert these measurements into signals that can be used as inputs to control or display devices.

Gas and chemical sensors sense the presence and properties of various gases or chemicals and relay signals to the inputs of controllers or visual displays.

Force sensors measure various parameters related to forces such as weight, torque, load, etc. and provide signals to the inputs of control or display devices.

Flow sensors sense the movement of gases, liquids or solids and provide signals to the inputs of control or display devices. They are used extensively in the processing industries.

Flaw sensors detect inconsistencies on surfaces or in underlying materials such as welds. They are used in a variety of manufacturing processes.

Flame detectors sense the presence and quality of fire and provide signals to the inputs of control devices. They are used in many combustion control applications like burners.

Electrical sensors sense current, voltage, etc. and provide signals to the inputs of control devices or visual displays.

Contact sensors detect physical touch or contact between the sensor and the object being observed or monitored. They are often used in alarm systems to monitor doors, windows and other access points. For example, when a door or window is opened/closed, a magnetic switch provides an indication to the alarm control unit so that the status of that entry point is known. They are also used as proximity sensors in robotics applications and automated machinery.

Non-contact sensors do not require a physical touch between the sensor and the object being monitored in order to function. Radar guns used by law enforcement to monitor the speed of vehicles is an example of a non-contact sensor.

Speed sensors detect the speed of an object or a vehicle.

Ultrasonic sensors detect noise and measure the distance between two objects. For this, high-frequency sound waves generated by active ultrasonic sensors are received back by the ultrasonic sensor for evaluating the echo. Time delay between transmitting and receiving the echo is used to calculate the distance to an object. Passive ultrasonic sensors are used for detecting ultrasonic noise present under specific conditions.

9.3 Artificial Intelligence of Things (AIoT)

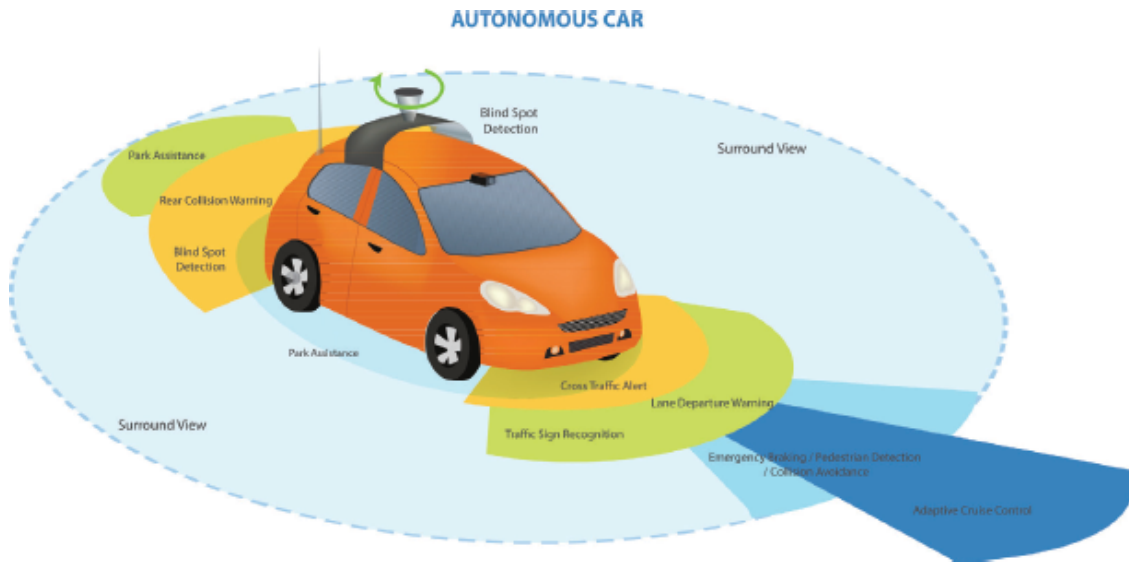
AIoT, or the combination of artificial intelligence (AI) technologies and the Internet of Things (IoT) infrastructure aims to make IoT operations more efficient, improve interactions between humans and machines and facilitates better data management and analytics.

We now clearly know that while AI simulates human intelligence processes by machines, IoT, on the other hand, is a system of interrelated computing devices with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. A *thing* or a computing device in IoT can be a person's heart monitor implant, a self-driving car with built-in sensors, a door mat or any other object that can be assigned an Internet Protocol address and transfer data over a network.

AIoT enhances the capability of IoT by adding the power of machine learning algorithms to improve decision-making processes.

9.3.1 How Does AIoT Work?

In AIoT devices, AI is embedded into programs and infrastructure components, like chips, which are all connected using IoT networks. Efficient exchange of data amongst all hardware, software and platform components without human intervention is ensured using pre-written APIs.



Credit: monicaodo / Shutterstock

FIGURE 9.12 Autonomous Vehicle

IoT devices generate as well as collect data, that is then analysed using AI techniques to provide better insights into data. This improves efficiency and productivity. **Figure 9.12** shows an autonomous vehicle in which AI is embedded into chipsets, which are all connected using IoT networks.

AIoT data when processed using Edge AI (discussed later in the chapter) minimizes the bandwidth needed to move data while avoiding possible delays to data analysis. This further adds to efficiency of IoT.

9.3.2 Where Does AI Unlock IoT?

IoT is all about implanting sensors into machines. These sensors collect streams of data through Internet connectivity. All IoT-related services follow five basic steps to perform the assigned task—**Create, Communicate, Aggregate, Analyse and Act**. Since Act depends on Analyse, the efficiency of any IoT system depends on the Analysis step. AI technology when used as the analysis stage, improves both efficiency and productivity manifold.

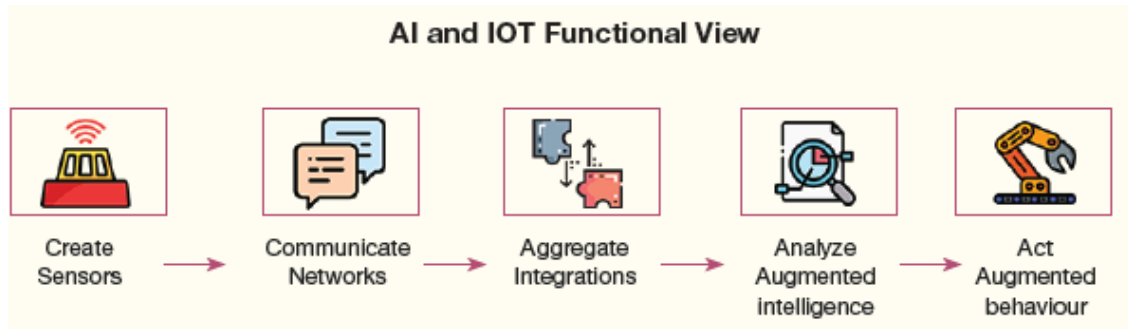


FIGURE 9.13 AI and IOT functional view

AIoT helps in achieving the following agile solutions:

1. Manage, analyse and obtain meaningful insights from data (refer **Fig. 9.13**)
2. Provide fast and accurate data analysis
3. Maintain a balance between requirements for localized and centralized intelligence
4. Protect data privacy and confidentiality
5. Ensure security against cyber attack

9.3.3 Applications and Examples of AIoT

AIoT is extensively used in the following domains.

1. **Smart cities:** Smart technology such as sensors, lights and meters collect data that is processed using AI to improve operational efficiency, drive economic growth and enhance quality of life for residents.
2. **Smart retail:** Smart cameras are now being used at retail outlets by managers to identify faces of customers who have scanned their items at the self-checkout counter before leaving the store. Moreover, cameras and sensors installed at multiple places to observe customers' movement and predict when they will reach the checkout line. This helps in maintaining dynamic staffing levels to enhance the productivity of the cashiers and reduce the customer's checkout time. Major retailers can use AIoT solutions to grow sales through customer insights. Data such as mobile-based user behaviour and proximity detection offer valuable insights to deliver personalized marketing campaigns to customers while they shop, increasing traffic in brick-and-mortar locations.

3. **Smart home:** In smart appliances installed at homes, data collected from human interaction with devices is analysed to understand user habits to provide customized support.
4. **Manufacturing:** These days, manufacturers use smart chips to detect when a particular equipment is not functioning properly, or when it is the right time to replace.
5. **Hiring:** AIoT technology when used to integrate data from social media and HR-related platforms can help to identify right talent without any bias.
6. **Autonomous vehicles:** These vehicles collect real-time data about nearby vehicles, pedestrians and other objects using multiple video cameras and sensors. Data is also analysed to monitor driving conditions and take appropriate decision instantly.
7. **Robots:** Autonomous robots have multiple sensors installed to gather data about the environment. AI is then used to make decisions that help robot to make smart and optimized moves.
8. **Healthcare:** Medical devices and wearables collect and monitor real-time health data, such as heart rate. Moreover, healthcare facilities produce high volumes of data including patient information, imaging and test results. This information is valuable and necessary to take good care of the patient care. So, this data needs to be accessed and processed quickly to suggest decision regarding patient's diagnostic and treatment.

IoT combined with AI not only improves diagnostic accuracy, enable telemedicine and remote patient care but also reduces the administrative burden of tracking patient health in the facility.
9. **Smart Thermostat Solution:** The smartphone integration with smart thermostat solution can be used by users to check and manage the temperature from anywhere. Users can set the temperature based on their work schedule and temperature preferences. For example, Nest's smart thermostat solution uses AI-powered IoT.
10. **Drone Traffic Monitoring:** AIoT can be used by drones to monitor real-time traffic and make adjustments to the traffic flow to reduce congestion. While drones can collect and transmit traffic data, AI can be used to analyse that data and make decisions about how traffic can be regulated- make diversions, adjustments to speed limits and timing of traffic lights without human intervention.

For example, the ET City Brain, a product of Alibaba Cloud, uses AIoT to optimize the use of urban resources. It can detect accidents, illegal parking and can alter traffic lights to help ambulances reach patients/hospitals faster.

11. **Office Buildings:** AIoT is used in smart office buildings. In such buildings, usually, a network of smart environmental sensors is installed that detect number of people present in the office and adjust temperatures and lighting accordingly to improve energy efficiency. These smart buildings are often accessed through facial recognition technology using a combination of connected cameras and AI to compare images taken in real-time against a database to determine who should be granted access. this technique is also used by employers to maintain attendance of employees for mandatory meetings.
12. **Fleet Management:** AIoT is used in fleet management to monitor a fleet's vehicles, reduce fuel costs, track vehicle maintenance and identify unsafe driver behaviour. Through IoT devices such as GPS and other sensors and an artificial intelligence system, companies are able to manage their fleet better thanks to AIoT.

9.3.4 Benefits and Challenges of AIoT

The benefits of AIoT include the following:

1. **With AI, operational efficiency of IoT is increased.** AIoT devices analyse data to reveal patterns and insights and automatically adjusts system operations.
2. **On the fly decision-making: With AIoT and edge computing, vital decisions can be taken instantly. Therefore, data generated by the devices are** analysed to identify points of failure/concerns to make appropriate adjustments as and when required.
3. **Reduced workload: Since data is analysed by AI and not humans, lot of time and money spent for this time is saved.**
4. **Scalability:** The number of devices connected to an IoT system can be increased anytime as per requirements to optimize existing processes or introduce new features. IoT devices range from mobile devices and high-end computers to low-end sensors. Low-end sensors offer most of the data. An AI-powered IoT ecosystem analyses and summarizes the data from one device before transferring it to other devices. This not

only reduces large volumes of data to a handy level but also enables connecting a large number of IoT devices.

5. **Boosting Operational Efficiency:** AIoT instantly analyses constant streams of data to predict the operational conditions and identify the parameters to be modified to obtain better results. Thus, intelligent IoT can deduce which processes are redundant and time-consuming, and which tasks can be fine-tuned to enhance efficiency. For example, Google uses AIoT to reduce its data centre cooling costs.
6. **Better Risks Management:** Pairing AI with IoT facilitates businesses to understand as well as predict a wide variety of risks and automate process to provide a prompt response. This helps them to handle financial loss, ensure employee safety and prevent potential cyber threats. For example, Fujitsu ensures safety of the employees by using AI for analysing data coming from connected wearable devices.
7. **Triggering New and Enhanced Products and Services:** With NLP, users can easily communicate with devices. With this human-machine interaction, IoT and AI can together be used to create new products and services. Even existing products and services can be enhanced by quickly processing and analyzing real-time data. For example, Rolls Royce uses AI technologies in the implementation of IoT-enabled airplane engine maintenance to identify patterns and gain useful operational insights.
8. **Eliminates Costly Unplanned Downtime:** In industrial manufacturing and in some fields like offshore oil and gas exploration, equipment breakdown can result in costly unplanned downtime. Predictive maintenance with AI enabled IoT predicts equipment failure well in advance and schedule orderly maintenance procedures. This reduces the side effects of downtime. For example, Deloitte, integrates AI and IoT for the following:
 1. Reducing 20%–50% of their time spent in maintenance planning
 2. Increasing 10%–20% of equipment availability and uptime
 3. Reducing 5%–10% amount spent as maintenance costs

All this portrays a rosy picture but do not forget that there are always two faces of the same coin. AIoT may fail. For example, autonomous delivery robots that fail might cause delays in product delivery; smart retail stores may lead to accidentally stealing of a product if it fails to read a customer's

face, a road accident can be caused if an autonomous vehicle fail to read its surroundings, like a red light on traffic signal.

9.3.5 Future of AIoT

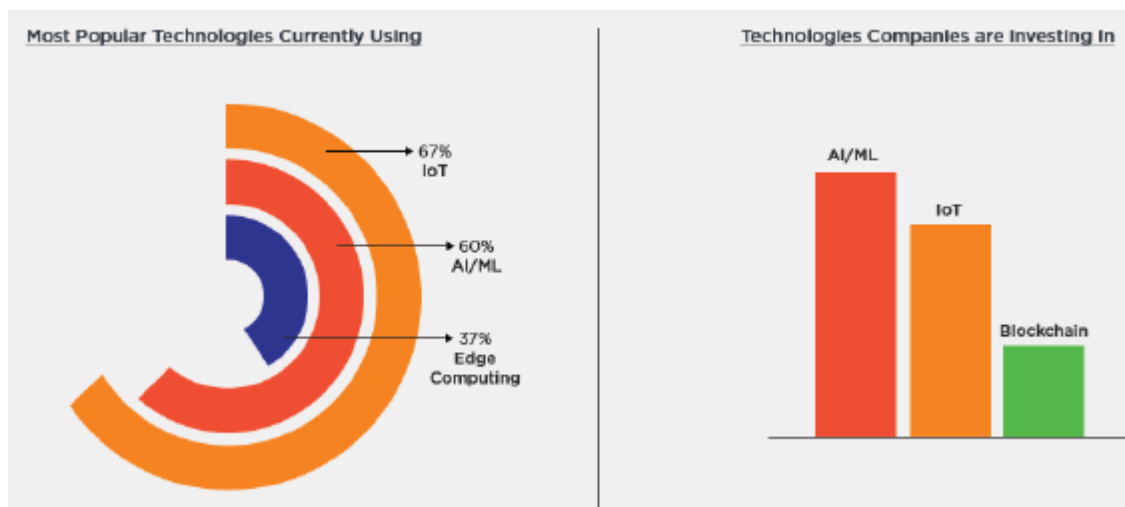
Integration of AI and IoT (refer **Fig. 9.14**) creates a much smarter system that makes accurate judgments without human intervention. AIoT will get a tremendous boost with 5G networks that is designed to enable faster transfer of large data files through higher bandwidth and lower latency. AIoT can revolutionize supply chains and delivery models and solve existing operational problems very efficiently. Several businesses have already adopted AI and IoT as part of their processes and products. In fact, AI and IoT are the top technologies in which companies are investing money to increase their operational efficiency and provide a competitive advantage. The graph in **Fig. 9.15** depicts the details.



Credit: elenabs1 / Shutterstock



Credit: marigranula / 123RF

FIGURE 9.14 An AIoT Machine**FIGURE 9.15** Visualization of emerging trends

It is surprising to note that every day, around one billion gigabytes of data is being generated by IoT devices. By 2025, it is estimated that 42 billion IoT-connected devices will be present globally and with growth in number of devices and network, the data generated is set to rise exponentially. With more data, there will be more challenges than opportunities.

With AI, IoT networks and devices can learn from past decisions, predict future activity, and continuously improve their decision-making capabilities and thus performance. AI allows the devices to ‘think for themselves’ by interpreting data without the delays and congestion that occur from data transfers.

9.4 Edge Computing

Edge computing is a distributed information technology (IT) architecture in which client data is processed at the periphery of the network, as close to the originating source as possible.

We have seen that data is the lifeline of any business as it provides valuable business insight and supports real-time control over critical business operations. Today, every business generates oceans of data, most of it coming from sensors and IoT devices operating in real time. This virtual flood of data is also changing the way computations are done. Traditional computing relies on centralized processing but bandwidth limitations, latency issues and unpredictable network disruptions poses a big challenge for hassle free movement of data and results. Edge computing offers a solution to overcome these challenges.

Edge computing moves some portion of storage and computations out of the central data centre and closer to the source of the data. This not only avoids transmission of raw data to central server but also facilitates data processing and analysis at the point where the data is actually generated—be it a retail store, a factory floor or a sprawling utility or across a smart city. Only the results of the computations that may include real-time business insights, equipment maintenance predictions or other actionable answers, is sent to the central processor for review.

Apart from efficiency benefits, edge computing also seems to be the only viable option available today. These days, the number of devices connected to the Internet, the volumes of data being produced by them is growing too fast to be handled by the traditional data centers. Gartner predicted that by 2025, 75% of the data will be created outside of centralized data centres. Movement of this big data amidst congestion and disruption is not possible.

So, IT architects have shifted focus from the central data centre to the logical *edge* of the infrastructure by taking storage space and computing resources from the data centre and moving them to locations where the data is generated. Therefore, they believe, ***‘if you can’t get the data closer to the data centre, get the data centre closer to the data.’***

A few applications that prefer edge computing include the following:

1. Retail environments combine video surveillance of the showroom floor with actual sales data to analyse the trend in consumer demand.
2. Predictive analytics that guide equipment maintenance and repair before actual defects or failures occur.
3. Utilities like water treatment or electricity generation.

Fog computing environments produce massive amounts of sensor or IoT data generated across large physical areas that are bigger than an edge. Examples include smart buildings, smart cities or even smart utility grids. In a smart city, data can be used to track, analyse and optimize the public transit system, municipal utilities, city services and guide long-term urban planning. A single edge deployment simply isn't enough to handle such a load, so here fog computing comes into picture.

9.4.1 Why is Edge Computing Important?

Consider the constant increase in demand of self-driving cars. These vehicles rely on intelligent traffic control signals which in turn need to produce, analyze and exchange data in real time. Multiply this requirement by huge numbers of autonomous vehicles to get a picture of how much data will be generated and processed in just one second. Clearly, this calls for a fast and responsive network and edge as well as fog computing addresses the concerns of a traditional computing system that include the following.

Bandwidth

Bandwidth refers to the amount of data that can be transmitted across a network over time. It is expressed in bits per second.

Every network has a limited bandwidth. Wireless networks have stringent limits of bandwidth as compared to wired networks. This means that only a limited data can be exchanged by a finite number of devices across the network.

However, network bandwidth can be increased to accommodate more devices and data but with a significant cost.

Latency

Latency is the time needed to transfer data from one point to another on a network. Ideally communication must take place at the speed of light, but it is often limited by large physical distances and issues like network congestion. Delays in data transfer adversely analysis and decision-making processes thereby hampering the ability for a system to respond in real time. In certain situations, latency can even cost lives. For example, in case of autonomous vehicles.

Congestion

The Internet is a global ‘network of networks’ that connects billions of computers all over the world to facilitate data exchange. Volumes of data transferred through these devices overwhelm the network with large number of data packets thereby increasing network traffic. Above this, network outages (or server/machine downtime) further add to these problems, making the Internet of Things useless.

By deploying servers and storage at a point near to the location where data is generated, edge computing (refer **Fig. 9.16**) can enable many devices to communicate over a much smaller and more efficient LAN that provides sufficient bandwidth making latency and congestion issues virtually nonexistent.

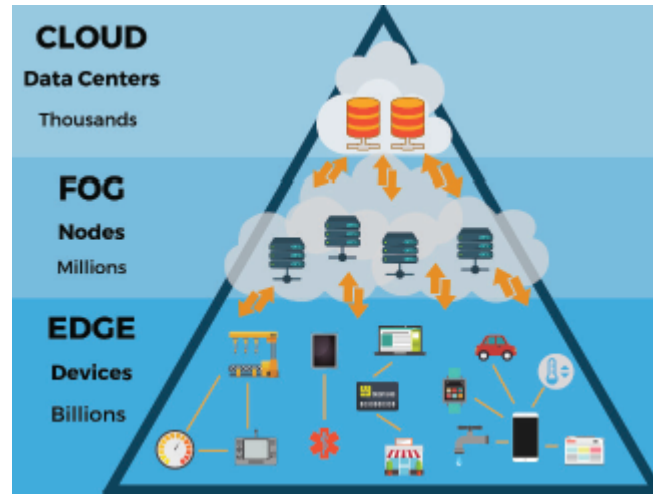


FIGURE 9.16 Edge Computing

Edge computing can reduce network costs, avoid bandwidth constraints, reduce transmission delays, limit service failures and provide better control over the movement of sensitive data.

In edge computing, local storage collects and protects the raw data, local servers perform essential data processing and analysis to make decisions in real time and then send the results, or just essential data, to the cloud or central data centre.

9.4.2 Edge Computing Use Cases and Examples

Technically, edge computing collects, filters, processes and analyses data ‘in-place’ at or near the network edge. Being a powerful technique to address vital issues including network bandwidth and congestion, it is being used in several domains including the following:

Manufacturing

Edge computing is used by manufacturers to monitor the manufacturing process, enable real-time analytics and identify production errors to improve quality of the products. Edge computing also supports addition of environmental sensors throughout the manufacturing plant to provide insight into how each product component is assembled and stored.

Manufacturing units also use this technology to make faster and more accurate business decisions regarding its operations.

Farming

Edge computing is used for indoor farming to make efficient use of sensors to track water use, nutrient density and determine optimal harvest. Data is collected and analysed to find the effects of environmental factors and improve the crop growing algorithms and ensure that crops are harvested in peak condition. Without edge computing it is very challenging to grow crops indoors without sunlight, soil or pesticides. The process reduces grow times by more than 60%. Hence, edge computing is a ray of hope to produce better results.

Network Optimization

Edge computing assists in optimizing the performance of a network by analysing data and identifying the most reliable, low-latency network path for each user's traffic.

Workplace Safety

Edge computing can combine and analyse data from on-site cameras, employee safety devices and various other sensors to help businesses keep a check on workplace conditions or ensure that employees follow established safety protocols. This is more important when either the workplace is at a remote location or is unusually dangerous (like construction sites or oil rigs).

Improved Healthcare

The healthcare industry generates massive amount of patient data that is often collected from devices, sensors and other medical equipment. This huge chunk of data requires edge computing to automate data storage and processing. Machine learning algorithms are used to access the data and extract information that requires immediate action to help patients avoid health incidents in real time.

Transportation

Autonomous vehicles require and produce anywhere from 5 TB to 20 TB every day. Data about location, speed, vehicle condition, road conditions, traffic conditions and other vehicles needs to be stored and analysed in real time, while the vehicle is in motion. In such a scenario, autonomous vehicle becomes an 'edge' to facilitate authorities and businesses manage vehicle fleets based on actual conditions on the ground.

Retail

Retail businesses generate enormous data volumes from surveillance, stock tracking, sales data and other real-time business details. In such a situation, edge computing helps to analyse this diverse data and identify business opportunities like an effective endcap or campaign, predict sales and optimize vendor ordering. Since retail businesses vary from one location to another, edge computing is preferred for processing the data locally at each store.

Immersive Experience

Real-time applications, such as AR/VR, cannot tolerate more than a few milliseconds of latency and can be extremely sensitive to jitter, or latency variation. In many applications, closed-loop automation is used to maintain high availability with response times in tens of milliseconds. This stringent requirement cannot be fulfilled without edge computing infrastructure.

Edge computing expands bandwidth capabilities thereby supporting new immersive applications including AR/VR, 4K video, and 360° imaging for verticals like healthcare. Caching and optimizing content at the edge is already becoming a necessity since protocols like TCP does not perform well when there are sudden changes in radio network traffic.

Edge computing infrastructure when used to access real-time data through network can reduce stalls and delays in video by up to 20% during peak viewing hours.

Education

Edge computing facilitates software-based education solutions that use on-device AI for personalized virtual assistance, natural language interaction and even augmented reality experiences. For example, ViewSonic digital whiteboard experience uses edge technology to re-create the classroom experience for students and teachers engaged in distance learning.

9.4.3 Benefits of Edge Computing

Besides solving critical issues like bandwidth limitations, excess latency and network congestion, edge computing also offers following benefits to the users.

Autonomy

In cases where connectivity is unreliable or bandwidth is restricted due to location's geography, edge computing is a viable solution. For example, oil rigs, ships at sea, remote farms, remote locations like a rainforest or desert extensively used edge computing to compute work on site, sometimes on the edge device itself. Like in a water purifier used in a remote village, that has water quality sensors installed in it, edge computing can save data in its local storage and then transmit it to a central point only when connectivity is available. By processing data locally, the amount of data to be sent is reduced significantly, thereby demanding less bandwidth or connectivity time than was required otherwise.

Data Sovereignty

Transmitting large volumes of data not only adds to technical woes but also poses a great threat to its security, privacy and other legal issues, especially when data crosses national and international boundaries. Edge computing keeps data close to its source and within the bounds of prevailing data sovereignty laws which defines how data should be stored, processed and exposed. Processing raw data locally secures sensitive data before it is exchanged with the cloud or primary data center (which may be in other jurisdictions).

Edge Security

Edge computing ensures data security. Since data and results obtained from analysing the data has to travel to reach the cloud or data center, steps are taken to protect the data through encryption. Edge deployment is often hardened against hackers and other malicious activities.

Moreover, edge computing can be thought of as similar to data centre computing due to the following reasons:

1. It includes compute, storage and networking resources.
2. Its resources may be shared amongst many users across several applications.
3. It can be benefitted by virtualization and abstraction of the resource pool.
4. It supports the ability to leverage commodity hardware.
5. It uses APIs to support interoperability.

9.4.4 Challenges of Edge Computing

Although edge computing provides a range of benefits across a multitude of use cases, the technology is far from foolproof. There are certain issues that can affect the adoption of edge computing at a large scale. These include the following.

Limited Capability

Edge computing deploys a variety of resources to provide a range of services. But this deployment of infrastructure at the edge must be clearly defined and justified to be effective. This is because even an extensive edge computing deployment serves a specific purpose at a pre-determined scale.

Connectivity

Edge computing still needs to transfer data to the cloud and hence even its capability is somewhat restricted by network connectivity.

Security

IoT devices are notoriously insecure, so edge computing deployment needs to take extra care for proper device management, policy-driven configuration enforcement and enforcement of security techniques in the computing and storage resources. Software patches and updates along with encryption ensure secure communications.

Data Lifecycles

Today, a lot of data is generated and collected by IoT devices. But most of this data is not required after a short span of time. So, it may not be stored for a long term. Therefore, it is necessary that the business has the capability to decide which data to keep and what to discard once the analyses process is over. The business also has the responsibility to retain and protect the required data in accordance with business and regulatory policies.

Greater Demands from Local Hardware

Edge computing requires more powerful and robust local hardware. For example, in a simple IoT camera a built-in computer is used to send its raw video data to a web server. But with edge computing, a more powerful computer with more processing power is required to run its own motion-detection algorithms.

Difficult to Scale

Scaling out edge servers to many small sites is more complicated than adding the equivalent capacity to a single data centre. Moreover, increase in physical locations also adds overhead to manage them.

Limited Support

Edge computing sites are usually remote with limited or non-availability of on-site technical expertise. If something fails on site, there must be an infrastructure in place that can be fixed easily by non-technical local labour. Further issues can be addressed centrally by a small number of experts located elsewhere.

Issues in Site Management

It takes more time, money, resources and efforts to manage all sites that are a part of edge computing. It is more challenging when software is implemented differently at each site.

More Prone to Attacks

Physical security of edge sites is often much lower than that of the central site. So, there is a greater risk of malicious or accidental attacks.

9.4.5 Edge Computing Implementation

Edge computing seems to be a straightforward idea but implementing it is a challenging exercise.

The first important step of deploying any technology successfully is the creation of a meaningful business and technical edge strategy which considers the need for edge computing. The ‘why’ answer demands a clear understanding of the technical and business problems that the organization needs to solve—overcome network constraints, maintain data sovereignty and so on.

An edge data center requires careful upfront planning and migration strategies.

The edge strategy starts with what the edge means, where it exists for the business and how it should benefit the organization. It should align with existing business plans and technology roadmaps.

As the project moves closer to implementation, evaluate hardware and software options carefully. Select a vendor by going through each vendor’s offering and evaluate it based on cost, performance, features, interoperability and support. The software tool should provide comprehensive visibility and control over the remote edge environment. Some of the vendors in edge computing are Adlink Technology, Cisco, Amazon, Dell EMC and HPE.

The actual deployment of an edge computing varies dramatically in scope and scale. No two edge deployments are the same. Because of these variations, edge strategy and planning become critical to the success of an edge project.

An edge deployment demands comprehensive monitoring. Deploying IT staff for this may not be possible, so edge deployments should be architected to provide tools that satisfy the following:

1. Ensure resilience, fault-tolerance and self-healing capabilities
2. Can be deployed remotely
3. Easy to configure
4. Offers comprehensive alerting and reporting about site availability or uptime, network performance, storage capacity and utilization and compute resources
5. Maintains security of the installation and its data. Tools that emphasize vulnerability management and intrusion detection and prevention should be installed.

9.4.6 Edge Computing, IoT and 5G Possibilities

Edge computing is evolving with new technologies and edge services are expected to become available worldwide by 2028. Today, edge computing is situation-specific, but it is expected to become more ubiquitous and support much more use cases in the future.

More number of devices and software (compute, storage and network) will be developed for edge computing.

Multivendor partnerships will provide better product interoperability and flexibility at the edge. For example, AWS and Verizon will together bring better connectivity to the edge.

Wireless communication technologies, such as 5G and Wi-Fi 6, will also boost edge deployments and utilization in the coming years. This will set the way for better and enhanced virtualization and automation capabilities, better vehicle autonomy and workload migrations to the edge, while making wireless networks more flexible and cost-effective.

The future will focus on the development of micro modular data centres (MMDCs). The MMDC is basically a data centre in a box, putting a complete data centre within a small mobile system that can be deployed closer to data—across a city or a region, to bring computing much closer to data.

Moreover, while AI algorithms that require large amounts of processing power are run on cloud-based services, AI chipsets that can do the work at the edge will help computations to be handled by edge systems.

Network function virtualization (NFV) is at the heart of edge computing application as it provides infrastructure functionality. Telecom operators are now transforming their service delivery models by running virtual network functions as part of, or layered on top of, an edge computing infrastructure.

9.5 Metaverse

The term ‘metaverse’ is a combination of two words—‘meta’ and ‘universe’ which means an anticipated future iteration of the Internet that is often referred to as Web 3.0. The evolution of the Internet from Web 2.0 to Web 3.0 will cause a tremendous boost of online 3-D or virtually integrated environments that provide users immersive experience in true sense (refer **Fig. 9.17**).



Credit: marigranula / 123RF

FIGURE 9.17 Metaverse

Metaverse is a combination of the virtual reality, augmented reality, mixed reality, blockchain technology, Internet of Things (IoT), 5G, **artificial intelligence (AI)** and spatial technologies that is accessed through a browser or headset. It is a device-independent technology that is not owned by a single vendor. Metaverse is an independent virtual economy that uses digital currencies and **non-fungible tokens (NFTs)**. It allows users to have real time interactions and experiences across distance.

Basically, metaverse is not just one type of experience. It actually encompasses a continuum of immersive digital experiences that will be available to users in the future to engage them in different activities in digital spaces. For example, a massive virtual reality multiplayer game can be accessed through a VR headset. Similarly, an integrated digital and physical spaces (like location-specific immersive digital content) that are often a need of business users, can be experienced using digital glasses or smartphones.

Many social media and tech companies like Meta Platforms (formerly known as Facebook) and Microsoft are investing heavily in Social VR to create platforms to allow people to connect socially and work remotely, respectively.

Metaverse is not one digital space but a combination of several digital spaces and experiences.

9.5.1 Use Cases of Metaverse

Metaverse is considered to be the next version of the Internet. The Internet started as individual bulletin board and rose to independent online destinations. Metaverse will make these destinations sites on a virtual shared space. Some examples are listed below.

Nikeland is a place to hang out, play and dress your avatar in virtual Nike products. More than 7 million people have already visited Nikeland.

In 2020, 12.3 million people attended a virtual concert by rapper Travis Scott, hosted in Fortnite. In this concert, the dancing figures were all people, connecting from different locations around the world.

The best part of Metaverse is that it allows users to mimic being together in ways that are not possible over Zoom. You can point to something to explain, use hand gestures or draw on a piece of paper or visit places together.

Metaverse makes it possible to collaborate with surgeons, or create a clay model for a new-car design.

Users can even purchase outfits and accessories for online avatars or buy digital land and construct virtual homes.

Virtual meetings in metaverse gives you the feeling of being together in a room. When participants need a work break, they can together play virtual games with colleagues from all over the world. This creates a real connection and builds social bonds.

Social media can move to the Metaverse, where users can interact through three-dimensional avatars.

Higher education, medical, military and other types of training can be delivered in a more immersive learning experience without creating additional infrastructure. Metaverse provides the framework to provide these services.

‘Metaverse will allow people to replicate or enhance their physical activities. This could happen by transporting or extending physical activities to a virtual world or by transforming the physical one,’ says Marty Resnick

Virtual malls use metaverse to extend their reach to an immersive shopping experience.

As of now, some aspects of the metaverse are integrated into existing Internet-enabled video games (like Second Life, Minecraft and Fortnite). These games offer immersive experiences in which players from around the world can engage simultaneously.

Did you know?

1. Microsoft is working hard to integrate ***Microsoft Teams into the metaverse.***
2. Roblox, an online gaming platform (particularly for youngsters), is creating an integrated set of virtual experiences for gamers.
3. Nvidia Corp is building an Omniverse platform that connects 3D virtual worlds in a shared universe. This can be used for creating simulations of buildings and factories.
4. In 2021, Snapchat launched AR glasses to augment users' experience of reality outside their app.
5. Epic Games developed one of the popular digital game- Fortnite. It is now creating virtual social experiences like music concerts or dance parties.

Gartner estimated 25% of people will spend at least one hour every day in the Metaverse for work, shopping, education, social media and/or entertainment by 2026.

Woolsey said, an organization must go for metaverse solution if the following conditions are satisfied.

1. The target customers are 15–30-year-olds. Metaverse engages the audience and makes them stay consistent with the brand. Businesses can even build a store that sells virtual goods or deliver an event or other type of engagement.
2. The organization has a dispersed team and often goes for hybrid meetings, multi-location training programmes and institutional events that bring people together to build a community, affiliation and engagement.
3. The organization needs to explore its internal operations—industrial or operational scenarios that would otherwise be very expensive to build in real life. **Figure 9.18** shows some key elements of metaverse.

Auto companies usually use digital twins to perform a dummy crash test in a metaverse world, and then collaborate on model modifications in augmented reality.

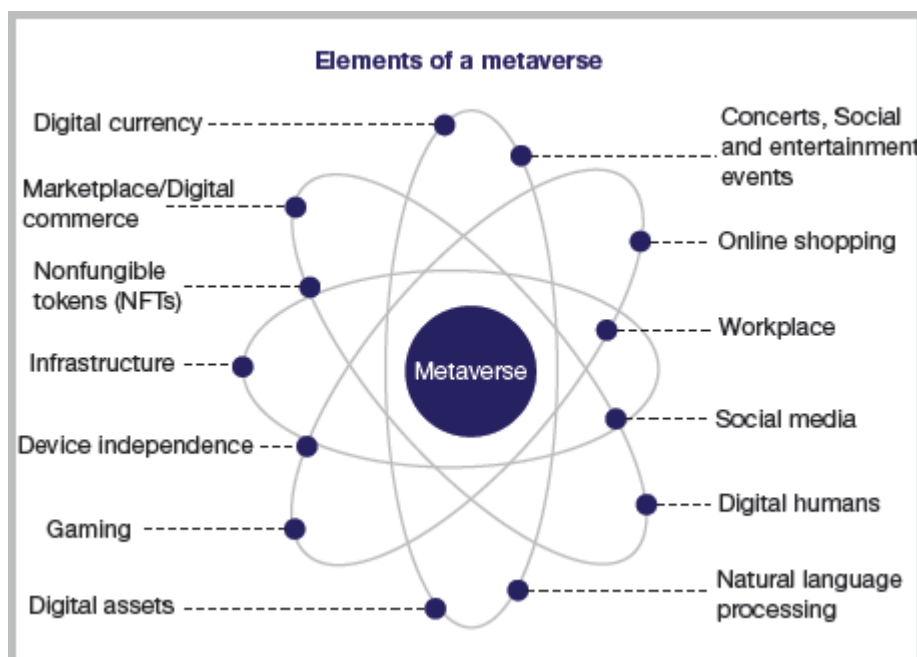


FIGURE 9.18 Elements of a metaverse

All in all, metaverse is seen as a significant business and financial opportunity for the tech industry and other sectors.

Woosely suggested that an organization must not dive into metaverse, just for the sake of doing metaverse. It should first analyse if any of the above-stated three scenarios are benefitted with this technology.

Once the technology is ready, first experiment it with a small audience so you can understand its actual value both qualitatively and quantitatively. Based on this learning, find the path that delivers strategic value and can be easily scaled.

However, the adoption of Metaverse technologies is still nascent and fragmented. Not many companies are planning to invest heavily in it, since it is still too early to determine which investments will be viable in the long term.

AR is used in Google Search to allow users to place 3D objects in their own space, allowing them to gain a better idea of an object's scale.

Many critics are skeptical about the growth of metaverse as it has mostly failed to live up to its hype and revenue expectations. Moreover, critics have exhibited concerns about the privacy and health implications of the metaverse vision. So, instead of investing, companies are first learning, exploring and preparing for a Metaverse.

Recommendations for Technology Innovation Leaders

1. Develop digital business strategies that leverage the built-in infrastructure and participants of the Metaverse.
2. Lead ideas and innovation that focuses on new opportunities and business models with the Metaverse.
3. Identify risks, privacy and security issues that are prevalent in this new persistent and decentralized environment.
4. Enumerate the outcomes, opportunities and obstacles faced by Metaverse.

Side-Effects of Metaverse

According to Indo-Asian News Service (IANS) report, Meta (formerly known as Facebook) is aware that VR can be a really 'toxic environment', especially for women and minorities. Harassment in digital spaces is not uncommon.

India's Role in Building Metaverse

The IANS report revealed that Mark Zuckerberg said that India and its vast pool of talent—the engineers, developers and creators and the whole vibrant startup ecosystem, is going to be a huge part of that journey. He is really excited about the role that India will play in building the future of Metaverse, which is going to be the successor to the mobile internet. India will have the largest app developer base in the world by 2024. It already has one of the largest 'Spark AR' developer communities.

9.5.2 Where is Metaverse Today?

As of now, Meta owns the leading producer of VR headsets, Oculus. Microsoft supports various enterprise use cases with its HoloLens MR smart glasses that were first released in 2016. Apple will be soon releasing its AR and VR headsets. HTC, Pico, MagicLeap and other manufacturers are also moving to manufacture devices to support this technology. At the same time, cascade technology architectures are shifting various computational loads across back-end server infrastructures to edge devices. For example, Unity's streams fully interactive real-time 3D environments. The job of rendering the environments is handled by their automatically scaling GPU server infrastructure.

Metaverse makes full use of ubiquitous computing experience to allow users leverage traditional devices like computers and mobile devices while experiencing with emerging immersive AR and VR wearables.

Like the internet, no one owns the metaverse. However, there are several key players in the space, and companies like Meta, Microsoft, Unity, Epic Games, Roblox and others are pouring billions of dollars into making the sci-fi dream come true.

The idea of metaverse was first coined by sci-fi author Neal Stephenson in his novel *Snow Crash*. Soon, it could become a reality because many large technology companies are betting that it will soon be as big as the Internet is today.



KEY TERMS

Internet of Things (IoT): A system of interrelated computing devices, machines, objects, animals or people that have unique identifiers for identification and also the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

Sensor: A device that detects and responds to some type of input that it receives from the physical environment.

Metaverse: A combination of the virtual reality, augmented reality, mixed reality, blockchain technology, Internet of Things (IoT), 5G, artificial intelligence (AI) and spatial technologies that is accessed through a browser or headset.

Edge computing: A distributed information technology (IT) architecture in which client data is processed at the periphery of the network, as close to the originating source as possible.

Expert system: A computer program that solves complex problems in a particular domain, at the level of extra-ordinary human intelligence and expertise.



RECAPITULATION POINTS

1. The user interface of an expert system must be very simple and intuitive so that even a non-expert user can communicate with the expert system to find solution to a complex problem.
2. Forward chaining starts from the known facts and rules. Backward chaining strategy is used for discovering cause or reason. For example, diagnosis of blood cancer in humans
3. AIoT enhances the capability of IoT by adding the power of machine learning algorithms to improve decision-making processes. IoT devices generate as well as collect data, that is then analysed using AI techniques to provide better insights into data.
4. The huge availability of address space in IPv6 has led to the development of the Internet of Things
5. Edge computing can reduce network costs, avoid bandwidth constraints, reduce transmission delays, limit service failures and provide better control over the movement of sensitive data.



REVIEW QUESTIONS

1. List the challenges faced by IoT devices.

2. What are the different types of sensors?
3. Why is Metaverse considered to be the next version of the Internet?
4. Write a short note on AIOT.
5. Discuss the capabilities, merits and demerits of an expert system.
6. List the steps involved in developing an expert system.
7. What challenges are faced by IoT today?
8. Explain the relevance of edge computing.
9. What is an expert system? Give its characteristics.
10. Explain the different components of an expert engine.
11. Explain the capabilities of an expert system.
12. Give any five applications of an expert system.
13. List the steps involved in developing an expert system.
14. Give some applications where AIoT and edge computing can be used.
15. Differentiate between the following:
 1. Forward and backward chaining
 2. Conventional system and expert system
 3. Human expert and artificial expert



1. A _____ is a device that detects and responds to some type of input that it receives from the physical environment.
2. All IoT-related services follow five basic steps to perform the assigned task—_____, _____, _____, _____ and _____.
3. Auto companies usually use _____ to perform a dummy crash test in a metaverse world.
4. A user interacts with expert system through the _____.
5. _____ chaining starts from the known facts and rules.
6. _____ is used in organic chemistry to detect unknown organic molecules.
7. _____ contains information and rules of a particular domain or subject.
8. _____ **knowledge** is based on practice, probability, evaluation and experiences.
9. _____ can lock out users from various devices.

10. _____ sensors detect forces per unit area in gases or liquids.
11. _____ sensors detect noise and measure distance between two objects.
12. _____ is the time needed to transfer data from one point to another on a network.



STATE TRUE OR FALSE

1. Expert systems are generic software.
2. More the amount of knowledge stored in the KB, better is the performance of the expert system.
3. CaDeT expert system could select specific software to generate a customized computer system based on user's requirements.
4. An expert system can deduce new facts from existing facts.
5. Metaverse is owned by Facebook.
6. More the amount of knowledge stored in the KB, better is the performance of the expert system.
7. Inference engines can also include explanation and debugging abilities.
8. Shell is an expert system without knowledge base.
9. Probabilistic Inference engine uses facts and rules to draw conclusions.
10. Forward chaining strategy is used for discovering cause or reason.
11. Results obtained by an expert system are highly reproducible.
12. The huge availability of address space in IPv6 has led to the development of the Internet of Things.
13. Particle sensors sense the movement or stoppage of parts, people, etc.
14. Metaverse allows users to interact through three-dimensional avatars.
15. Physical security of edge sites is often much lower than that of the central site.
16. Edge computing requires less powerful local hardware.



MULTIPLE CHOICE QUESTIONS

1. _____ is often referred to as Web 3.0.
 1. AIoT
 2. Edge Computing
 3. Metaverse
 4. IoT
2. _____ computing is a distributed IT architecture in which client data is processed at the periphery of the network, as close to the originating source as possible.
 1. Cognitive
 2. Affective
 3. Edge
 4. Transfer
3. Which expert systems was designed to find the bacteria causing infections like bacteraemia and meningitis?
 1. DENDRAL
 2. MYCIN
 3. PXDES
 4. CaDeT
4. _____ is an expert system that determines the type and level of lung cancer by analyzing picture(s) from the upper body.
 1. DENDRAL
 2. MYCIN
 3. PXDES
 4. CaDeT
5. _____ knowledge is based on practice, probability, evaluation and experiences.
 1. Factual
 2. Heuristic
 3. Both of these
 4. None of these
6. Identify the incorrect statement about expert systems.
 1. Expert systems cannot produce accurate output for inadequate knowledge base

2. Expert systems cannot refine knowledge base
 3. Expert systems can substitute human decision makers
 4. An expert system can predict results using past data.
7. Identify which of the following is not a characteristic of an expert system.
1. Reliable
 2. Highly responsive
 3. Limited memory
 4. Reduced cost
8. _____ is the brain of the expert system.
1. User interface
 2. Inference engine
 3. Knowledge base
 4. CPU
9. _____ represents that knowledge in the format accepted by the knowledge base.
1. User
 2. Domain expert
 3. Knowledge engineer
 4. Any of these
10. What can an expert system do?
1. Possess human capabilities
 2. Produce inaccurate output for inadequate knowledge base
 3. Refine knowledge base
 4. Use emotions to make decisions
11. _____ sensors sense objects passing within their field of detection.
1. Photoelectric
 2. Radiation
 3. Proximity
 4. Position



Fill Up

1. Sensor
2. Create, Communicate, Aggregate, Analyse and Act
3. digital twins
4. user interface
5. Forward
6. DENDRAL
7. knowledge base
8. Heuristic
9. Ransomware
10. Pressure
11. Ultrasonic
12. Latency

State True or False

1. False
2. True
3. False
4. True
5. False
6. True
7. True
8. True
9. False
10. False
11. True
12. True
13. False
14. True
15. True
16. False

Multiple Choice Questions

1. c
2. c
3. b
4. c
5. b
6. c
7. c
8. b
9. c
10. b
11. a

Previous chapter

< [8. Current Trends in Artificial Intelligence](#)

Next chapter

[10. AI Evolving with New Age Techniques](#) >

[Table of contents expanded](#)

10

AI Evolving with New Age Techniques

10.1 Fuzzy Logic

The word ‘fuzzy’ means the things that are not clear (or are vague). In real-life situations, we often encounter circumstances in which it is difficult to determine whether to go this way or that or given a statement, it is difficult to judge whether it is true or false. In such a situation, fuzzy logic provides many values between the true and false thereby offering the flexibility to find the best solution for the given situation at that particular point in time. As seen from **Fig. 10.1**, fuzzy logic contains multiple logical values that are the truth values of a variable.

In fuzzy logic, instead of saying just hot or cold, we can have multiple values like Certainly Yes, Possibly Yes, Cannot Say, Possibly No and Certainly No.

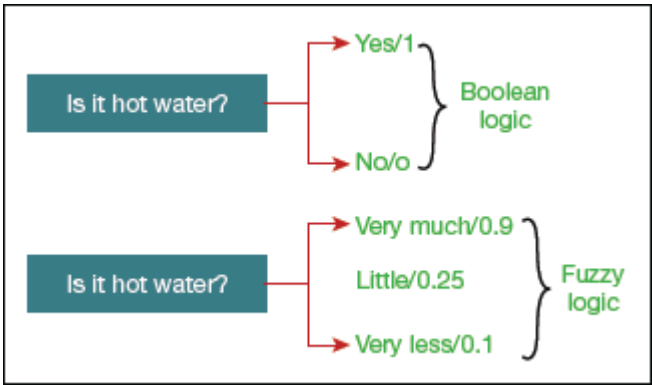


FIGURE 10.1 Example of Fuzzy Logic as compared to Boolean Logic

Fuzzy logic that is based on the **Fuzzy Set Theory** was introduced by **Lofti Zadeh** in **1965**. It provides a range of possibilities that used to be

generated by humans but were not offered by computers. Computer systems followed Boolean system that were limited to only two values- 0 or 1, where 1 denotes the absolute truth value and 0 denotes the absolute false value. Fuzzy system offers multiple possibilities between the 0 and 1, so that the solution can be partially false and partially true. It is a very important concept that is often implemented in micro-controllers, workstation-based or large network-based systems in their hardware or software for achieving the definite output.

10.1.1 Characteristics of Fuzzy Logic

1. It is flexible.
2. It is easy to understand and implement.
3. Fuzzy logic is useful for finding the best solution of problems that have approximate or uncertain reasoning.
4. It is used to develop functions which are non-linear of arbitrary complexity.
5. In fuzzy logic, everything is a matter of degree.
6. It allows any logical system to be easily fuzzified.
7. It is based on natural language processing.
8. It helps to improve the execution time of the algorithms.

10.1.2 Architecture of a Fuzzy Logic System

The architecture of the **fuzzy logic** system as shown in **Fig. 10.2** consists of four components which are discussed below.

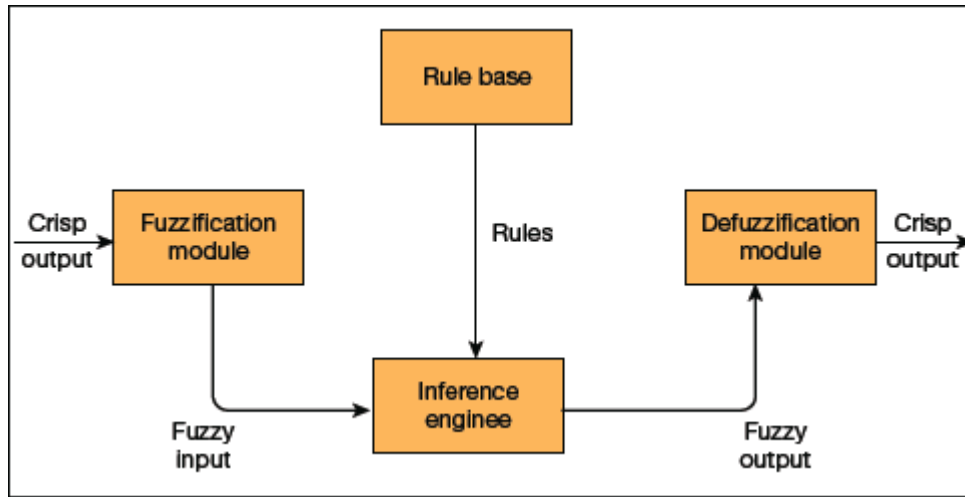


FIGURE 10.2 Architecture of fuzzy logic system

Rule base, as we know, is used for storing rules in the form of If-Then conditions. These conditions are added by the experts and are usually used by the decision-making systems for making vital decisions. Recent advances in Fuzzy theory offer effective methods for designing and tuning of fuzzy controllers. With these updates, the number of rules stored in the rule base has come down considerably.

Fuzzification is used for converting the crisp system inputs into fuzzy inputs. The crisp inputs usually come from sensors but cannot be directly processed by the inference engine. So, they first undergo fuzzification module where the crisp input is divided into following five states in any fuzzy logic system as follows:

1. Large positive (LP)
2. Medium positive (MP)
3. Small (S)
4. Medium negative (MN)
5. Large negative (LN)

Inference engine is the most important component of a fuzzy logic system that processes information. The engine interacts with the rules base to find a rule that is best applicable on the current fuzzy input. Once all rules are fired, the results of their application on fuzzy input are combined to give the control actions.

Defuzzification component accepts fuzzy inputs from the **inference engine** to transform them into a crisp value that is acceptable by the user.

10.1.3 Membership Function

The membership function, also known as indicator or characteristics function, was introduced by Zadeh. The function represents the graph of fuzzy sets to facilitate users to quantify the linguistic term. This graph maps each element of the fuzzy set, x to a value between 0 and 1.

For example, for the Fuzzy set B , the membership function for X is defined as:

$$\mu_B: X \rightarrow [0,1]$$

This implies that, in function X , each element of set B is mapped to a value between 0 and 1. This is called a degree of membership or membership value. It quantifies the degree of membership of the element in X to the fuzzy set A . When plotting these values on a graph, the universal set or the universe of discourse is marked on X-axis and the degrees of membership in the $[0, 1]$ interval is represented on the Y-axis.

There may be multiple membership functions that can fuzzify a numerical value. However, we usually use simple membership functions rather than complex functions as complexity of these functions do not improve the precision in the output.

10.1.4 Example of a Fuzzy Logic System

Let us consider a fuzzy logic system in an air conditioning system having five levels. The system adjusts the temperature of the air conditioner by comparing the room temperature and the target temperature value (as shown in **Fig. 10.3**). To develop such a system, follow the steps given below.

Step 1 – Define linguistic variables and terms

In this system, linguistic variables are input and output variables in the form of simple words or sentences. To specify room temperature, linguis-

tic terms can be cold, warm, hot, etc. So, we can write it as,

$$\text{Temperature (t)} = \{\text{very-cold, cold, warm, very-warm, hot}\}$$

Step 2 – Construct membership functions for them

The membership functions of temperature variable are given in **Fig. 10.4**.

Step 3 – Construct knowledge base rules

In this step, a matrix of room temperature values versus target temperature values that an air conditioning system is expected to provide is created. In the matrix given below, room temperature is written in the left-most (first) column and the first row of other columns gives the target temperature. Except the cells in the first column and the first row, other cells in the matrix specify the action that needs to be performed.

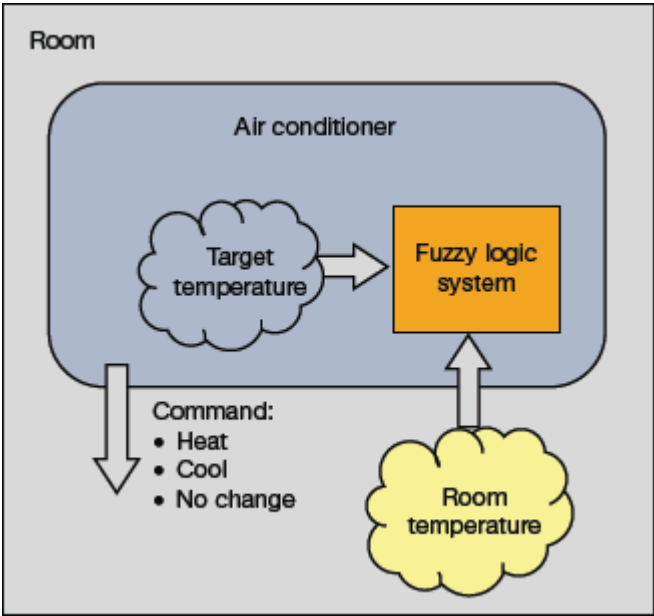


FIGURE 10.3 Fuzzy logic system in an air conditioning system

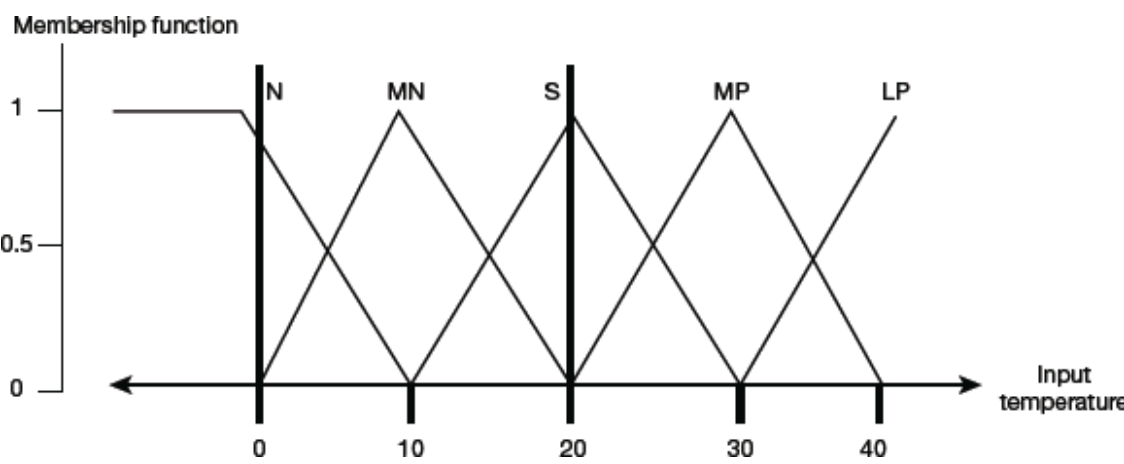


FIGURE 10.4 Membership functions of temperature variable

TABLE 10.1 Knowledge base for Temperature

Roomtemp. /Target	Very_cold	Cold	Warm	Hot	Very_hot
Very_Cold	No_Change	Heat	Heat	Heat	Heat
Cold	Cool	No_Change	Heat	Heat	Heat
Warm	Cool	Cool	No_Change	Heat	Heat
Hot	Cool	Cool	Cool	No_Change	Heat
Very_Hot	Cool	Cool	Cool	Cool	No_Change

For example, the highlighted cell states that if the room temperature is WARM and the expected temperature is HOT, then we need to increase HEATING.

Now, we will create a set of rules into the knowledge base in the form of IF-THEN-ELSE structures as shown in **Table 10.2**.

TABLE 10.2 Creating Rules

Sr. No.	Condition	Action
1	IF temperature = (Cold OR Very_Cold) AND target = Warm THEN	Heat
2	IF temperature = (Hot OR Very_Hot) AND target = Warm THEN	Cool
3	IF (temperature = Warm) AND (target = Warm) THEN	No_Change

Step 4 – Obtain fuzzy value

Fuzzy values are obtained by applying fuzzy set operations on the rules. The final fuzzy value is obtained by combining all results obtained by evaluating the rules.

Step 5 – Perform defuzzification

Defuzzification is performed based on the membership function for output variable as shown in **Fig. 10.5**.

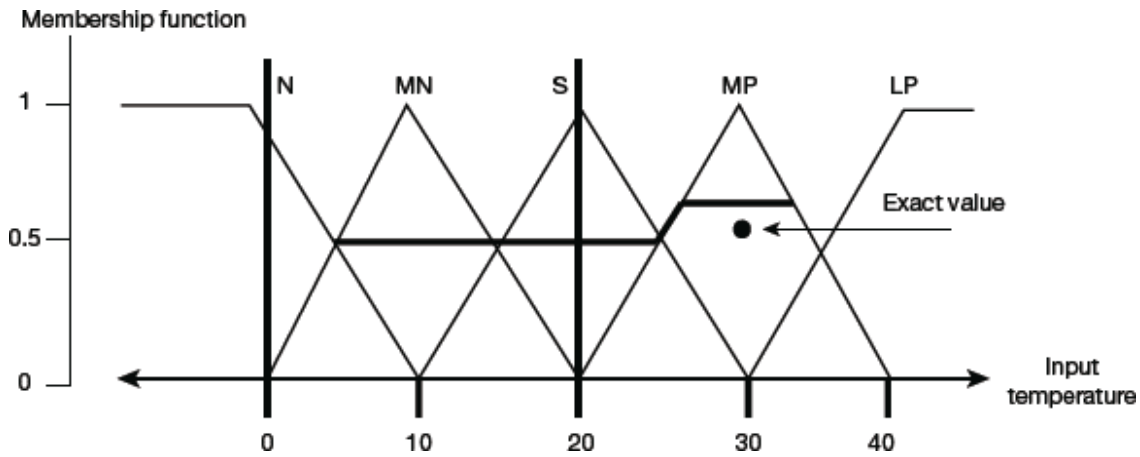


FIGURE 10.5 Membership function for output variable in case of Defuzzification

10.1.5 Classical and Fuzzy Set Theory

Before going into the Fuzzy set theory, let us first quickly revise the set theory which we are all already familiar with.

10.1.5.1 Set Theory

A set is a collection of unordered or ordered elements. For example, we have a set of, all-natural numbers, even numbers, students in a class, rivers in a country, etc. these sets can be classified in various categories as shown in **Fig. 10.6**.

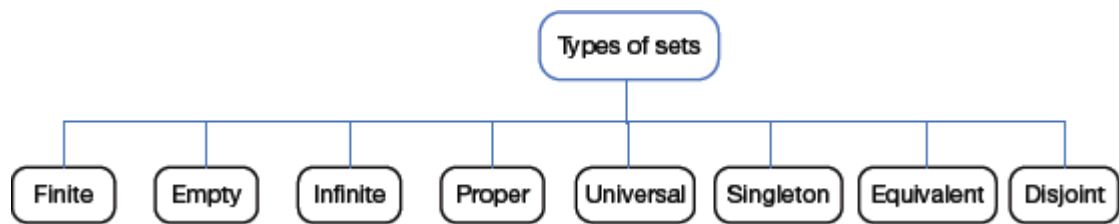


FIGURE 10.6 Set Theory

However, a classical set is a collection of distinct elements and has crisp boundaries. The classical set is defined in such a way that the universe of discourse (set of all possible values, also known as universal set) is divided into two groups—**members** and **non-members**. Therefore, such a set does not support **partial membership**. We can represent a classical set either in roaster form or in set-builder form.

Roaster form, also known as a tabular form, denotes a set by enclosing comma-separated members of the set within the brackets (as given below):

Set_name = {element1, element2, element3, ..., element N}

For example, set of Natural Numbers can be written as: $\mathbf{N} = \{1, 2, 3, 4, 5, 6, 7, \dots, n\}$.

Similarly, set of Even Numbers less than 20 can be written as,

$\mathbf{X} = \{2, 4, 6, 8, 10, 12, 14, 16, 18\}$.

Set builder form defines a set by stating the common properties of elements in the set. It can be denoted as:

$\mathbf{A} = \{x : p(x)\}$

For example, the set $\{2, 4, 6, 8, 10, 12, 14, 16, 18\}$ is written as:

$\mathbf{B} = \{x : 2 \leq x < 20 \text{ and } (x\%2) = 0\}$

10.1.5.2 Operations on Classical Set

We can perform different operations on classical sets like union, intersection, difference and complement.

Union ($A \cup B$)

The result of this operation generates a set of those elements which exist in either appear in A or in B or in both. This means that the union operation combines all the elements from both the sets in a new set. For this reason, union operation is also known as a Logical OR operation on the set. It is denoted as follows:

$$A \cup B = \{ x \mid x \in A \text{ OR } x \in B \}.$$

For example, given

Set A = {0, 1, 2, 3}

Set B = {1, 2, 3, 4, 5}, then

$$A \cup B = \{0, 1, 2, 3, 4, 5\}$$

Intersection ($A \cap B$)

The intersection or Logical AND operation returns a new set of those elements which are present in both set A and in set B. The operation is described as, $A \cap B = \{ x \mid x \in A \text{ AND } x \in B \}$.

For example, given

Set A = {0, 1, 2, 3}

Set B = {1, 2, 3, 4, 5}, then

$$A \cap B = \{1, 2, 3\}$$

Difference Operation ($A - B$)

The difference operation returns a new set of those elements which exist only in set A but not in set B. It is described as follows:

$$A - B = \{ x \mid x \in A \text{ AND } x \notin B \}$$

For example, given

Set A = {0, 1, 2, 3}

Set B = {2, 3, 4, 5}, then

A - B = {0,1}

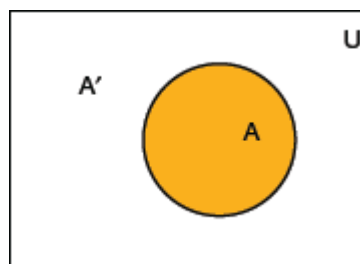
Complement Operation (A')

The complement operation is applied on a single set. It returns a set of elements which do not exist in set A and can be represented as,

$$A' = \{x \in U : x \notin A\}$$

Here, U is the Universal set. Therefore, we can say that **complement of a set** is a subset of the universal set as it has all the elements of the universal set except those that are not present in the given set. In other words, the complement of a set A is the difference between the universal set and set A. that is $A' = U - A$.

For example, if the universal set consists of all prime numbers up to 25, that is, $U = \{2, 3, 5, 7, 11, 13, 17, 19, 23\}$ and,



A = {2, 3, 5} then,

$$A' = (U - A)$$

$$= \{2, 3, 5, 7, 11, 13, 17, 19, 23\} - \{2, 3, 5\}$$

$$= \{7, 11, 13, 17, 19, 23\}$$

10.1.5.3 Properties of Classical Set

In this section, we will discuss some basic properties of a classical set.

Commutative Property

The property states that operations on a set can be performed irrespective of order of the sets. That is,

$$A \cup B = B \cup A$$

$$A \cap B = B \cap A$$

For example, given:

$A = \{1, 2, 3\}$, and $B = \{2, 3, 4\}$, then

$A \cup B = \{1, 2, 3, 4\}$ and $B \cup A = \{1, 2, 3, 4\}$

Hence, proved. Similarly, let us prove it in case of intersection.

$A \cap B = \{2, 3\}$ and $B \cap A = \{2, 3\}$

Associative Property

Associativity property allows us to perform the operations by grouping the operands and keeping them in similar order.

$$(A \cup B) \cup C = A \cup (B \cup C)$$

$$(A \cap B) \cap C = A \cap (B \cap C)$$

For example, given:

$A = \{1, 2, 3\}$, $B = \{2, 3, 4\}$ and $C = \{5, 6\}$, then

$A \cup B = \{1, 2, 3, 4\}$ and $(A \cup B) \cup C = \{1, 2, 3, 4, 5, 6\}$

$B \cup C = \{2, 3, 4, 5, 6\}$ and $A \cup (B \cup C) = \{1, 2, 3, 4, 5, 6\}$

Hence, proved. Now,

$A \cap B = \{2, 3\}$ and $(A \cap B) \cap C = \varnothing$

$B \cap C = \varnothing$. Therefore, $A \cap (B \cap C) = \varnothing$

Idempotency Property

It is defined as

$$A \cup A = A$$

$$A \cap A = A$$

For example, given $A = \{1, 2, 3\}$,

$$A \cup A = \{1, 2, 3\} \cup \{1, 2, 3\} = \{1, 2, 3\}$$

Similarly, $A \cap A = \{1, 2, 3\} \cap \{1, 2, 3\} = \{1, 2, 3\}$

Absorption Property

This property when defined on two sets states that,

$$A \cup (A \cap B) = A$$

$$A \cap (A \cup B) = A$$

For example, given:

$A = \{1, 2, 3\}$, $B = \{2, 3, 4\}$, then

$$A \cup (A \cap B) = \{1, 2, 3\} \cup \{2, 3\} = \{1, 2, 3\}$$

Similarly, $A \cap (A \cup B) = \{1, 2, 3\} \cap \{1, 2, 3, 4\} = \{1, 2, 3\}$

Distributive Property

Given three finite sets A, B and C, distributive property is defined as,

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

For example, given:

$A = \{1, 2, 3\}$, $B = \{2, 3, 4\}$ and $C = \{5, 6\}$, then

$$A \cup (B \cap C) = \{1, 2, 3\} \cup \{\varnothing\} = \{1, 2, 3\} \text{ (because } A \cup \varnothing = A\text{)}$$

$$(A \cup B) \cap (A \cup C) = \{1, 2, 3, 4\} \cap \{1, 2, 3, 5, 6\} = \{1, 2, 3\}$$

Hence, proved. Now,

$$A \cap (B \cup C) = \{1, 2, 3\} \cap \{2, 3, 4, 5, 6\} = \{2, 3\}$$

$$(A \cap B) \cup (A \cap C) = \{2, 3\} \cup \{\varnothing\} = \{2, 3\}$$

Identity Property

For a given finite set A and Universal set X, identity property can be given as,

$$\begin{aligned} A \cup \{\varnothing\} &= A \\ A \cap X &= A \\ A \cap \{\varnothing\} &= \{\varnothing\} \\ A \cup X &= X \end{aligned}$$

For example, given $A = \{1, 2, 3\}$ and $X = \{1, 2, 3, 4, 5, 6\}$

$$A \cup \{\varnothing\} = \{1, 2, 3\} \cup \{\varnothing\} = \{1, 2, 3\}$$

$$A \cap X = \{1, 2, 3\} \cap \{1, 2, 3, 4, 5, 6\} = \{1, 2, 3\}$$

$$A \cap \{\varnothing\} = \{1, 2, 3\} \cap \{\varnothing\} = \{\varnothing\}$$

$$A \cup X = \{1, 2, 3\} \cup \{1, 2, 3, 4, 5, 6\} = \{1, 2, 3, 4, 5, 6\}$$

Transitive Property

This property state that for the finite sets A, B and C, if A is a subset of B and B is a subset of C, then A is also a subset of C. This can be written as,

$$\text{IF } A \subseteq B \subseteq C, \text{ then } A \subseteq C$$

For example, given

$$A = \{1, 2, 3\}, B = \{1, 2, 3, 4, 5\} \text{ and } C = \{1, 2, 3, 4, 5, 6\}, \text{ then}$$

Since all the elements of A are a part of set B, A is said to be a subset of B.

Since all the elements of set B are a part of set C, B is said to be a subset of C.

Moreover, we see that A is also a subset of C.

Involution Property

The property states that for any finite set A,

$$\overline{\overline{A}} = A$$

For example, given $A = \{1, 2, 3\}$ and $U = \{1, 2, 3, 4, 5, 6\}$

$$A' = \{1, 2, 3, 4, 5, 6\} - \{1, 2, 3\} = \{4, 5, 6\}$$

$$A'' = \{1, 2, 3, 4, 5, 6\} - \{4, 5, 6\} = \{1, 2, 3\}$$

De Morgan's Law

Given two sets A and B, the De Morgan's law says that,

$$(A \cup B)' = A' \cap B'$$

$$(A \cap B)' = A' \cup B'$$

For example, given;

$$A = \{1, 2, 3\}, B = \{2, 3, 4\} \text{ and } U = \{1, 2, 3, 4, 5, 6\}$$

$$(A \cup B)' = \{\{1, 2, 3\} \cup \{2, 3, 4\}\}' = \{1, 2, 3, 4\}' = \{5, 6\}$$

$$A' \cap B' = \{4, 5, 6\} \cap \{1, 5, 6\} = \{5, 6\}$$

Hence, proved. Similarly,

$$(A \cap B)' = \{2, 3\}' = \{1, 4, 5, 6\}$$

$$A' \cup B' = \{4, 5, 6\} \cup \{1, 5, 6\} = \{1, 4, 5, 6\}$$

10.1.6 Fuzzy Set

The classical set theory is the subset of Fuzzy set theory. Or we can say that the fuzzy set theory is a generalization of the classical theory of set (i.e., crisp set). Introduced by Lofti A. Zadeh in 1965, the entire fuzzy logic is based on the fuzzy set theory.

A fuzzy set denoted by the tilde ($\sim$) character is a collection of values which exist between 0 and 1. In the fuzzy set, there is also a concept of partial membership.

A fuzzy set ($\tilde{A}$) is a pair of U and M , where U is the Universal set and M is the membership function which takes on values in the interval $[0, 1]$. The universal set (U) is also denoted by Ω or X .

$$\tilde{A} = \{(x, m_{\tilde{A}}(x)) \mid x \in X\}$$

10.1.6.1 Operations on Fuzzy Set

Given two fuzzy sets, and a universal set X , with member functions,

$$m_{\tilde{A}}(x) \text{ and } m_{\tilde{B}}(x)$$

The operations defined on the fuzzy set are as follows:

Union Operation

The union operation of a fuzzy set is written as,

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$$

For example, given:

$$A = \{(X_1, 0.2), (X_2, 0.4), (X_3, 0.6), (X_4, 0.8)\}$$

$$B = \{(X_1, 0.1), (X_2, 0.5), (X_3, 0.7), (X_4, 0.3)\} \text{ then,}$$

$$A \cup B = \{(X_1, 0.2), (X_2, 0.5), (X_3, 0.7), (X_4, 0.8)\}$$

Justification:

For X_1	For X_2	For X_3	For X_4
$\mu_{A \cup B}(X_1) = \max$ $(\mu_A(X_1), \mu_B(X_1))$	$\mu_{A \cup B}(X_2) = \max$ $(\mu_A(X_2), \mu_B(X_2))$	$\mu_{A \cup B}(X_3) = \max$ $(\mu_A(X_3), \mu_B(X_3))$	$\mu_{A \cup B}(X_4) = \max$ $(\mu_A(X_4), \mu_B(X_4))$
$\mu_{A \cup B}(X_1) = \max$ $(0.2, 0.1)$	$\mu_{A \cup B}(X_2) = \max$ $(0.4, 0.5)$	$\mu_{A \cup B}(X_3) = \max$ $(0.6, 0.7)$	$\mu_{A \cup B}(X_4) = \max$ $(0.8, 0.3)$
$\mu_{A \cup B}(X_1) = 0.2$	$\mu_{A \cup B}(X_2) = 0.5$	$\mu_{A \cup B}(X_3) = 0.7$	$\mu_{A \cup B}(X_4) = 0.8$

Intersection Operation

The intersection operation of a fuzzy set is written as,

$$\mu_{A \cap B}(x) = \min (\mu_A(x), \mu_B(x))$$

For example, given:

$$A = \{(X_1, 0.2), (X_2, 0.4), (X_3, 0.6), (X_4, 0.8)\}$$

$$B = \{(X_1, 0.1), (X_2, 0.5), (X_3, 0.7), (X_4, 0.1)\} \text{ then,}$$

$$A \cap B = \{(X_1, 0.1), (X_2, 0.4), (X_3, 0.6), (X_4, 0.1)\}$$

Justification:

For X_1	For X_2	For X_3	For X_4
$\mu_{A \cap B}(X_1) = \min$ $(\mu_A(X_1), \mu_B(X_1))$	$\mu_{A \cap B}(X_2) = \min$ $(\mu_A(X_2), \mu_B(X_2))$	$\mu_{A \cap B}(X_3) = \min$ $(\mu_A(X_3), \mu_B(X_3))$	$\mu_{A \cap B}(X_4) = \min$ $(\mu_A(X_4), \mu_B(X_4))$
$\mu_{A \cap B}(X_1) = \min$ $(0.2, 0.1)$	$\mu_{A \cap B}(X_2) = \min$ $(0.4, 0.5)$	$\mu_{A \cap B}(X_3) = \min$ $(0.6, 0.7)$	$\mu_{A \cap B}(X_4) = \min$ $(0.8, 0.1)$
$\mu_{A \cap B}(X_1) = 0.1$	$\mu_{A \cap B}(X_2) = 0.4$	$\mu_{A \cap B}(X_3) = 0.6$	$\mu_{A \cap B}(X_4) = 0.1$

Complement Operation

The complement operation of fuzzy set is defined by:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x),$$

For example, given:

$$A = \{(X_1, 0.5), (X_2, 0.4), (X_3, 0.2), (X_4, 0.1)\}, \text{ then}$$

$\bar{A} = \{(X_1, 0.5), (X_2, 0.6), (X_3, 0.8), (X_4, 0.9)\}$

Justification:

For X_1

$\mu_{\bar{A}}(X_1) = 1 - \mu_A(X_1)$
 $\mu_{\bar{A}}(X_1) = 1 - 0.5$
 $\mu_{\bar{A}}(X_1) = 0.5$

For X_3

$\mu_{\bar{A}}(X_1) = 1 - \mu_A(X_1)$
 $\mu_{\bar{A}}(X_1) = 1 - 0.2$
 $\mu_{\bar{A}}(X_1) = 0.8$

For X_4

$\mu_{\bar{A}}(X_1) = 1 - \mu_A(X_1)$
 $\mu_{\bar{A}}(X_1) = 1 - 0.1$
 $\mu_{\bar{A}}(X_1) = 0.9$

TABLE 10.3 Differences between classical set theory and fuzzy set theory

Classical Set Theory	Fuzzy Set Theory
This theory is applicable on sets having sharp boundaries.	This theory is applicable on sets that do not have sharp boundaries.
This set theory is defined by exact boundaries that can be either 0 or 1	This set theory is defined by ambiguous boundaries.
There is no uncertainty about the boundary's location of a set.	There is an uncertainty about the boundary's location of a set.
This theory is used in the design of digital systems.	It is mainly used for designing fuzzy controllers.

10.1.7 Applications of Fuzzy Logic

Fuzzy logic is widely used in different application areas in **businesses** for making vital decisions.

Automotive systems apply fuzzy logic to control brakes in hazardous cases depending on the speed, acceleration, wheel speed and acceleration of the car. It is also used to select the gear based on engine load, driving style and road conditions.

In auto-transmission systems, fuzzy logic is used to improve the efficiency by controlling the fuel injection and ignition based on throttle setting, cooling water temperature, RPM, etc.

Systems developed for defence use fuzzy logic for underwater target recognition and the automatic target recognition of thermal infrared images.

Fuzzy logic can be used to search fuzzy images.

Pattern recognition and classification systems use fuzzy logic for handwriting recognition.

In **microwave ovens**, fuzzy logic is used to set the power and cooking strategy.

Modern control systems like expert systems often use fuzzy logic.

Fuzzy logic is widely used to predict the stock market and for managing funds.

In chemical industry, fuzzy logic is used to control the pH and chemical distillation process.

In manufacturing industry, fuzzy logic is used for optimizing production.

In devices like vacuum cleaners, washing machines, heaters, air conditioners and humidifiers, fuzzy logic is used for controlling the timing of events. For example, an elevator uses fuzzy logic to reduce waiting for time-based on passenger traffic.

In a dish-washer, fuzzy logic is used for adjusting the cleaning cycle, rinse and wash strategies depending on the number of dishes and the amount of food served on the dishes.

Copy machine by Canon uses fuzzy logic for adjusting drum voltage based on picture density, humidity and temperature.

In a golf diagnostic system, fuzzy logic selects golf club based on golfer's swing and physique.

In a kiln, fuzzy logic is used to control the mixing of cement.

10.1.8 Advantages of Fuzzy Logic

IBM's Watson is a well-known AI system that uses variations of fuzzy logic and fuzzy semantics.

1. Fuzzy logic works similar to human reasoning.
2. The concept is easy to understand.
3. It does not need large memory as the algorithms can work well with fewer data.
4. Fuzzy logic is easily applicable in diverse fields.
5. It provides effective solutions to even highly complex problems.
6. Fuzzy set theory is based on the set theory of mathematics, so users can easily relate to its basic concepts.
7. It facilitates controlling micro-controllers and thus electronic devices.
8. Developing a system based on fuzzy logic takes less time than developing it using conventional methods.
9. Fuzzy logic is very flexible as it allows users to easily add and delete rules in the system.
10. Fuzzy logic supports the concept of partial truth, such as 0.9 or 0.5.
11. Fuzzy logic mimics real-life circumstances, where we rarely have statements that are absolutely true or absolutely false.
12. Fuzzy logic, when used by quantitative analysts, improves the execution of algorithms.
13. Algorithms based on fuzzy logic produce accurate results even with imprecise or inaccurate data.
14. Systems based on fuzzy logic are more robust as they are not dependent on precise inputs.
15. Given the information about how far the subject in a photograph is, a fuzzy logic system can focus the lens on it.

10.1.9 Disadvantages of Fuzzy Logic

1. The execution time of fuzzy logic systems is more than conventional systems, so these systems are often slow to generate output.
2. Output generated by fuzzy logic system is not 100% accurate.
3. In fuzzy logic system, there are multiple ways to solve a given problem. However, this may lead to ambiguity.
4. Problems that need highly accurate solutions should not be solved using fuzzy logic.
5. A fuzzy logic system needs a lot of testing for verification and validation.

6. The accuracy of results obtained from a fuzzy logic system depends on human expertise and knowledge.
7. The design of a fuzzy logic system is understandable only when it is simple.
8. Fuzzy systems lack the capability of machine learning and neural network type pattern recognition.

You must not use fuzzy logic in the following cases:

1. If it is difficult to map an input space to an output space
2. When application of common sense can easily solve the problem
3. Controllers work well without the use of fuzzy logic

10.1.10 Trends

Nowadays, the trend is to use fuzzy logic along with neurocomputing and genetic algorithms. A combination of all these techniques forms the basis of *soft computing*. *In soft computing, the goal is to exploit the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness and low solution cost.* At present, despite of the fact that soft computing systems are expensive, they are being preferred as they build AI system with MIQ (Machine IQ) much higher than that of conventional neuro-fuzzy systems which combine fuzzy logic and neurocomputing and induce rules from observations. An effective method developed by Dr Roger Jang for this purpose is called ANFIS (Adaptive Neuro-Fuzzy Inference System).

Fuzzy logic approximates human reasoning and balances the tradeoff between precision and significance (illustrated in **Fig. 10.7**). For example, when there is an unexpected alert of a mass falling on you, the alert need not be precise about the exact mass and speed as is not at all necessary to make a quick decision.

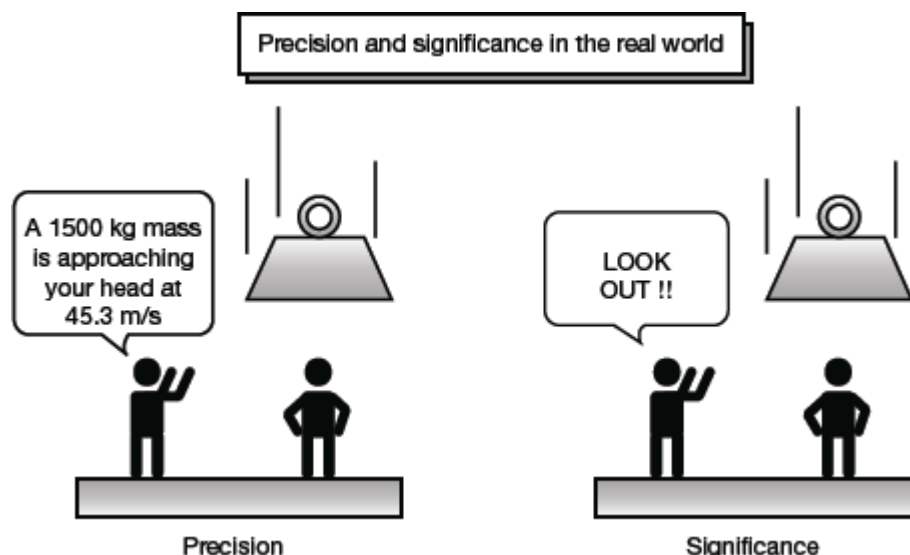


FIGURE 10.7

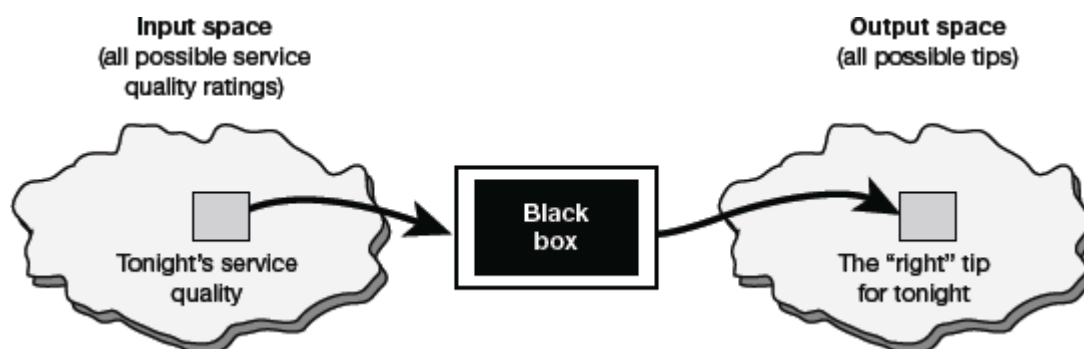


FIGURE 10.8 An input/output map for the tipping problem: “Given the quality of service, how much should I tip?”

A fuzzy system behaves like a black box that maps an input space to an output space. Consider the example given in **Fig. 10.8** where fuzzy logic is used to determine the appropriate amount of tip by mapping the input space of all possible restaurant service ratings to all possible tip values. Note that between the input and the output, there is a black box that may contain fuzzy systems, linear systems, expert systems, neural networks, differential equations, interpolated multidimensional lookup tables or any other algorithm.

10.2 Genetic Algorithms

Nature has always been a great source of inspiration to all mankind. Genetic algorithm is a subset of **evolutionary computation**. It is a search-based optimization technique that draws its concept from the

principles of **genetics and natural selection**. The main take away of these algorithms is that they can find optimal or near-optimal solutions to difficult optimization problems in research, and in machine learning problems. Genetic algorithms can find solutions to those problems which otherwise would take a lifetime to solve.

The term ‘optimization’ refers to the process of **making something better**. For this, the algorithm needs to find what input values will give the ‘best’ output values. Though, what ‘best’ means will vary from problem to problem, mathematically, it refers to maximizing or minimizing one or more objective functions, by varying the input values.

We know that the search space comprises all possible solutions or values which the inputs can take. The problem of optimization is to find those solutions or values in the search space that gives the optimal solution.

In genetic algorithms, the given problem has a **pool or a population of possible solutions**. Crossover and mutation are applied on these solutions to produce new children. This process is repeated over several generations. Centred around the Darwinian Theory of ‘Survival of the Fittest’, each individual (or candidate solution) thus produced is assigned a fitness value. Individuals having a higher fitness value are given a priority to mate and yield more ‘fitter’ individuals. This helps in the process of evolution that gives better solutions over generations, till stopping criterion is met.

Though genetic algorithms are randomized, they perform much better than random local search as they exploit historical information as well.

10.2.1 Advantages of Genetic Algorithms

1. In real-world problems, derivative information may not be available. Genetic algorithms work well without this information.

Derivative information means any notes, summaries, evaluations, analyses, **as well** as feedback, suggestions, improvements and other material derived by the Restricted Party Group from any of the confidential information.

2. Compared to traditional algorithms, genetic algorithms are fast and efficient. For example, some difficult problems like the Travelling Salesperson Problem (TSP) are widely applied to solve complex real-world problems like path finding, GPS Navigation system and VLSI Design. Delay in computing the 'optimal' path from the source to destination is not acceptable. Genetic algorithms come to rescue in such situations to get optimal results very quickly.
3. They can perform parallel computations.
4. They optimize continuous as well as discrete functions.
5. They give optimized solutions for multi-objective problems.
6. Instead of returning a single 'good' solution, it returns a list of all 'good' solutions.
7. They always find solution to a problem and this solution improves over time.
8. They are very helpful in problems that have a very large search space and involve a large number of parameters.
9. They can be used for solving NP Hard problems. There are a large set of NP Hard problems and they give optimal or near—optimal solutions for these problems in a reasonable amount of time.

NP Hard problems are the most powerful computing problems that take a very long time (even years!) to solve a problem.

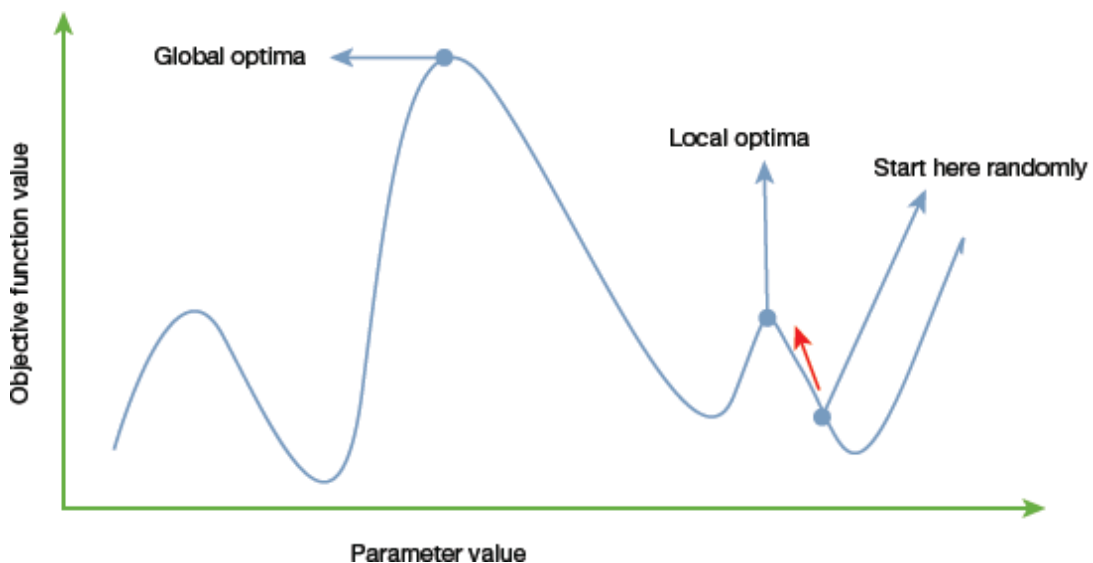


FIGURE 10.9 Global and Local Optima

10. Traditional calculus-based mathematical techniques (like gradient-based methods) start at a random point and move in the direction of the gradient until the top of the hill is reached. This technique is effi-

cient and works very well for functions having a single-peak. For example, the cost function in linear regression can easily be solved using gradient-based methods. However, in most real-world problems, there are many peaks and valleys. These problems cannot be solved using the traditional methods as they end-up being stuck at the local optima (refer to **Fig. 10.9**).

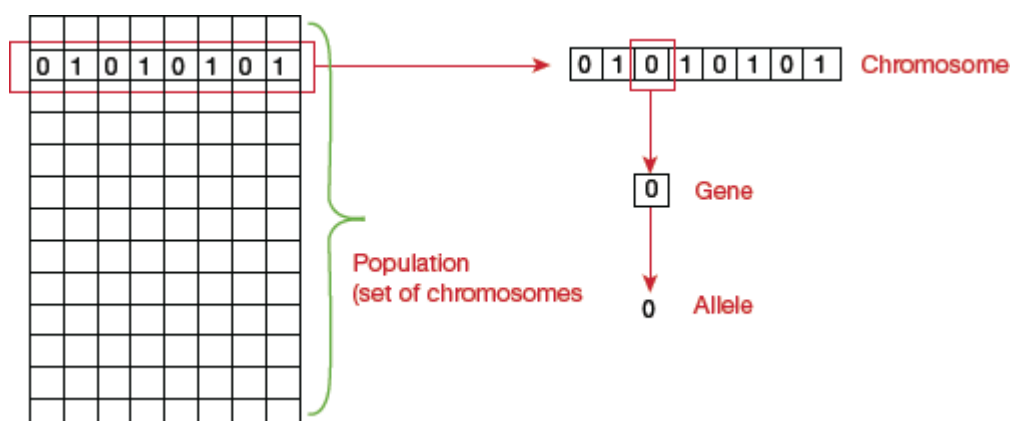
10.2.2 Limitations of Genetic Algorithms

1. Genetic algorithms are not suited for all problems such as those that are either very simple or those for which derivative information is available.
2. The algorithm is computationally expensive for some problems as fitness value is calculated repeatedly.
3. There is no guarantee on the optimality or the quality of the solution.
4. Genetic algorithms may not find the optimal solution if not implemented properly.

10.2.3 Basic Terminology

To understand genetic algorithms, let us familiarize with some basic terminology that will be frequently used.

1. **Population:** It is a subset of all the possible solutions to the given problem.
2. **Chromosomes:** A chromosome is the individual solution to the given problem.
3. **Gene:** A gene is one element position of a chromosome.
4. **Allele:** It is the value which a gene takes for a particular chromosome.



5. **Genotype:** Genotype is the population (solution) in the computation space that can be represented in a simple and easy-to-understand manner so that it can be easily manipulated using a computing system.
6. **Phenotype:** Phenotype is the population (solution) in the real-world solution space that can be represented in real world situations (refer **Fig. 10.10**).
7. **Decoding and Encoding:** For simple problems, the **phenotype and genotype** spaces are the same. But for complex, real-world problems, these spaces are different. While decoding is the process of transforming a solution from the genotype to phenotype space, encoding, on the other hand, transforms solution from the phenotype to genotype space. A genetic algorithm must perform decoding at a very fast pace as it has to be done repeatedly while calculating the fitness value. For example, in the 0/1 Knapsack problem, the phenotype space consists of solutions which has the item numbers of the items to be picked. But the genotype space is represented as a binary string of length n (where n is the number of items). A **0 at position x** indicates that the x^{th} item is not picked, whereas a 1 indicates that it was picked. Thus, in this case, genotype space is different from that of the phenotype.

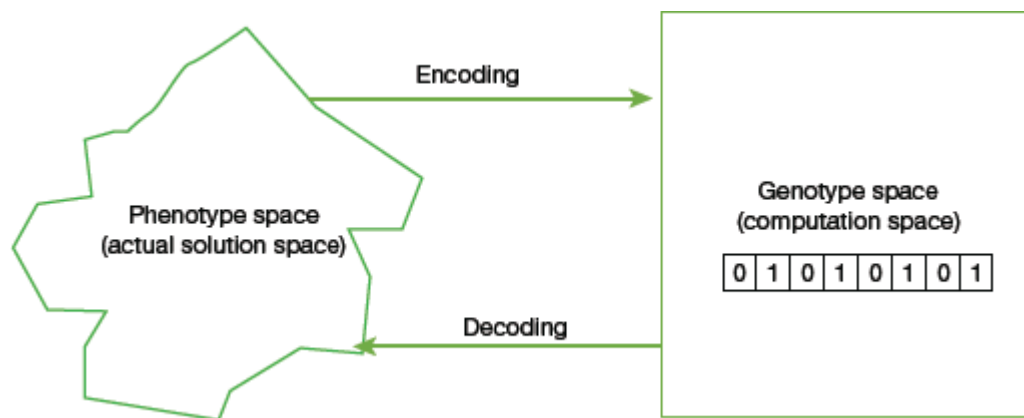


FIGURE 10.10 Genotype and Phenotype Space

8. **Fitness Function (or Evaluation Function):** It is the function which accepts the solution as input and produces the suitability of the solution as the output. That is, the **fitness function** determines how fit a solution is. This is done by evaluating how close a given solution is to the optimum solution of the desired problem.

In some cases, the fitness function and the objective function may be the same but usually they both are different. While the objective function is the function that has to be optimized, fitness function, on the other hand, is the function which guides the optimization.

9. Genetic Operators: Operators which change the genetic composition of the offspring are known as genetic operators. These include crossover, mutation, selection, etc.

10.2.4 Basic Structure

The basic structure of a genetic algorithm can be given as follows.

Step 1: Start with an initial population. The population may either be generated randomly or seeded by using heuristics.

Step 2: Repeat steps 3 and 4 until the specified condition is satisfied.

Step 3: Select parents from this population for producing new offspring. This is done by applying genetic operators like crossover and mutation.

Step 4: Off-springs generated in Step 3 replace the existing individuals in the population.

In this way genetic algorithms mimic the human evolution to some extent.

10.2.5 Genotype Representation

To implement a genetic algorithm (refer **Fig. 10.11**), we need to first decide how the solutions will be represented. A genetic algorithm can give poor performance if the representation is not correct or inefficient.

Therefore, choosing a proper representation that accurately defines a mapping between the phenotype and genotype spaces is essential for the success of a genetic algorithm.

In this section, we will discuss commonly used representations for genetic algorithms. These representations are problem specific and a mix of these representations is usually used for better results.

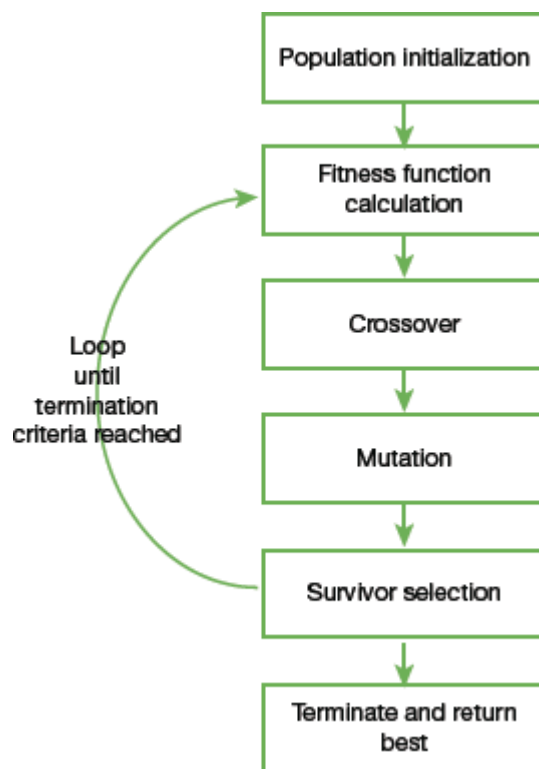


FIGURE 10.11 Steps in a genetic algorithm

Binary Representation

It is one of the simplest and most widely used representation in genetic algorithms. In this representation, the genotype consists of bit strings. Binary representation is best-suited for problems in which the solution space consists of Boolean decision variables—yes or no. For example, in the 0/1 Knapsack problem containing n items, a solution is represented by a binary string of n elements. The value at position x is 1 if it was picked and 0, otherwise.

0	0	1	0	1	1	1	0	0	1

For problems that deal with numbers, we can represent the numbers with their binary representation. However, in this encoding, different bits have different significance. Therefore, mutation and crossover operators may have undesired consequences. This can be resolved to some extent by using gray coding.

0.5	0.2	0.6	0.8	0.7	0.4	0.3	0.2	0.1	0.9

Real Valued Representation

Real-valued representation is used for representing problems in which genes must have continuous rather than discrete values.

1	2	3	4	3	2	4	1	2	1

Integer Representation

In some problems, the value of genes cannot be restricted to ‘yes’ or ‘no’, ‘0’ or ‘1’. For discrete valued genes, that can have multiple integer values, integer representation is a preferred choice. For example, four directions—north, south, east and west can be encoded as {0, 1, 2, 3}.

1	5	9	8	7	4	2	3	6	0

Permutation Representation

Permutation representation is best-suited for problems in which the solution is represented by an order of elements. For example, in the travelling salesman problem (TSP), the salesman has to take a tour of all the cities, visit each city exactly once and come back to the city from where he started. This problem is an optimization problem as the total distance to be travelled has to be minimized. The solution to this problem is an ordering or permutation of all the cities. Hence, permutation representation is used for solving such problems.

10.2.6 Population

As discussed, population is a subset of solutions in the current generation. Every individual in population is called chromosome. So, population can be thought of as a set of chromosomes. In biology, we have read that the population must have diversity to support existence. Similarly, in genetic algorithms, diversity of the population should be maintained to avoid premature convergence.

The size of the population should be chosen very carefully by repeatedly using trial and error technique. While a very large population can slow

down the execution of genetic algorithm, a smaller population may not have 'good' mating pool. Therefore, an optimal population size needs to be decided by trial and error.

10.2.7 Population Initialization

In a genetic algorithm, population can be initialized in two ways.

1. **Random initialization** populates random solutions in the initial population. This technique helps to get more optimal solutions.

The population is usually defined as a two-dimensional array of – size population, size x, chromosome size.

2. **Heuristic initialization** uses a known heuristic for the problem to populate the initial population. Note that heuristic initialization should not be used for the entire population as it may in the population having similar solutions with least diversity. Therefore, we can just seed the population with a few good solutions and fill up the rest with random solutions.

10.2.8 Population Models

Genetic algorithms widely use two types of population models:

Steady state, also known as **incremental GA**, produces one or two off-springs in each iteration. These off-springs replace one or two individuals from the population.

Generational genetic algorithms generate 'n' off-springs in each iteration. Here, n refers to the size of the population. In this case, the entire population is replaced by the new one after every iteration.

10.2.9 Fitness Function

The fitness function takes a candidate solution to the problem as input and produces a results statement as output. This output describes how 'fit' (or how 'good') the solution is with respect to the given problem. The fitness function should be fast as it will be repeatedly applied to find the fitness value. Hence, a slow fitness function can adversely affect the per-

formance of a genetic algorithm. A good fitness function must possess the following characteristics.

1. It should be fast to compute.
2. It must quantitatively measure how fit a given solution is.

For some complex problems, it may not be possible to directly apply the fitness function to calculate the fitness value. In such cases, fitness approximation is done.

0	1	2	3	4	5	6	Item number
0	1	0	1	1	0	1	Chromosome
2	9	8	5	4	0	2	Profit values
7	5	3	1	5	9	8	Weight values

Knapsack capacity = 15
 Total associated profit = 18
 Last item not picked as it exceeds knapsack capacity

FIGURE 10.12 Calculating Fitness value for a solution of the 0/1 Knapsack problem.

Consider **Fig. 10.12** which demonstrates the calculation of fitness value for a solution of the **0/1 Knapsack problem**. A simple fitness function is applied which adds the profit values of the items that have been picked. Note that these items have a '1' in their chromosome. We start from the leftmost end and continue to move toward the right until the knapsack is full and cannot take more items. Here, starting from left, we will add values for Item Number 1, 3 and 4, giving a score of $9 + 5 + 4 = 18$. We cannot include Item Number 6 because the knapsack capacity will be exceeded. Currently, it is obtained by adding weights of Item Number 1, 3 and 4 which is $5 + 1 + 5 = 11$. If we add Item Number 6, then the capacity would be $11 + 8 = 19$, which is more than 15 and hence, not possible.

10.2.10 Parent Selection

As the name suggests, parent selection refers to the process of selecting individuals from the population (or parents) which will mate and recombine to generate off-springs for the next generation. Parent selection is a very important task as only 'good' parents can produce better and fitter solutions. Therefore, the convergence rate of a genetic algorithm depends on this selection.

While selecting parents, utmost care must be taken to ensure that one extremely fit solution should not take over the entire population in a few generations, as this will diminish diversity in the population and all the solutions produced will be similar to other solutions. One extremely fit solution taking up of the entire population results in a situation known as **premature convergence**. Clearly, it is an undesirable condition in a genetic algorithm.

The most popularly used technique for selecting parents is the fitness proportionate selection. In this technique, every individual can become a parent with a probability which is proportional to its fitness. This means that fitter individuals have a higher chance of mating and propagating their features to the next generation. Fitness proportionate selection puts selection pressure to fitter individuals in the population, generating better individuals over time.

Consider a circular wheel which is divided into **n pies**, where **n** is the number of individuals in the population. Each individual gets a portion of the circle that is in proportion to its fitness value. Given below are two implementations of this technique.

Roulette Wheel Selection

From **Fig. 10.13**, we see that chromosome A gets a bigger portion of the circle, since it has the highest fitness score. Similarly, other chromosomes have a portion proportional to their fitness score.

In a roulette wheel selection, we have a fixed point on the wheel circumference. When the wheel is rotated, the region of the wheel which comes

in front of the fixed point is chosen as the parent. To find the second parent, the same process is repeated.

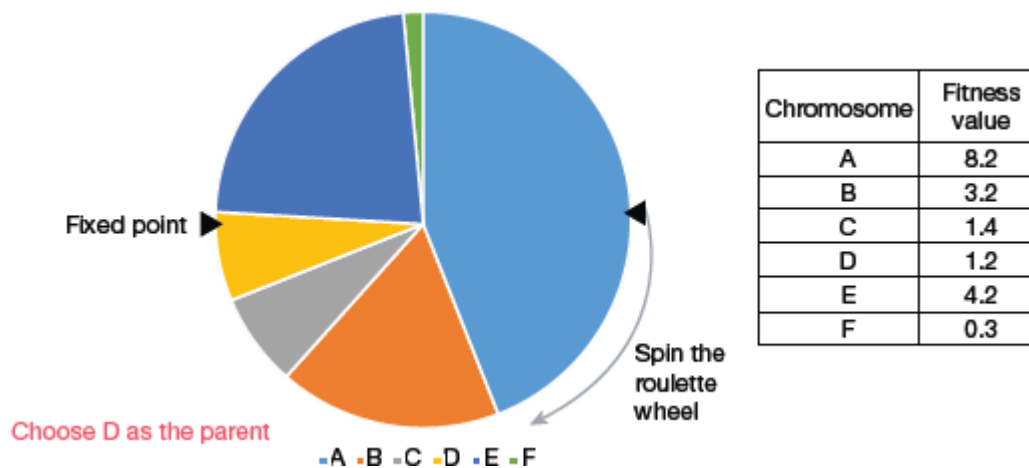


FIGURE 10.13 Roulette Wheel Selection

Since fitter individual has a larger portion of pie on the wheel, it also has a higher chance of coming in front of the fixed point when the wheel is rotated. This clearly indicates that the probability of choosing an individual is directly proportional to its fitness value.

Stochastic Universal Sampling (SUS)

SUS is similar to Roulette wheel selection, but instead of having just one fixed point, the technique has more than one fixed point as shown in **Fig. 10.14**. Therefore, the parents are selected in a single spin of the wheel. Another advantage of SUS is that it increases the probability of selecting the highly fit individuals at least once (refer **Fig. 10.15**).

Fitness proportionate selection methods do not work for cases where the fitness value is negative.

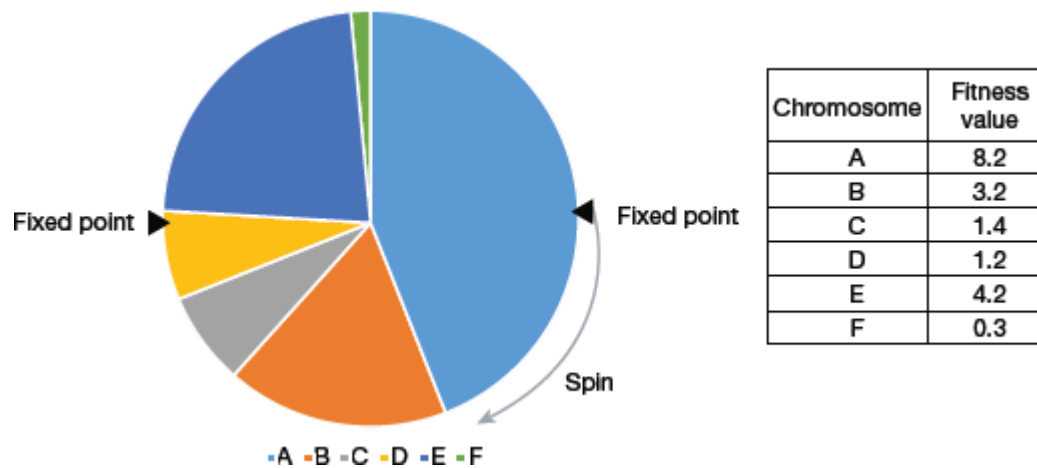


FIGURE 10.14 Stochastic Universal Sampling

Chromosome	Fitness Value	Rank
A	8.1	1
B	8.0	4
C	8.05	2
D	7.95	6
E	8.02	3
F	7.99	5

FIGURE 10.15

Tournament Selection

In K-Way tournament selection, K individuals are randomly selected from the population and then the best out of K individuals is selected to become a parent (as illustrated in **Fig. 10.16**). The same process is repeated for selecting the other parent. The main reason for wide acceptability of Tournament Selection is that it can even work with negative fitness values.

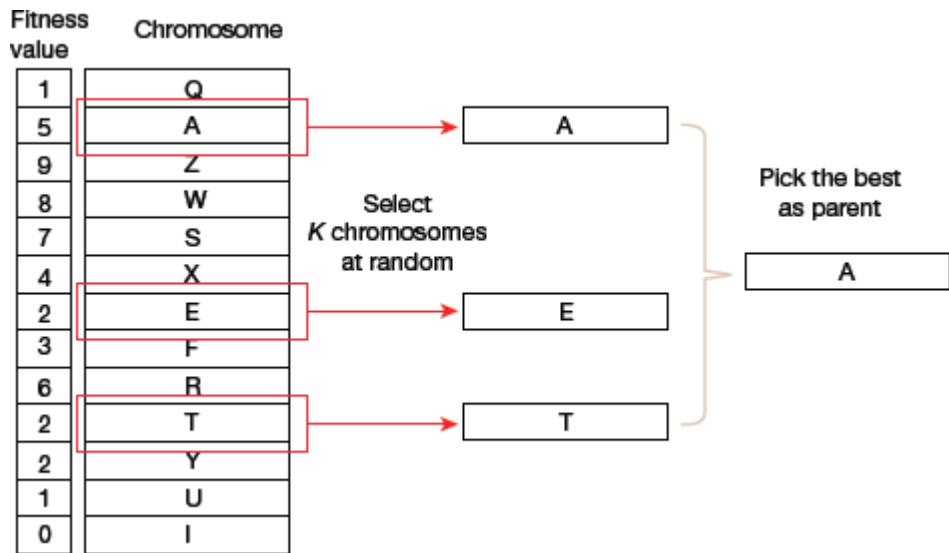


FIGURE 10.16 Tournament Selection

Rank Selection

Rank selection is used when individuals in the population have very close fitness values. In such a situation, if fitness proportionate selection is used, then each individual shares an equal portion of the pie as shown in Fig. 10.17. This indicates that irrespective of one’s fitness value, every individual has nearly the same probability of being chosen to become a parent. Clearly, in this process, there is a reduced focus on selecting fitter individuals, thereby making the genetic algorithm make poor parent selections and thus gives disappointing results.

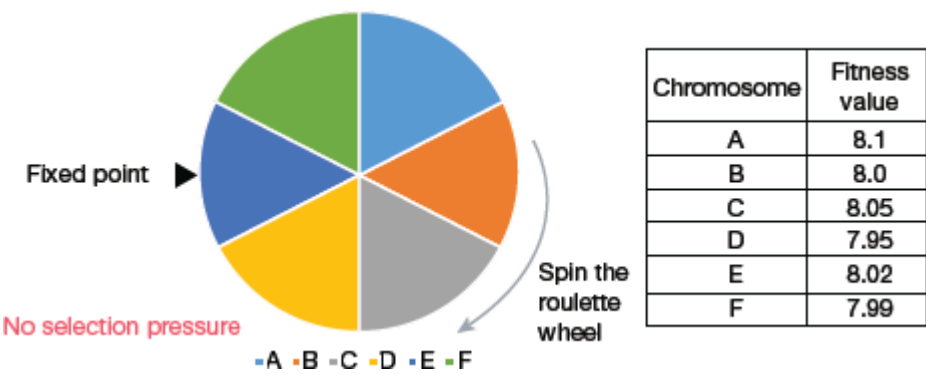


FIGURE 10.17 Rank Selection

In such a situation, a better technique is rank the individuals in the population based on their fitness values. Therefore, the selection of the parents depends on the rank of each individual and not its fitness score. Higher

the rank, higher is the probability of being chosen as a parent. Rank selection can work with negative fitness values as well.

Random Selection

In this strategy, parents are selected randomly from the existing population. There is no pressure to select fitter individuals. Because of this randomness, this technique is rarely used.

10.2.11 Crossover

The crossover operator is analogous to reproduction and biological crossover. In this technique, more than one parent is selected and one or more off-springs are produced using the genetic material of the parents. Given below are some crossover operators which are generic in nature.

However, designer of the algorithm may even implement a problem-specific operator.

1. **One Point Crossover:** In this technique (illustrated in **Fig. 10.18**), a random crossover point is selected. Then, new off-springs are generated by swapping the tails of its two parents.

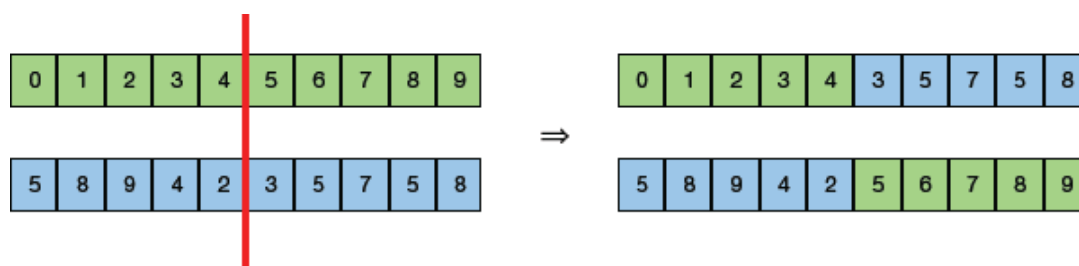


FIGURE 10.18 One Point Crossover

2. **Multi-point Crossover:** It is a generalization of the one-point crossover as it produces new off-springs by swapping alternating segments. Note that in **Fig. 10.19**, we have two (not one) crossover points.

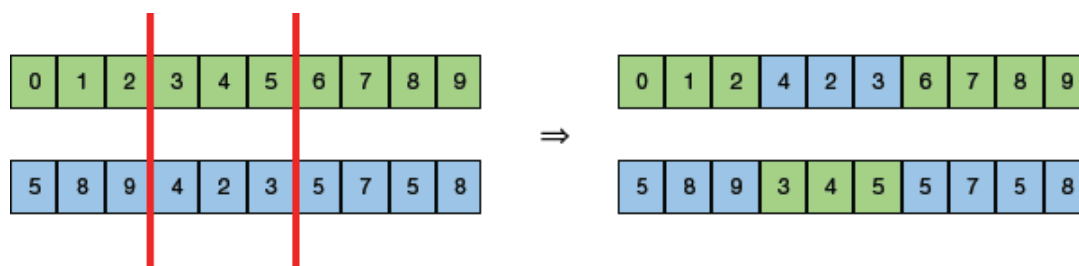


FIGURE 10.19 Multi-point Crossover

3. **Uniform Crossover:** Instead of dividing the chromosome into segments, uniform crossover treats each gene separately (as shown in **Fig. 10.20**). Each gene may or may not be included in the offspring. To determine whether to include that gene or not, a coin is flipped for each chromosome. We can also add a bias in this technique to bias the coin to one parent. This will help the offspring to get more genetic material that particular parent.



FIGURE 10.20 Uniform Crossover

4. **Whole Arithmetic Recombination:** For integer representations, we take the weighted average of the two parents by using the formula,

$$1. \text{Child1} = \alpha \cdot x + (1 - \alpha) \cdot y$$

$$2. \text{Child2} = \alpha \cdot x + (1 - \alpha) \cdot y$$

So, diverse offspring can be obtained by varying their value of α . If $\alpha = 0.5$, then both the children will be identical as shown in **Fig. 10.21**.



FIGURE 10.21 Whole Arithmetic Recombination

5. **Davis' Order Crossover (OX1):** OX1 is used for permutation-based crossovers. Follow the steps given below to apply this operator.

Step 1: Create two random crossover points in Parent 1.

Step 2: Copy the segment between the two crossover points from the first parent.

Step 3: Starting from the second crossover point in the second parent, copy the remaining unused numbers from the second parent to the child.

Repeat the steps while wrapping around the list.

Step 4: If you desire a second child from the two parents, flip Parent 1 and Parent 2 and go back to Step 1.

Example: Consider the solution given in **Fig. 10.22**. We have two parents.

Parent 1									
8	4	7	3	6	2	5	1	9	0

Parent 2									
0	1	2	3	4	5	6	7	8	9

Child 1									
0	4	7	3	6	2	5	1	8	9

FIGURE 10.22 Davis' Order Crossover

The child is produced by copying elements 3, 6, 2, 5, 1 from the first parent (according to Step 2).

Then, we copy elements from parent 2 that have not been used so far. Starting on the right side of the second crossover point, grab alleles from parent 2 and insert them in Child 1 at the right edge of the second crossover point. Since 8 is in that position in Parent 2, it is inserted into Child 1 first. Similarly, element 9 and 0 from Parent 2 are copied in Child 1 at the right of second crossover point (while wrapping around the list).

Since elements 1 and 2 from parent 2 are already in child, we will not copy them again.

Copy elements 4 and 7 from Parent 1 to complete the chromosome.

10.2.12 Mutation

Mutation is the process in which a new solution is obtained by introducing a small random tweak in the chromosome. Mutation plays an important role in genetic algorithms as it maintains as well as introduces diversity in the genetic population. Mutation performs 'exploration' of the search space and is very essential for a genetic algorithm to converge successfully.

In this section, we will discuss some widely used mutation operators. A genetic algorithm designer can use one of operators or a combination of two or more operators or use a problem-specific mutation operator.

1. **Bit Flip Mutation:** In this technique (illustrated in **Fig. 10.23**), one or more bits are selected randomly and flipped. However, it is applied on binary encoded genetic algorithms.

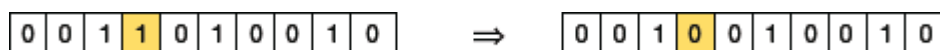


FIGURE 10.23 Bit Flip Mutation

2. **Random Resetting:** It is an extension of the bit flip technique. Random setting operator is used for genetic algorithms that use integer representation. It selects a random value from the set of permissible values and assigns it to a gene that is again chosen randomly.
3. **Swap Mutation:** According to this approach (demonstrated in **Fig. 10.24**), two positions on the chromosome are selected randomly and their values are swapped (interchanged). This operator is commonly used in permutation-based encodings.

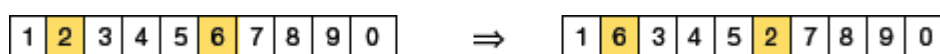


FIGURE 10.24 Swap Mutation

4. **Scramble Mutation:** It is a popular technique used with genetic algorithms that use permutation representations. As the name suggests, scramble mutation selects a subset of genes from the chromosome and then randomly scrambles (or shuffles) their values as shown in **Fig. 10.25**.

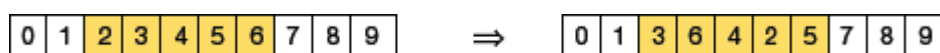


FIGURE 10.25 Scramble Mutation

5. **Inversion Mutation:** As in scramble mutation (refer **Fig. 10.26**), inversion mutation also chooses a subset of genes but instead of shuffling the subset, the string in the entire subset is inverted.

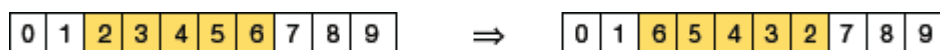


FIGURE 10.26 Inversion Mutation

Survivor Selection

The survivor selection policy is used to determine which individuals will be a part of the next generation and which will be discarded. This is an important step to ensure that the fitter individuals are passed on to the next generation. We also need to ensure that diversity in the population should be maintained in every generation.

Some genetic algorithms use the strategy of **elitism**, which ensures that the fittest member of the current population is always propagated to the next generation. Therefore, under no circumstance can the fittest member of the current population be replaced. Though it is the simplest way to kick random members out of the population, this approach frequently suffers from convergence issues. Therefore, policies discussed below are widely used.

Age-Based Selection

Age-based selection does not consider fitness value for selection. It allows each individual in the population to reproduce for a finite generation. After that, irrespective of how good it is, it is kicked out of the population.

For example, in **Fig. 10.27**, we see that age is a number that specifies for how many generations the individual has been in the population. The oldest members of the population, P4 and P7 are kicked out of the population and the ages of the other members are incremented by one.

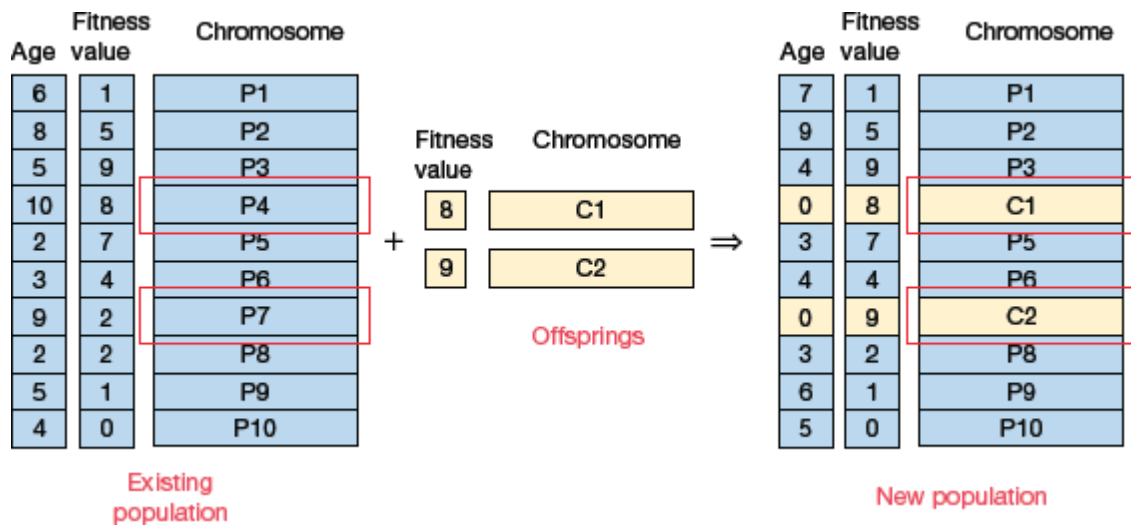


FIGURE 10.27 Age-Based Selection

Fitness-Based Selection

In this selection strategy, the children replace the least fit individuals in the population. The least fit individuals can be selected using any selection policies (like tournament selection, fitness proportionate selection, etc.).

For example, in **Fig. 10.28**, we see that the least fit individuals P1 and P10 of the population are replaced by the children. Note that P1 and P9 also have the same fitness value, so even they could have been removed. The decision of which individual to remove from the population is arbitrary.

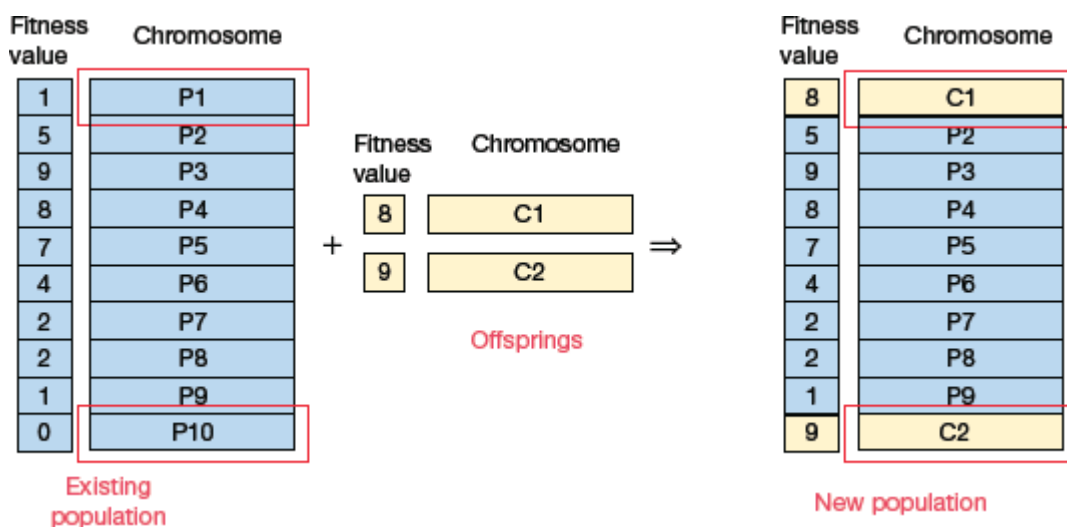


FIGURE 10.28 Fitness-Based Selection

10.2.13 Termination Condition

The termination condition of a genetic algorithm determines when the execution of the algorithm will halt. It has been observed that a genetic algorithm initially works very fast and gives good better solutions in subsequent iterations. But once the saturation point is reached, the improvements in solutions become almost negligible. At such a point, it is better to terminate the algorithm. The termination condition must give a solution that is close to the optimal. The algorithm can terminate under the following conditions:

1. When there is no improvement for X iterations in the population.
2. When the absolute number of generations has been reached.
3. When a certain pre-defined value of the objective function value has been reached.

To implement termination condition in a genetic algorithm, we can maintain a counter (with initial value = 0), to track the generations for which there has been no improvement in the population. When the generated off-spring is not better than the individuals in the population, the counter is incremented.

If the fitness value of any of the off-springs is better than other individuals in the population, then the counter is again reset to zero. The algorithm terminates when the value of the counter becomes equal to a pre-determined value. Like other parameters, the termination condition of a genetic algorithm is specific to the problem at hand. And the designer of the genetic algorithm should try various options to see what best-suits that particular problem.

10.2.14 Application Areas

Genetic algorithms are widely used in a variety of application areas as given below.

1. **Optimization:** Genetic algorithms are widely used in optimization problems. The goal here is to either maximize or minimize the value of a given objective function as per the specified constraints. Genetic

algorithms can also be used for multimodal optimization to find multiple optimum solutions. Travelling salesman problem (TSP) can also solve optimally using genetic algorithms.

2. **Economics:** Economic models (like the cobweb model, game theory equilibrium resolution, asset pricing, etc.) are characterized using genetic algorithms.
3. **Neural Networks:** Genetic algorithms are used to train neural networks, particularly recurrent neural networks.
4. **Parallelization:** Genetic algorithms can perform parallel computations. Therefore, these algorithms can effectively solve certain complex problems.
5. **Image Processing:** Digital image processing (DIP) and dense pixel matching can be easily performed using genetic algorithms.
6. **Scheduling applications:** Scheduling problems, particularly the time tabling problem, can easily be solved by genetic algorithms.
7. **Machine Learning:** Genetic algorithms is an important concept used in machine learning algorithms.
8. **Robot Trajectory Generation:** Genetic algorithms can be used to plan the path which will be taken by a robot arm while moving from one point to another.
9. **Parametric Design of Aircraft:** Better aircraft designs can be obtained by applying genetic algorithms while varying certain important parameters.
10. **DNA Analysis:** Genetic algorithms when applied on spectrometric data about the sample, can be used to determine the structure of DNA.

10.3 Soft Computing

Soft computing refers to a group of computational techniques that are based on artificial intelligence (AI) and natural selection. They are often used to develop cost-effective solutions for complex real-world problems that are either difficult or impossible to be solved by conventional computing techniques.

Soft computing differs from conventional computing as it deals with approximation models.

The term ‘soft computing’ was first coined in 1992 by **Zadeh** to provide precise approximation and quick solutions for complex real-life problems. For example, a soft computing system can be used for character recognition as shown in **Fig. 10.29**.

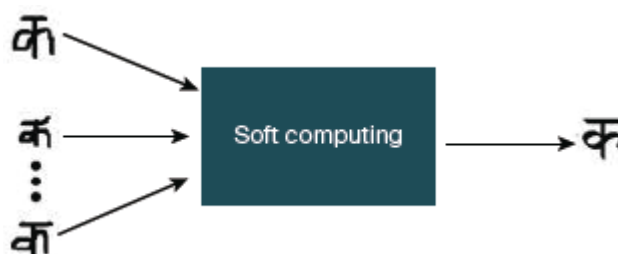


FIGURE 10.29 Soft Computing

Soft computing is an emerging domain that can map a human mind. In fact, the human mind is a role model for soft computing.

Computation is a process of converting input of one form to another form using certain control actions. The input is called an antecedent and the output is called the consequent.

10.3.1 Characteristics of Soft Computing

Soft computing provides an approximate but precise solution for real-world complex problems that cannot be solved by conventional computing techniques.

Soft computing algorithms are not affected by changes in the environment. This makes these algorithms adaptive for the current process.

Soft computing algorithms do not require any mathematical model to solve the problem as it learns **from experimental data**.

Solutions given by soft computing are uncertain since we get different solutions on solving the same problem using the same set of input data at different intervals of time. Refer **Fig. 10.30** to understand how soft computing differs from hard computing

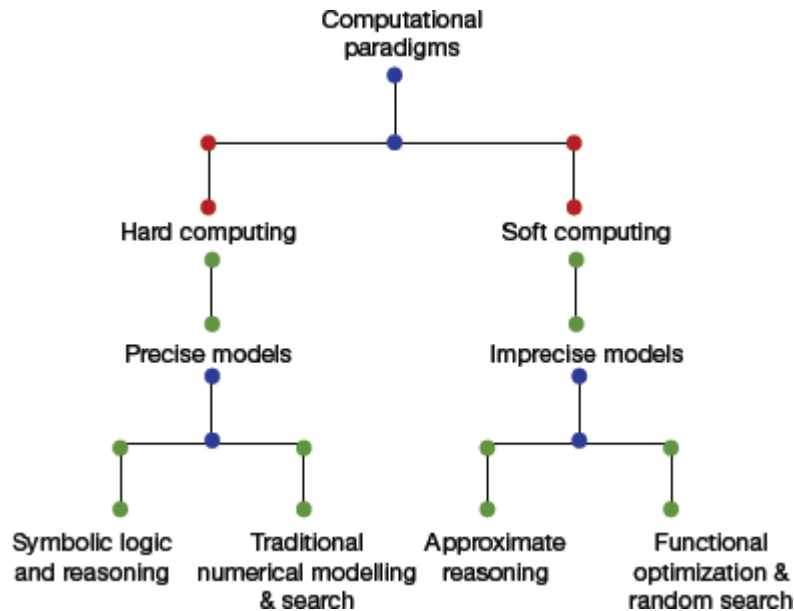


FIGURE 10.30 Computational paradigms and Soft Computing

It is based on Fuzzy logic, genetic algorithms, machine learning, ANN and expert systems. **Table 10.4** draws comparison between hard and soft computing.

- Soft-computing algorithms are robust as they can work well even in the presence of noise in the input data.
- Soft computing provides low-cost solutions for problems which could be computationally very expensive if solved using hard computing.

10.3.2 Need for Soft Computing

1. Hard computing or conventional techniques are usually good for solving mathematical problems that need a precise answer. But when the task is to solve a complex real-world problem where either approximate solution works better or there exist no precise solution, then soft computing is the only possibility.
2. For applications that need to map human mind, soft computing is the only viable technique. For better understanding of the concept, refer **Table 10.4** and **Table 10.5** that makes a comparison between Soft Computing and Hard Computing and between Soft Computing and AI respectively.

TABLE 10.4 Soft computing vs hard computing

TABLE 10.5 Differences between AI and Soft Computing

Example: Let us consider a problem that cannot be solved by traditional computing, but soft computing gives an approximate solution. If you ask, ‘is the tea is hot or cold?’ Traditional computing techniques may give an answer to this question but when asked, ‘how much hot?’, they will not be able to tell. Soft computing rescues this situation by saying, ‘moderately hot’.

Advantages

1. Applies simple mathematical calculations
2. Gives efficient solutions
3. Can be applied to solve problems in real-time
4. Based on human reasoning

Disadvantages

1. Instead of precise, it gives approximate results.
2. Even a small error may halt the entire system. In such a case, the system must be corrected from the beginning. And this may take more time to process input data to give the output.

10.3.3 Applications of Soft Computing

1. Soft computing can be used to control motors automatically like the induction motor.
2. Intelligent control systems to control power plants can be designed using soft computing.
3. Soft computing takes an image, music or video as input to get an exact duplicate of the original image, music or video.
4. In biomedical field, soft computing techniques (refer **Fig. 10.31**) are used for diagnosis, monitoring, treatment and therapy.
5. To perform a task, smart devices automatically communicate with other devices using different communication protocols. There is no proper standard protocol to communicate and this often creates problem. Soft computing techniques are used in such a situation to facilitate communication between smart devices with high privacy and robustness.

In fact, there are many more applications where soft computing is widely used.

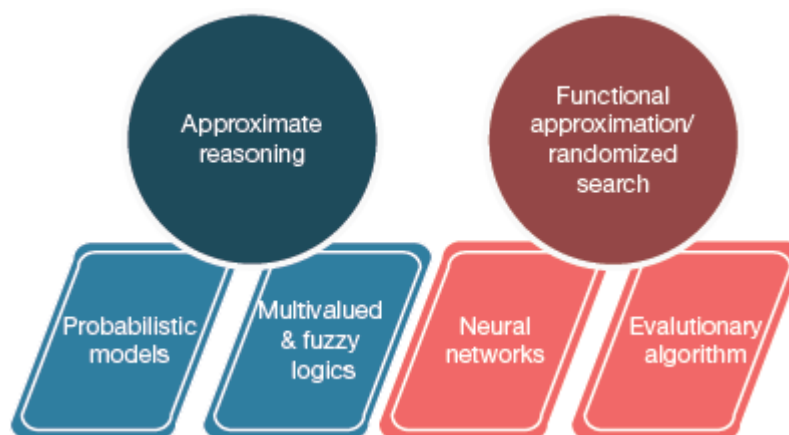


FIGURE 10.31 Soft Computing Techniques

10.3.4 Elements of Soft Computing

Soft computing lays the foundation for an emerging field of conceptual intelligence. Technologies like fuzzy logic (FL), machine learning (ML), neural network (NN), probabilistic reasoning (PR) and evolutionary computation (EC) are the supplements of soft computing and are used to resolve any complex problem.

Fuzzy Logic

As discussed, fuzzy logic applies mathematical concepts to solve problems with an open and imprecise spectrum of data to get an array of precise conclusions.

Fuzzy logic achieves the best possible solution to complex problems using the available information on the input data. Fuzzy logic is considered the best solution finder.

FIGURE 10.32 Fuzzy Logic for movement of a Robot

Example: Fuzzy logic is used to control the movement of a robot (refer **Fig. 10.32**) from one place to another within a short time where there are

many obstacles on the way. To calculate its movement to reach the destination without colliding to any obstacle, fuzzy solutions may give better results than precise solutions.

Probabilistic Reasoning

The technique applies the concept of probability to determine the uncertainty of a given event. The probability of an event is calculated using the below formula:

Probabilistic reasoning is mainly used in the following scenarios:

1. When conducting an experiment, we get an unknown error.
2. When we are not sure about uncertain outcomes.
3. When number of predicates are too large to handle.

Neural Network (ANN)

Neural networks, developed in the 1950s, facilitate users to solve real-world problems, which a traditional program can never solve. They work in the same way as the human brain, by emulating a network of neurons that makes a human brain. ANN enables a machine to mimic the human brain so that it can think and take an effective decision. The output obtained is mathematically justifiable on the current input data based on past information stored.

A neural network is made of several interconnected layers. Each layer has one or more nodes for processing inputs received. A layer accepts the output of the previous layer as its input, processes it to give the output to the layer succeeding it. This is known as forward propagation.

After giving the final output, backward propagation starts. In this phase, the result obtained from neural network is compared with the actual output. If an error is identified (due to forward propagation), then the task is to propagate backwards in the network to minimize the value/weight of that neuron(s) that are responsible for the error.

Soft computing usually uses deep neural networks. A deep neural network has a greater number of hidden layers as compared to a simple ANN. While a traditional neural network has two to three hidden layers, a deep neural network, on the other hand, has as many as 150 layers.

Genetic Algorithms (GA)

Genetic algorithm finds solutions by deriving concepts from natural selection and survival of the fittest which are very important concepts in genetics.

Genetic algorithms that come under the category of evolutionary algorithms were introduced by Prof. John Holland in 1965. They mainly solve optimization problems like maximization and minimization of objective functions.

Genetic algorithms are based on heuristic or randomized search that provides an initial set of solutions and then generates a solution to the NP-hard problems efficiently and effectively in real time. This means that complex problems which cannot be solved mathematically can be easily solved by using genetic algorithm.

Example: A person wants to invest some money in the bank. There are numerous banks where he can invest in different schemes and policies (refer **Fig. 10.33**). The decision regarding bank and policy for investment must be taken to maximize profits. In such a scenario, application of evolution computing using genetic algorithms gives the best results.

FIGURE 10.33 Genetic Algorithm for Selecting Bank and its Policies

When applying genetic algorithm, remember that the initial population contains several solutions to a concerning problem. Let us quickly enu-

merate the steps performed to solve a problem using genetic algorithm.

Step 1: The algorithm calculates the fitness of each individual solution and provides them with the fitness score. Higher the fitness score, higher is the probability of a solution being selected for the reproduction process.

Step 2: The algorithm selects the solution with the highest fitness score.

Step 3: Perform crossover on the selected individual from the population to get an offspring.

Step 4: Add this offspring to the initial population.

Step 5: To maintain population diversity, mutation phase inserts random genes into the offspring.

The algorithm will stop producing the offsprings when they are different than the previous generation.

10.4 Transfer Learning

Transfer learning is the reuse of a pre-trained model on a new problem (as shown in **Fig. 10.34**). So, a machine exploits the knowledge gained from a previous task to improve generalization about another. For example, a classifier that is trained to predict whether an image contains food or not can also be used to recognize drinks with the help of knowledge it gained during the training process.

FIGURE 10.34 Transfer Learning

It is currently being used in deep learning in which deep neural networks are trained with very limited data. Since most of the real-world applications do not have millions of labelled data points to train complex models, transfer learning is an important concept.

In transfer learning, we transfer the weights that a network has learned at 'task A' to a new 'task B.' While task A has a lot of labelled training data available, task B on the other hand, lacks volumes of data. In such a scenario, transfer learning starts the learning process with patterns learned from solving a related task rather than starting from the scratch.

Though transfer learning is not a machine learning technique, it is popularly used in combination with neural networks that require huge amounts of data and computational power. Applications like computer vision and natural language processing extensively use this technique.

10.4.1 How Transfer Learning Works?

In a computer vision application, neural networks detect edges in the earlier layers, shapes in the middle layer and some task-specific features in the later layers. Similarly, in transfer learning, we use the early and middle layers of the model and retrain the latter layers. It helps to take advantage of labelled data of the task it was initially trained on and then transfer as much knowledge as possible from the previous task.

10.4.2 When to Use Transfer Learning

Some guidelines which dictate when transfer learning might be used are as follows and summarized in **Fig. 10.35**:

FIGURE 10.35 When to Use Transfer Learning

1. When there is very limited labelled training data available to train the neural network from scratch.
2. When there is a neural network that has already been trained on a similar task with massive amounts of data.
3. When the two tasks have the same input.
4. If the existing model was trained using an open-source library (like TensorFlow), you can simply restore some layers and retrain others for the task at hand.

Transfer learning is an optimization, a shortcut to save time and get better performance. Lisa Torrey and Jude Shavlik described three possible benefits to look for when using transfer learning:

1. **Higher start:** The initial skill on the source model before even the model was refined, is higher than it otherwise would be (refer **Fig. 10.36**).
2. **Higher slope:** The rate of improvement of skill during training of the source model is steeper than it otherwise would be.
3. **Higher asymptote:** The converged skill of the trained model is better than it otherwise would be.

FIGURE 10.36 Graph depicting merits of Transfer Learning

However, the choice of source data or source model is an open problem and may require domain expertise and/or Three ways in which transfer might improve learning intuition developed via experience.

Note that transfer learning only works in the following situations:

1. If the features learned from the first task are general so that they can be useful for another related task.
2. Input of the model needs to have the same size as it was initially trained with. In case this condition is not satisfied, then resize the input to the needed size.

You can visit pretrained.ml which offers a list of sortable and searchable compilation of pretrained deep learning models with demos and code.

10.4.3 Approaches to Transfer Learning

Training a Model to Reuse It

To solve a task A having limited data to train a deep neural network, find a related task B with an abundance of data. Train the deep neural network on task B and use the same model to solve task A. Whether to use the entire model or just few layers of the model will depend on the problem at hand.

If the input for both the tasks is same, then re-use the model, else make some changes in the task-specific layers of the model before using it.

Using a Pre-Trained Model

Use a pre-trained model but do some research to choose the best one from all available options. Again, whether to reuse the entire model or re-train few task-specific layers depends on the problem to solve. For example, Keras provides numerous pre-trained models that can be used for transfer learning, prediction, feature extraction and fine-tuning. This type of transfer learning is most commonly used throughout deep learning.

Some popular pre-trained machine learning models include the Inception-v3 model, which was trained for the ImageNet to classify images into 1,000 classes like ‘zebra,’ ‘Dalmatian’ and ‘dishwasher.’

Microsoft also offers some pre-trained models that can be used with both R and Python. Developers just have to incorporate the MicrosoftML R and the Microsoftml Python package. Other commonly used models are ResNet and AlexNet.

The more you want to inherit features from a pre-trained model, the more you have to freeze layers.

Feature Extraction

Use deep learning techniques to discover the best representation of the problem. In other words, identify the most important features. This approach is therefore, known as representation learning, and often gives much better performance than hand-designed representation.

Pre-trained model of ImageNet cannot be used with biomedical images because ImageNet does not contain images belonging to the biomedical field.

The learned representation can then be used to solve other problems. But to solve other problems, only first few layers are reused as they are the ones that identify the most useful features of the problem. Latter layers are retrained as they are task-specific. Feature extraction is extensively used in computer vision as it reduces the size of the dataset (refer **Fig. 10.37**). This, in turn, decreases computation time and enhances the performance of the algorithm.

FIGURE 10.37

10.4.4 Traditional Machine Learning vs Transfer Learning

Deep learning experts introduced transfer learning to overcome the limitations of traditional machine learning models. Refer **Table 10.6** and Figure 10.38 to see a comparison between Traditional Machine Learning and Transfer Learning.

TABLE 10.6 Differences between traditional machine learning and transfer learning

FIGURE 10.38 Traditional Machine Learning vs Transfer Learning

10.4.5 Classical Transfer Learning Strategies

Based on the domain of the application, the problem to solve and the availability of data, we may choose different transfer learning strategies. However, before deciding on the strategy, it is important to have answers for the following questions to improve performance of the target task:

1. *Which* part of the knowledge can be transferred from the source to the targets?
2. *When* to transfer knowledge and when not to?
3. *How* to transfer the knowledge gained from the source model?

Traditionally, there are three types of transfer learning strategies that can be chosen based upon the domain of the task and the amount of labelled/unlabelled data available.

Inductive Transfer Learning

This requires the source and target domains to be the same even if the specific tasks on which the model is working on are different (refer **Fig. 10.39**).

The algorithms try their best to use knowledge from the pre-trained source model and apply them to improve the target task. Since the source model has the expertise on the features of the domain, it is better to start with it than to start from scratch.

Inductive transfer learning is further divided into two groups—multi-task learning and self-taught learning. While task learning is applicable when the source domain contains labelled data, the latter is used when labelled data is not available.

FIGURE 10.39 Inductive Transfer Learning

Transductive Transfer Learning

This type of learning is used when the domains of the source and target tasks are not exactly the same but are interrelated. In this scenario, the source domain has a lot of labelled data but the target domain has only unlabelled data. So, this learning can be used to derive similarities between the source and target tasks.

Unsupervised Transfer Learning

Though this is similar to inductive transfer learning, the algorithms in this approach focus on unsupervised tasks and work on unlabelled datasets both in the source and target tasks.

Transfer learning strategies can also be categorized based on similarity of the domain. This type of division is independent of the type of data samples present for training.

Homogeneous Transfer Learning

As the name suggests, this approach is used when the domains are of the same feature space. There exists only a slight difference in marginal distributions in the domains which can be catered by correcting the sample selection bias or covariate shift.

Instance Transfer

It is done when there is a large amount of labelled data in the source domain, a limited number in the target domain and marginal distributions in both the domains and feature spaces.

For example, imagine that you have to build a model to diagnose cancer in a specific region where there are mostly elderly people. But you have limited target-domain instances. The relevant data are available from another region where the majority comprises young people. In this scenario, directly transferring all the data from another region may not be a viable solution due to the existence of marginal distribution and the fact that elderly have a higher risk of cancer than younger people.

So, here, the marginal distributions are adapted by reassigning weights to the source domain instances in the loss function.

Parameter Transfer

In this technique, it is believed that a well-trained model on the source domain has learned a well-defined structure, and if two tasks are related, this structure can be transferred to the target model. Parameter transfer learning technique transfers the knowledge at the model/parameter level. Knowledge is transferred through shared parameters of the source and target domain learner models. Generally, there are two ways to share the weights-soft weight sharing and hard weight sharing. In ***soft weight sharing***, the model is expected to be close to the already learned features and is penalized if its weights deviate significantly from a given set of weights. In ***hard weight sharing***, the exact weights among different models are shared.

Feature-Representation Transfer

As implicit from the name, this approach transforms the original features to create a new feature representation. This technique can further be categorized as asymmetric and symmetric feature-based transfer learning.

1. *Asymmetric approaches* transform the source features to match the target ones. Therefore, features from the source domain are transformed to make them fit into the target feature space. However, there can be some information loss in this process due to the marginal difference in the feature distribution.
2. *Symmetric approaches* identify a common feature space and then transform both the source and the target features into this new feature representation.

Relational-Knowledge Transfer

This technique focuses on learning the relations between the source and a target domain to extract past knowledge so that it can be used in the

current context. Logical relationship or rules learned in the source domain are then transferred to the target domain.

For example, if we learn the relationship between different elements of the speech in a male voice, it can be then used to analyse the sentence in another voice.

Heterogeneous Transfer Learning

In transfer learning, representations from a previous network are derived to extract meaningful features from new samples for an inter-related task. Though this technique seems simple and straightforward, its applicability is not a trivial task due to the following reasons.

1. There may exist a difference between the source and the target domain and feature spaces.
2. Limited availability of labelled data in the source domain with the same feature space as the target domain.

In such a scenario, heterogeneous transfer learning methods are preferred. This learning strategy is, therefore, applied in cross-domain tasks like cross-language text categorization, text-to-image classification and many others.

10.4.6 Transfer Learning for Deep Learning

Transfer learning is popularly used for applications like NLP and image recognition. There are several pre-trained models that achieve state-of-the-art performance in this field.

Off-The-Shelf Pre-Trained Models as Feature Extractors

We know that deep learning systems comprises multiple layers and each layer learns different features. Initial layers extract higher-level features that narrow down to fine-grained features as we go deeper into the network. These layers are finally connected to the last layer that gives the final output. This opens the scope of using popular pre-trained networks (such as Oxford VGG Model, Google Inception Model, Microsoft ResNet Model) without its final layer to extract a fixed feature for other tasks.

Transfer Learning with Pre-Trained Deep Learning Models as Feature Extractors

In this technique, the pre-trained model's weighted layers are just used to extract features. The model's weights are not updated during training with new data for the new task. Since the pre-trained models are trained on a large and general enough dataset, it effectively serves as a generic model of the visual world.

FIGURE 10.40 Transfer learning with Pre-trained deep learning models as feature extractors

Fine Tuning Off-The-Shelf Pre-Trained Models

In this technique, we do not just directly depend on the features extracted from the pre-trained models and replace the final layer. Rather, we selectively retrain some of the previous layers.

Deep neural networks have a layered architecture and have many tunable hyperparameters. While the initial layers capture generic features, the later ones focus on the explicit task at hand. Thus, the higher-order feature representations in the base model must be fine-tuned to make them more relevant for the specific task. We can re-train some layers of the model and keep others frozen in training.

An example is depicted in **Fig. 10.41** on an object detection task, where initial lower layers of the network learn very generic features and the higher layers learn very task-specific features.

FIGURE 10.41 Freeze vs Fine-tune in Transfer Learning for Deep Learning

Freezing and Fine-Tuning Layers

To increase the model's performance, we often re-train (or 'fine-tune') the weights of the top layers of the pre-trained model and train the newly added classifier. Fine tuning a small number of top layers that are generic rather than the entire model, therefore, it updates the weights for a specific learning and allows the model to apply past knowledge in the target domain and re-learn some things again.

Freezing initial layers to reuse the basic knowledge derived from the past training and fine-tuning higher layers to adapt specialized features to work with the new dataset offers an efficient solution.

10.4.7 Steps in Transfer Learning

Transfer learning can be performed in just a few steps that are illustrated in **Fig. 10.42**.

FIGURE 10.42 Transfer Learning process

Step 1: Obtain pre-trained model. The model is kept as the base of our training. Its selection entirely depends on the task at hand. Transfer learning that the knowledge of the pre-trained source model must be directly related and compatible with the target task domain. Few examples of pre-trained models are,

For computer vision: VGG-16, VGG-19, Inception V3, Xception, ResNet-50

For NLP tasks: Word2Vec, GloVe, FastText

Step 2. Create a base model. The base model is one of the architectures (such as ResNet or Xception) which is in close relation to our task. We can either download the network weights to save time or use the network architecture to train our model from scratch. If the base model has more neurons in the final output layer than required, then we need to remove the final output layer and change it accordingly.

Step 3: Freeze layers. Freeze the starting layers from the pre-trained model to save time in making the model learn the basic features. If we do not freeze the initial layers, the learning already been done will be lost. The process will then be equivalent to training the model from scratch, thereby losing time and other resources.

Step 4: Add new trainable layers. From the pre-trained model, we only use knowledge of feature extraction layers.

Rest of the layers that would perform the specialized tasks needs to be added in the model. These are generally the final output layers.

FIGURE 10.43 Fine Tuning the model

Step 5: Train the new layers. The pre-trained model's final output may be different from that of our new model. For example, if a pre-trained model was trained on the ImageNet dataset to generate 1000 classes and the new task at hand expects the output to be having just two classes, then we have to train the model with a new output layer in place.

Step 6: Fine-tune your model. In this step, we unfreeze some layers of the base model and train the entire model again on the whole dataset at a very low learning rate. This improves performance of the model on the new dataset while preventing overfitting issues (refer **Fig. 10.43**).

10.4.8 Types of Deep Transfer Learning

Domain Adaptation

It is done when the source and target domains have different feature spaces and distributions. So, domain adaptation technique adapts (or alter) one or more source domains to transfer information to improve the performance of a target learner. The adaptation of source domain is pur-

posely done bring the distribution of the source closer to that of the target.

Domain Confusion

In a neural network, different layers identify different features. For best results, the source and target domains should be as similar as possible. The domain confusion technique creates confusion at the source domain to confuse the high-level classification layers of a neural network. The confusion is created by matching the distributions of the target and source domains to ensure that samples are indistinguishable to the classifier.

While using domain confusion, we need to minimize the classification loss for the source samples, and the domain confusion loss for all samples.

Multi-Task Learning

In this approach (refer **Fig. 10.44**), several tasks from the same domain are learned simultaneously without distinction between the source and targets. This helps to transfer knowledge from each scenario of the same domain. The shared knowledge helps the learner to optimize the learning (or performance) across all of the n tasks.

Multi-task Learning in Deep Transfer Learning

One-Shot Learning

In this learning, the classification task is given one or a few examples to learn from. After learning, the classifier then classifies many new examples in the future. For example, a face recognition algorithm learns a person's face that must be classified correctly with different facial expressions, lighting conditions, accessories and hairstyles. This model is fed with one or a few template photos as input.

FIGURE 10.44 Multi-Task Learning

Zero-Shot Learning

In this learning technique, transfer learning is done with zero instances of a class. This learning does not depend on labelled data samples. But it does require additional data during the training phase to understand the unseen data.

Zero-shot learning focuses on the traditional input variable, x , the traditional output variable, y , and the task-specific random variable. It is extensively used in applications like machine translation, where we may not have labels in the target language.

10.4.9 Applications of Transfer Learning

1. **Learning from simulations:** Transfer learning is used when traditional machine learning applications that rely on hardware for interaction makes data gathering and model training an expensive and time-consuming or simply too dangerous affair.

In such a scenario, simulation is the preferred tool. Transfer learning is done by applying the knowledge gained by simulation process to the real world. Though the feature spaces between source and target domain are the same (both generally rely on pixels), the marginal probability distributions are different, as objects in the simulation and the source *look* different. Even the conditional probability distributions

between simulation and real world might be different as the simulation is not able to fully replicate all reactions in the real world.

Udacity has open-sourced the simulator it uses for teaching its self-driving car engineer nanodegree.

Learning from simulations has the following benefits:

1. Data gathering becomes easy as objects can be easily bounded and analysed.
2. Fosters training as learning can be parallelized across multiple instances.

Simulation is also important in robotics. Training models on a real robot consume a lot of time and other resources. So, a better alternative is to learn from a simulation and transfer the knowledge to a real-world robot. Figure 10.45 shows a self-driving car simulator.

Credit: Have a nice day Photo / Shutterstock

FIGURE 10.45 Self-driving car simulator

2. **Adapting to new domains:** We know that the data where labelled information is easily available and the data that we actually care about are different. Even if the training and the test data *look* the same, training data may have a bias that is unnoticeable to humans but the model may exploit to overfit on the training data.

Domain adaptation is also used to adapt to different text types. For example, if an NLP model is trained, then it will have difficulty coping with more novel text forms such as social media messages.

Recently, domain adaptation is also being used in automatic speech recognition (ASR) applications. Such an application may be trained with voice of 500 speakers, but only those people with a similar accent can be benefited. The rest may have trouble being understood.

3. Transferring knowledge across languages: Another important application of transfer learning is learning from one language and applying that knowledge to another language. Reliable cross-lingual adaptation methods allow the vast amounts of labelled data in English and apply them to any language, particularly those that are truly low-resource languages.

10.4.10 When Does Transfer Learning *Not* Work?

Transfer learning should not be used when the weights trained from the source task are different from that of the target task. For example, if the model that was trained to identify tigers cannot be used to identify a sports car. Initializing a model with pre-trained weights that correspond with similar outputs is better than using weights with no correlation with the expected output.

Removing layers from a pre-trained model may cause issues with the architecture of the model. For example, if the first layer is removed, then the model will have a low learning rate as it has to juggle working with low-level features. Moreover, the model may suffer from overfitting issues as removing layers reduces the number of parameters that can be trained. Therefore, using a model with the correct number of layers is vital to the success of a project using transfer learning.



KEY TERMS

Population: It is a subset of all the possible solutions to the given problem.

Chromosomes: A chromosome is the individual solution to the given problem.

Gene: A gene is one element position of a chromosome.

Allele: It is the value which a gene takes for a particular chromosome.

Genotype: Genotype is the population (solution) in the computation space that can be represented in a simple and easy-to-understand manner so that it can be easily manipulated using a computing system.

Phenotype: Phenotype is the population (solution) in the real-world solution space that can be represented in real world situations.

Fitness Function (or Evaluation Function): It is the function which accepts the solution as input and produces the suitability of the solution as the output.

Genetic Operators: Operators which change the genetic composition of the offspring are known as genetic operators. These include crossover, mutation, selection, etc.

Mutation: The process in which a new solution is obtained by introducing a small random tweak in the chromosome.

Parent selection: The process of selecting individuals from the population (or parents) which will mate and recombine to generate off-springs for the next generation.

Soft computing: A group of computational techniques that are based on artificial intelligence (AI) and natural selection. It is often used to develop cost-effective solutions for complex real-world problems that are either difficult or impossible to be solved by conventional computing techniques.

Transfer learning: The reuse of a pre-trained model on a new problem.



RECAPITULATION POINTS

1. A set is a collection of unordered or ordered elements. The classical set is defined in such a way that the universe of discourse (set of all possible values, also known as universal set) is divided into two groups—members and non-members.

2. A fuzzy set denoted by the tilde ($\sim$) character is a collection of values which exist between 0 and 1. In the fuzzy set, there is also a concept of partial membership.
3. In genetic algorithms, the given problem has a pool or a population of possible solutions. Crossover and mutation are applied on these solutions to produce new children. This process is repeated over several generations.
4. While decoding is the process of transforming a solution from the genotype to phenotype space, encoding, on the other hand, transforms solution from the phenotype to genotype space.
5. The most popularly used technique for selecting parents is the fitness proportionate selection. In this technique, every individual can become a parent with a probability which is proportional to its fitness. This means that fitter individuals have a higher chance of mating and propagating their features to the next generation.
6. The survivor selection policy is used to determine which individuals will be a part of the next generation and which will be discarded.



REVIEW QUESTIONS

1. Differentiate between the following.
 1. Speech and voice recognition
 2. Classical set theory and fuzzy set theory
 3. Soft computing and hard computing
 4. Artificial intelligence and soft computing
2. Explain the architecture of a fuzzy logic system.
3. Discuss the different techniques of selecting parent in case of genetic algorithms.
4. Explain the crossover technique.
5. Mutation plays an important role in genetic algorithms as it maintains as well as introduces diversity in the genetic population. Justify this statement.
6. Write short notes on genetic algorithms.
7. Using order crossover, generate two children from the given information.

Parent 1 = **1 2 3 | 4 5 6 7 | 8 9**

Parent 2 = 4 5 2 | 1 8 7 6 | 9 3

8. With a help of an example, highlight the difference between fuzzy logic and Boolean logic.
9. 'Manya is looking for a **match**.' is lexically ambiguous. Comment.
10. When is it recommended to use transfer learning?
11. Explain the different approaches to transfer learning.
12. What techniques will be used for transfer learning in case of deep learning?
13. Differentiate between traditional machine learning and transfer learning.



1. _____ ambiguity usually happens when pronouns are used.
2. _____ occurs when we omit words that are actually needed for grammatical completion of the syntax.
3. _____ uses the most plausible interpretation is used.
4. _____ is a superset of regular grammar.
5. _____ is a compiler that breaks the data coming from lexical analysis phase into smaller elements.
6. _____ provides many values between the true and false.
7. _____ systems combine fuzzy logic and neurocomputing and induces rules from _____.
8. _____ is a subset of all the possible solutions to the given problem.
9. _____ is the population (solution) in the computation space.
10. _____ is the function which accepts the solution as input and produces the suitability of the solution as the output.
11. One extremely fit solution taking up of the entire population, results in a situation known as _____.
12. In _____ technique, more than one parent is selected and one or more off-springs are produced using the genetic material of the parents.
13. _____ transfer learning requires the source and target domains to be the same.
14. In _____ approach, several tasks from the same domain are learned simultaneously without distinction between the source and targets.



STATE TRUE OR FALSE

1. A set is a collection of only ordered elements.
2. Algorithms based on fuzzy logic produce accurate results even with imprecise or inaccurate data.
3. Fuzzy system should be used when it is difficult to map an input space to an output space.
4. Chromosomes is the value which a gene takes for a particular chromosome.
5. Genotype is the population (solution) in the real-world solution space that can be represented in real world situations.
6. Generational genetic algorithm generates 'n' off-springs in each iteration.
7. Crossover is the process in which a new solution is obtained by introducing a small random tweak in the chromosome.
8. Soft computing algorithms are not affected by changes in the environment.
9. Solutions given by soft computing are uncertain as we get different solutions on solving the same problem using the same set of input data at different intervals of time.
10. Referential ambiguity exists when you are referring to something using the pronoun.



MULTIPLE CHOICE QUESTIONS

1. Identify the incorrect statement about fuzzy system.
 1. Fuzzy logic works similarly as the human reasoning.
 2. Fuzzy set theory is based on the set theory of mathematics, so users can easily relate its basic concepts.
 3. Developing a system based on fuzzy logic takes more time than developing it using conventional methods.
 4. Fuzzy logic supports the concept of partial truth, such as 0.9 or 0.5.
2. Identify the incorrect statement about genetic algorithms.
 1. Can perform parallel computations.
 2. Optimize continuous as well as discrete functions.

3. Give optimized solutions for multi-objective problems.
4. Return only one solution that best performs in all scenarios.
3. A _____ is one element position of a chromosome.
 1. Population
 2. Gene
 3. DNA
 4. Allele
4. _____ representation is used for problems in which the solution is represented by an order of elements
 1. Integer
 2. Real-valued
 3. Permutation
 4. Binary
5. _____ selection is used when the individuals in the population have very close fitness values.
 1. Rank
 2. Tournament
 3. Roulette wheel
 4. Stochastic universal sampling
6. _____ learning is the reuse of a pre-trained model on a new problem.
 1. Transfer
 2. Machine
 3. Deep
 4. Virtual
7. Transfer learning gives better performance when there is a higher _____.
 1. Start
 2. Slope
 3. Asymptote
 4. All of these
8. Identify the incorrect statement.
 1. Metaverse uses digital currencies and nonfungible tokens (NFTs).
 2. Transfer learning is a machine learning technique.
 3. In transfer learning, we use the early and middle layers of the model and retrain the latter layers.
 4. If input for two tasks is the same, then we should re-use the model.
9. _____ provides numerous pre-trained models for transfer learning.

1. Scikit
 2. numpy
 3. pandas
 4. keras
10. _____ is done when the source and target domains have different feature spaces and distributions
1. Domain adaptation
 2. Domain confusion
 3. Multi-task learning
 4. One-shot learning
11. In applications like machine translation, _____ type of deep learning is used.
1. Domain adaptation
 2. Zero-shot learning
 3. Multi-task learning
 4. One-shot learning



ANSWERS

Fill Up

1. Referential
2. Ellipsis
3. *World knowledge*
4. Context-free grammar
5. Parser
6. Fuzzy logic
7. Neuro-fuzzy, observations
8. Population
9. Genotype
10. Fitness function
11. premature convergence
12. Crossover
13. Inductive
14. Multi-task learning

State True or False

1. False
2. True
3. False
4. False
5. False
6. True
7. False
8. True
9. True
10. True

Multiple Choice Questions

1. c
2. d
3. b
4. c
5. a
6. a
7. d
8. b
9. d
10. a
11. b

Previous chapter

< [9. Where AI Is Heading Today?](#)

Next chapter

[Programming Language](#) >
[Prolog](#)

Table of contents expanded