Unit IV: Learning (10 Hours)

4.1 Introduction

4.2 Concept of Learning

4.3 Types of Learning: Supervised, Unsupervised and Reinforcement Learning

4.4 Learning by Genetic Algorithms

4.5 Learning with Neural Networks

4.5.1 Introduction, Biological Neural Networks Vs Artificial Neural Networks (ANN)

4.5.2 Mathematical Model of ANN

4.5.3 Activation Functions: Linear, Step Sigmoid

4.5.4 Types of ANN: Feed-forward, Recurrent, Single Layered, Multi-Layered

4.5.5 Application of Artificial Neural Networks, Learning by Training ANN, Perceptron Learning, Back-propagation Learning

4.1 Introduction

Learning

Learning is a key aspect of intelligence in both humans and machines.
In Artificial Intelligence (AI), learning refers to the ability of a system to improve its behavior or performance by gaining experience from data or interactions with the environment.
In AI, learning refers to the process by which a system:
Observes data or experiences
Identifies patterns or relationships
Uses those patterns to make better decisions in the future
Instead of being programmed with fixed rules, a learning system adapts its internal structure based on past observations.
The importance of learning in AI arises from the complexity and dynamic nature of real-world problems.
Many situations cannot be solved using static rules because environments change and new patterns emerge over time.
Learning allows AI systems to recognize patterns, make predictions, and take decisions automatically.
Today, learning techniques form the foundation of applications such as speech recognition, image processing, medical diagnosis, recommendation systems, robotics, and autonomous systems.

Examples of Learning in AI
Email spam filters learning to detect spam messages
Recommendation systems suggesting movies or products

4.2 Concept of Learning

In AI, learning can be understood as a process through which a system acquires knowledge and improves its performance over time.
A system is said to learn if its performance on a given task improves with experience.
This experience is usually provided in the form of data, examples, or feedback from the environment.

In simple terms: Learning = Experience + Improvement

The learning process generally involves:
• Collecting data or experiences
• Extracting useful information or patterns
• Updating internal parameters or knowledge structures
• Using the updated knowledge to perform better in future situations

Learning is evaluated using a performance measure such as accuracy, error rate, speed, or reward.
For example, if a program learns to classify emails more accurately as spam or non-spam after seeing more examples, it is considered to have learned. Thus, learning bridges the gap between raw data and intelligent behavior.

Concept of Learning using KNN (Simple Example)

Given Data (6 People)
Each person is described using two features:
Age
Cigarettes per day

Person	Age	Cigarettes/Day	Class
P1	22	10	Smoker
P2	25	12	Smoker
P3	28	15	Smoker
P4	23	0	Non-Smoker
P5	27	1	Non-Smoker
P6	30	0	Non-Smoker

New Person (Unknown Class)

Age	Cigarettes/Day
26	11

KNN Algorithm (k = 3, simple idea)
1. Measure distance between the new person and all 6 people
2. Select 3 nearest neighbors
3. Use majority voting to decide the class

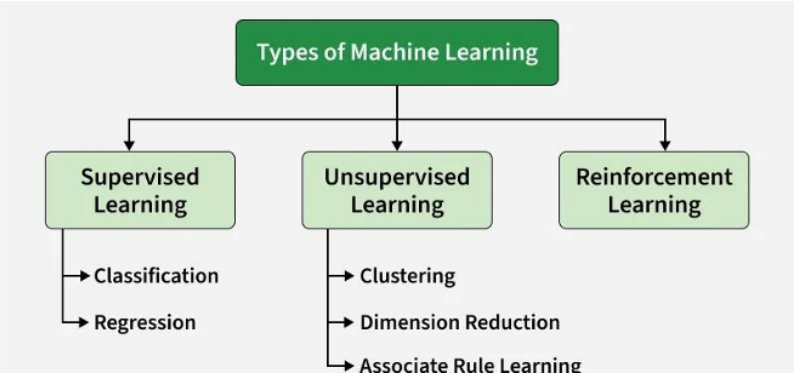
Nearest Neighbors (Closest 3)
• P2 (25, 12) → Smoker
• P1 (22, 10) → Smoker
• P3 (28, 15) → Smoker

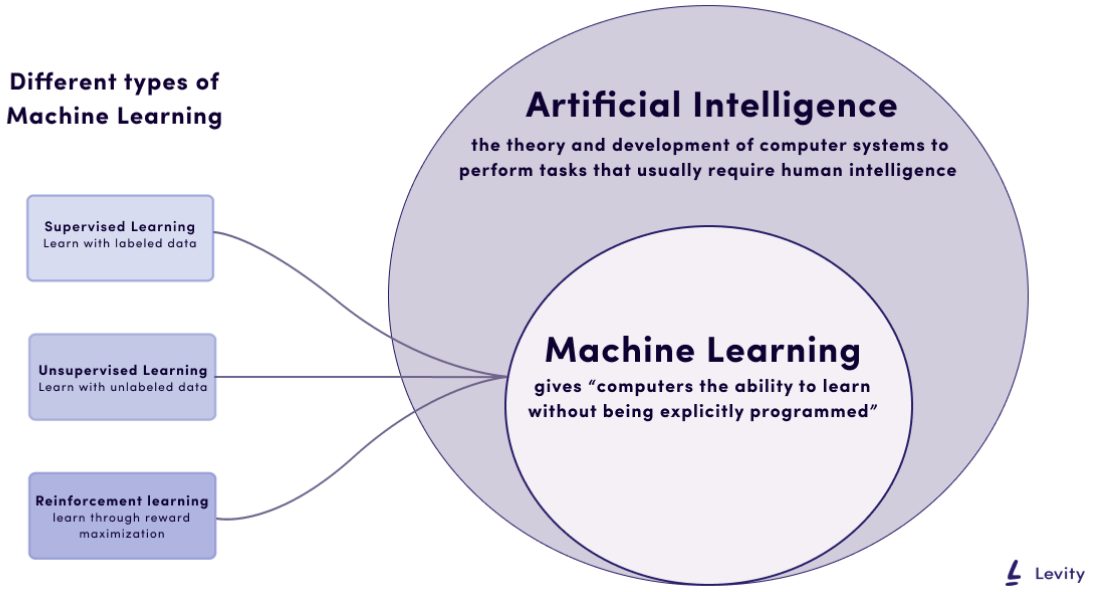
Output (Learning Result)
Smoker = 3
Non-Smoker = 0
Prediction: Smoker

Why this is Learning?
The system uses past data (experience)
It compares new data with known examples
It classifies without explicit rules
This is supervised learning using the K-Nearest Neighbors (KNN) algorithm.

4.3 Types of Learning: Supervised, Unsupervised and Reinforcement Learning

In Artificial Intelligence, learning methods are broadly classified based on how the system receives data and feedback. The three main types are:
Supervised Learning
Unsupervised Learning
Reinforcement Learning





Supervised Learning

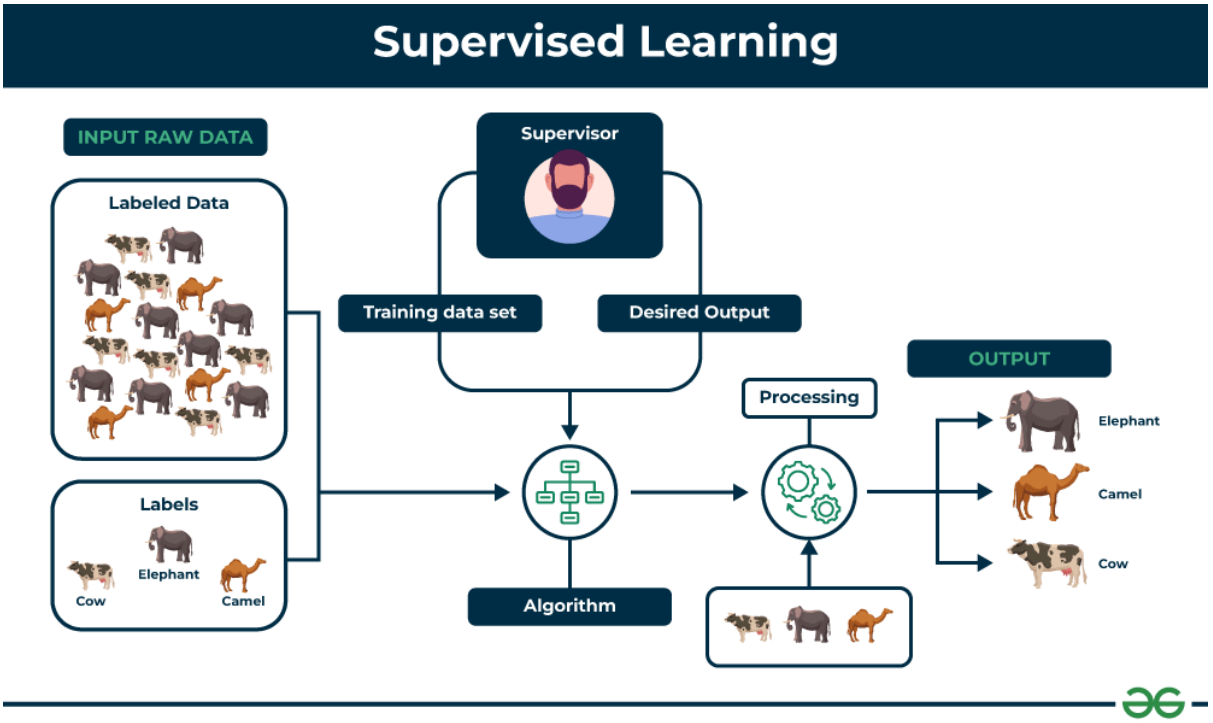
Supervised learning is a type of learning in which the system is trained using labeled data.
Each input is provided with a correct output, and the system learns by comparing its predicted output with the actual output.
The objective of supervised learning is to learn a mapping between input and output so that the system can correctly predict outputs for new, unseen data.

Key characteristics
• Uses labeled training data
• Learning is guided by a "teacher" (known output)
• Error is calculated and minimized

Common tasks
• Classification
• Regression

Examples
1. Email Spam Detection
Emails are labeled as spam or not spam. The system learns patterns and classifies new emails.

2. Student Result Prediction
Inputs: study hours, attendance, test scores Output: pass or fail



3. Salary Prediction
Inputs: experience, education, skills
Output: salary amount

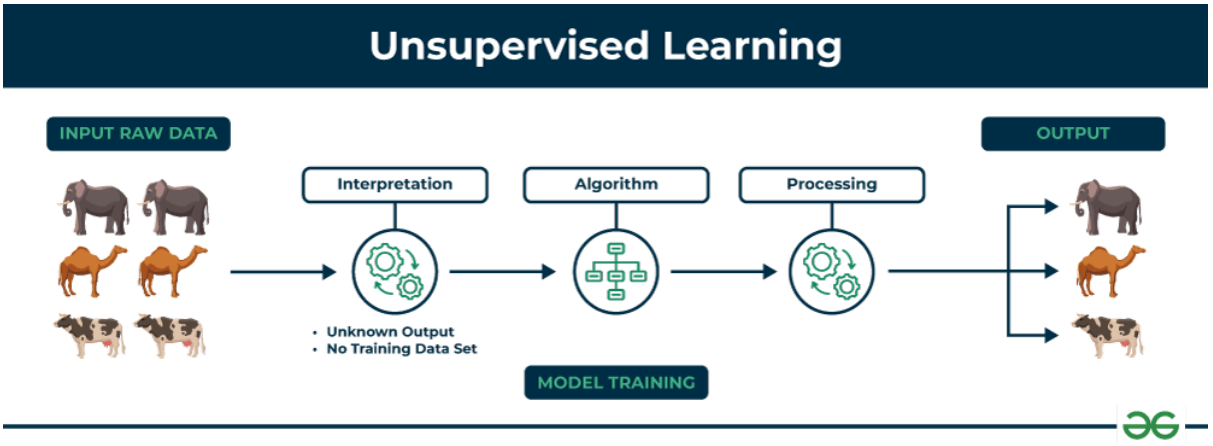
Popular algorithms
• Linear Regression
• Decision Tree
• K-Nearest Neighbors (KNN)
• Support Vector Machine (SVM)
• Neural Networks

Unsupervised Learning

Unsupervised learning is a type of learning in which the system is trained using unlabeled data.
The system does not know the correct output in advance and must discover hidden patterns or structures on its own.
The main goal of unsupervised learning is pattern discovery rather than prediction.

Key characteristics
• No labeled output data
• No teacher or guidance
• Learns structure from data

Common tasks
• Clustering
• Association
• Dimensionality reduction



Examples
1. Customer Segmentation
Customers are grouped based on buying behavior without predefined labels.

2. Grouping Students
Students are grouped based on marks or interests without naming groups in advance.

3. Market Basket Analysis
Finding products that are often bought together.

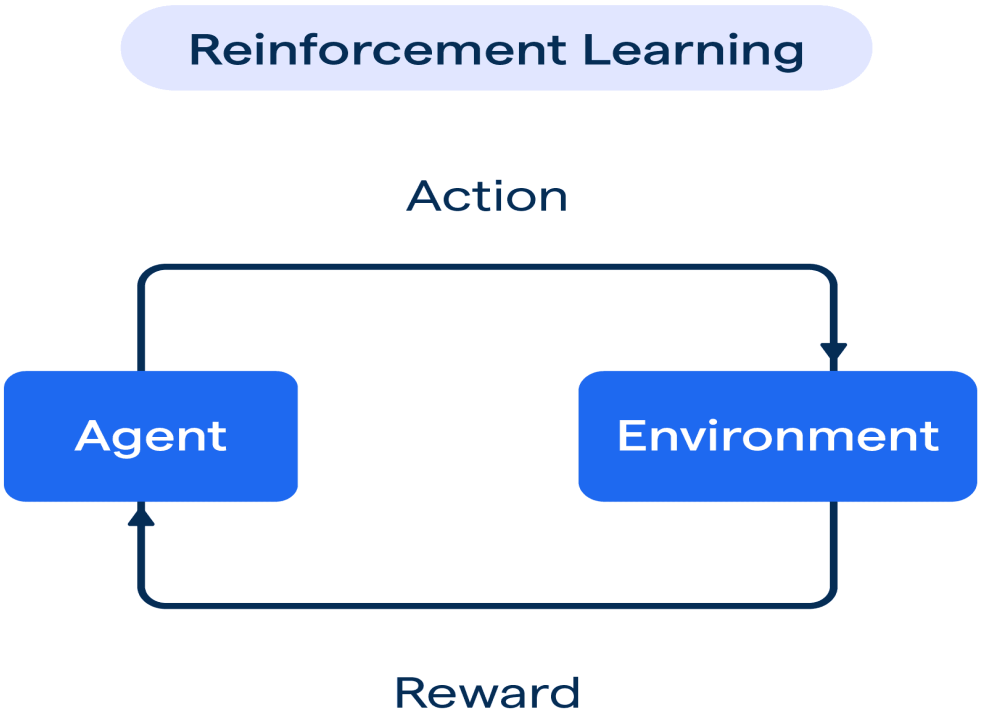
Popular algorithms
• K-Means Clustering
• Hierarchical Clustering
• Apriori Algorithm
• Principal Component Analysis (PCA)

Reinforcement Learning

Reinforcement learning is a type of learning where an agent learns by interacting with an environment.
The agent performs actions and receives feedback in the form of rewards or punishments.
The goal is to learn a strategy (policy) that maximizes total reward over time.

Key characteristics
• Learning through trial and error
• No labeled data
• Feedback is delayed
• Behavior is guided by reward or punishment

Main components
• Agent
• Environment
• Action
• Reward



Examples
1. Baby Touching Electric Socket
Pain acts as punishment, and the baby avoids the socket next time.

2. Dog Training
Dog sits → gets food (reward) → repeats action.

3. Game Playing (Chess / Video Games)
Winning gives reward; losing gives penalty.

4. Robot Navigation
Robot learns the best path by avoiding obstacles and reaching the goal.

Popular algorithms
• Q-Learning
• SARSA
• Deep Q-Network (DQN)

KNN Numerical (Smoking Prediction)
Question
You are given data of 10 people with two features:
Age (years)
Cigarettes per day
Class label: Smoker (S) or Non-Smoker (NS)
Use KNN to predict whether the new person is Smoker or Non-Smoker for k = 1, 2, 3 using Euclidean distance:

$$d=\sqrt{(Age2-Age1)2+(Cig2-Cig1)2}$$

Given Training Data (10 People)

Person	Age	Cig/Day	Class
P1	22	0	NS
P2	24	1	NS
P3	25	2	NS
P4	26	1	NS
P5	29	0	NS
P6	27	5	S
P7	30	9	S
P8	33	12	S

P9	35	14	S
P10	40	18	S

New Person (Query Point) : X = (Age = 26, Cig/Day = 4)

Step 1: Compute Distances from X

Person	Point (Age, Cig)	Class	Distance to X
P6	(27, 5)	S	√12+12=1.414
P3	(25, 2)	NS	√12+22=2.236
P4	(26, 1)	NS	√02+32=3.000
P2	(24, 1)	NS	√22+32=3.606
P5	(29, 0)	NS	√32+42=5.000
P1	(22, 0)	NS	√42+42=5.657
P7	(30, 9)	S	√42+52=6.403
P8	(33, 12)	S	√72+82=10.630
P9	(35, 14)	S	√92+102=13.454
P10	(40, 18)	S	√142+142=19.799

Step 2: Sort Nearest Neighbors (Smallest distance)
1. P6 (S) — 1.414
2. P3 (NS) — 2.236
3. P4 (NS) — 3.000

Predictions
For k = 1
Nearest neighbor = P6 (Smoker)
Prediction: Smoker

For k = 2
Neighbors: P6 (S) and P3 (NS) → 1 Smoker, 1 Non-Smoker (Tie)
Tie rule (common): choose the class of the closest neighbor
Closest is P6 (S)
Prediction (k=2): Smoker
(Note: Another tie rule is “distance-weighted voting”; but nearest-neighbor tie break is simplest.)

For k = 3
Neighbors: P6 (S), P3 (NS), P4 (NS)
Votes → Smoker = 1, Non-Smoker = 2

Prediction (k=3): Non-Smoker

Final Answer
• k = 1 → Smoker
• k = 2 → Smoker (tie → pick nearest)
• k = 3 → Non-Smoker

4.4 Learning by Genetic Algorithms

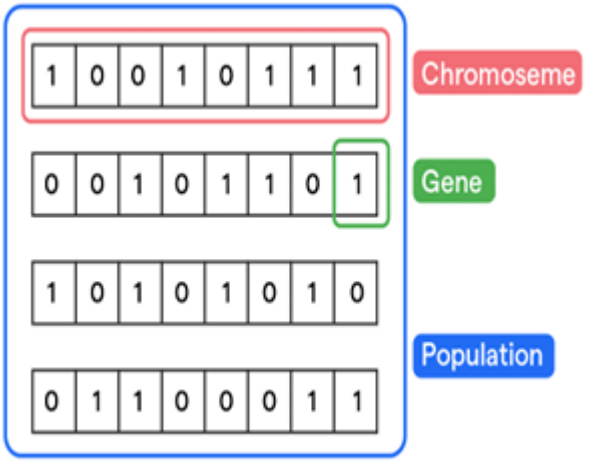
Genetic Algorithms (GA) are search and learning techniques inspired by the process of natural evolution.
They are used to find optimal or near-optimal solutions to complex problems where traditional methods are slow or ineffective.
In genetic algorithms, learning occurs by evolving better solutions over generations, rather than learning from labeled data or direct feedback.

Basic Idea of Genetic Algorithms

Genetic Algorithms imitate the principle of "survival of the fittest".
A population of possible solutions is created, and the best solutions are selected, combined, and modified to produce improved solutions.
Each solution is treated as an individual, and its quality is measured using a fitness function.

Key Components of Genetic Algorithms

1. Chromosome
A chromosome represents a possible solution to a problem.
It is usually encoded as a binary string, numbers, or symbols.
Example:
101101 → a possible solution



2. Population
A population is a set of chromosomes (solutions) at a given time.
Example:
{101101, 110010, 011011, 100111}

3. Fitness Function
The fitness function evaluates how good a solution is.
• Higher fitness → better solution
• Lower fitness → weaker solution
Example:
Fitness = total profit, accuracy, or performance score

4. Selection
Selection chooses the best chromosomes for reproduction.
Common methods:
• Roulette wheel selection
• Tournament selection
Better solutions have a higher chance of being selected.

5. Crossover
Crossover combines two parent chromosomes to produce new offspring.
Example:
Parent 1: 101|011
Parent 2: 110|100
After crossover:
Child 1: 101100
Child 2: 110011

6. Mutation
Mutation randomly changes some bits to maintain diversity.
Example:
Before mutation: 101100
After mutation: 101110
Mutation helps avoid local optimum solutions.

Genetic Algorithm Learning Process

1. Initialize population randomly
2. Evaluate fitness of each chromosome
3. Select best chromosomes
4. Apply crossover
5. Apply mutation

6. Create new population
7. Repeat until stopping condition is met

Suitable Examples

Example 1: Exam Timetable Scheduling
A college wants to create an exam timetable with:
• No subject clashes
• Minimum gaps between exams
GA generates multiple timetables, evaluates them using a fitness function, and evolves better schedules over generations.

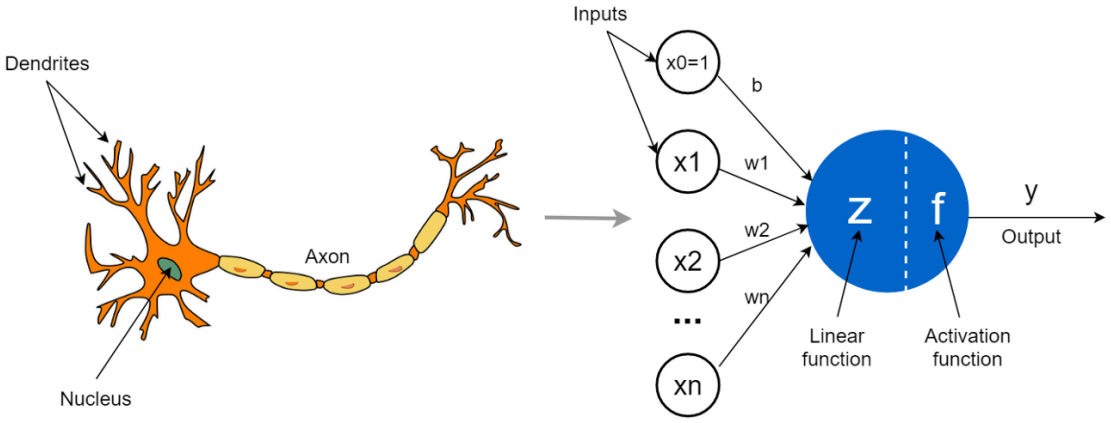
Example 2: Feature Selection in Machine Learning
From many features, GA selects the best subset that gives maximum accuracy.
Poor feature combinations are eliminated, and better ones are evolved.

4.5 Learning with Neural Networks

Learning with Neural Networks refers to the process by which a system learns patterns, relationships, or functions from data by adjusting internal parameters called weights.
Neural Networks are inspired by the working of the human brain, where learning occurs through experience and repeated exposure.
A neural network consists of interconnected processing units called neurons. These neurons work together to process input data and produce an output.
Learning in neural networks happens when the network modifies its weights to reduce errors between the predicted output and the actual output.

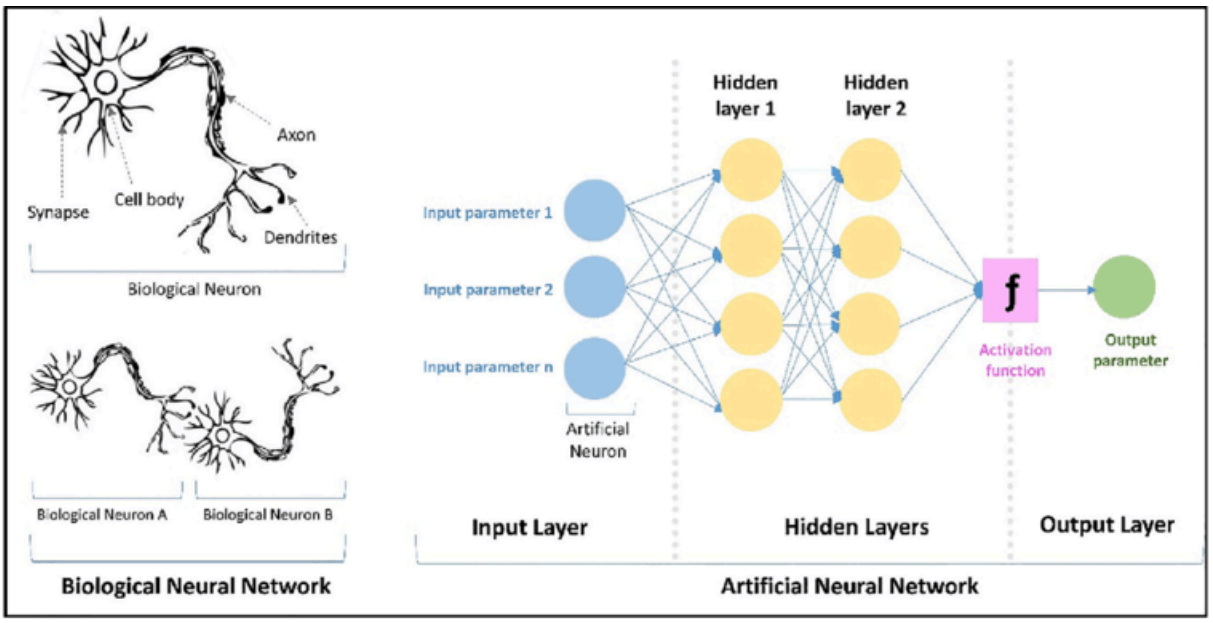
Neural Networks are widely used because they can:
• Learn complex and non-linear relationships
• Generalize knowledge to unseen data
• Improve performance with more training data

Applications include image recognition, speech recognition, handwriting recognition, medical diagnosis, stock prediction, and natural language processing.



4.5.1 Introduction, Biological Neural Networks Vs. Artificial Neural Networks (ANN)

Biological Neural Networks	Artificial Neural Networks (ANN)
Biological Neural Networks exist in the human brain and nervous system.	Artificial Neural Networks are computer-based models inspired by biological neurons.
Key characteristics:	Key characteristics:
• Neurons are biological cells	• Neurons are mathematical functions
• Communication occurs through electrical and chemical signals	• Communication happens through numerical weights
• Learning happens by strengthening or weakening synapses	• Learning occurs by adjusting weights using algorithms
• Extremely complex and highly parallel	• Faster and scalable on machines
• Capable of learning, reasoning, and creativity	• Designed for specific tasks
Example: Humans learning to recognize faces or languages through experience.	Example: A neural network trained to recognize handwritten digits or predict house prices.



BNNs are the foundation of cognition in living organisms. They consist of biological components like dendrites, cell bodies, and axons. Dendrites receive signals, the soma integrates them, and axons transmit the output to other neurons via synapses. BNNs excel at parallel processing, handling ambiguous inputs, and adapting to real-time learning. However, they lack centralized control and operate at slower speeds due to electrochemical transmission.

ANNs, on the other hand, are computational models inspired by BNNs. They consist of input, hidden, and output layers, where data flows in a structured manner. ANNs are designed for specific tasks, such as pattern recognition and forecasting, and rely on mathematical weights adjusted during training. While they are faster and more precise, ANNs are hardware-dependent and lack interpretability due to their black-box nature.

4.5.2 Mathematical Model of ANN

A mathematical model of an Artificial Neural Network (ANN) expresses how input variables are mathematically transformed through layers of neurons to produce an output prediction.
In the MDPI study, ANN was used to predict nickel removal efficiency based on input parameters (initial concentration, adsorbent dosage, pH).
An ANN captures complex non-linear relationships between inputs and outputs by learning weights and biases through training data.

The mathematical model of ANN explains how an artificial neuron receives inputs, processes them using weights and bias, applies an activation function, and produces an output.
This model is inspired by biological neurons but is completely mathematical.

Components of the Mathematical Model

An artificial neuron consists of:
1. Inputs
Signals from other neurons or external sources.
$x1, x2, x3, \ldots, xn$

2. Weights
Each input is multiplied by a weight representing its importance.
$w1, w2, w3, \ldots, wn$

3. Bias
A threshold term that allows the neuron to shift the activation function.
b

4. Summation Function
Computes the weighted sum of inputs:

$$z = \sum_{i+1}^{n} w_i x_i + b$$

5. Activation Function
Determines the output of the neuron.
$y=f(z)$

Numerical Example
Given:
$x1=1,\ x2=2$
$w1=0.4,\quad w2=0.6\quad b=0.2$
Activation: Sigmoid
Step 1:
$Z = (1\times0.4)+(2\times0.6)+0.2=1.8$
Step 2:
$Y = 1/1+e\text{pow}(-1.8)$
≈0.858

4.5.3 Activation Functions: Linear, Step Sigmoid

Activation functions in neural networks decide whether a neuron should be "activated" (i.e., contribute to the output).
They introduce non-linearity, which allows neural networks to model complex relationships.

a) Linear Function
Output is directly proportional to input.
$\mathbf{f}(z)=z$

(b) Step (Threshold) Function
Produces a binary output (0 or 1)

$$\mathbf{f}(z)= \begin{cases} \mathbf{1}, z \geq \mathbf{0} \\ \mathbf{0}, z < \mathbf{0} \end{cases}$$

(c) Sigmoid Function
Smooth and differentiable.
Output range: 0 to 1

$$f(z) = \frac{1}{1+e^{-z}}$$

4.5.4 Types of ANN: Feed-forward, Recurrent, Single Layered, Multi-Layered

ANN has 3 layers i.e. Input layer, Hidden layer, and Output layer. Each ANN has a single input and output but may also have none, one or many hidden layers.
Neural networks can be classified based on architecture (structure of connections) and number of layers.

A. Based on Architecture

1. Feed-Forward Neural Network (FFNN)

Signals move only forward: from input → hidden → output.
No loops or cycles.
Simple and widely used.
Use Cases: Pattern recognition, image classification.

Diagram:
Input → Hidden → Output

2. Recurrent Neural Network (RNN)

Has feedback loops, i.e., outputs can affect future inputs.
Can maintain a form of memory of previous inputs.
Use Cases: Time-series prediction, speech recognition, language modeling.

B. Based on Number of Layers

1. Single-Layer Neural Network

Contains only one layer of neurons (the output layer) connected to input.
Simple, but can solve only linearly separable problems (e.g., AND, OR).
Example: Perceptron

2. Multi-Layer Neural Network

Has one or more hidden layers between input and output.
Can solve non-linear problems.
Uses backpropagation for learning.
Use Cases: Handwriting recognition, speech recognition, image classification.

4.5.5 Application of Artificial Neural Networks, Learning by Training ANN, Perceptron Learning, Back-propagation Learning

Applications of Artificial Neural Networks (ANN)

Artificial Neural Networks are widely used because they can learn patterns, handle non-linear data, and adapt from experience.

Major Applications

1. Pattern Recognition
o Face recognition
o Handwritten digit recognition (MNIST)
o Speech recognition

2. Prediction and Forecasting
o Stock market prediction
o Weather forecasting
o Sales and demand forecasting

3. Classification
o Spam vs non-spam email
o Disease diagnosis (positive / negative)
o Credit approval systems

4. Image and Signal Processing
o Image enhancement
o Object detection
o Noise removal

5. Natural Language Processing
o Language translation
o Chatbots
o Text summarization

6. Control Systems
o Robotics
o Autonomous vehicles
o Industrial process control

Key Point:
ANNs are used where rule-based programming is difficult.

Learning by Training ANN

Training
Training is the process by which an ANN learns from data by adjusting its weights and biases to reduce error.

Training Process Steps
1. Provide input data
2. Perform forward propagation
3. Generate output
4. Compare with target output
5. Calculate error
6. Adjust weights using a learning algorithm
7. Repeat until error is minimized

Mathematically:

$$\text{Error} = \text{Target} - \text{Output}$$

Learning happens iteratively over many epochs.

Perceptron Learning

Perceptron
A perceptron is the simplest neural network model with:
• Single neuron
• Binary output (0 or 1)

Perceptron Model Equation

$$y = f\left(\sum w_i x_i + b\right)$$

Where:
• f= step (threshold) function

Learning Rule (Perceptron Learning Rule)

$$w_{new} = w_{old} + \eta(t - y)x$$

Where:
• η= learning rate
• t= target output
• y= predicted output

Characteristics
Works only for linearly separable problems
Cannot solve XOR problem
Fast and simple

Example: AND, OR logic gates

Back-Propagation Learning

Back-Propagation
Back-propagation is a supervised learning algorithm used to train multi-layer neural networks.
It minimizes error by propagating the error backward from output layer to hidden layers.

Back-Propagation Steps

1. Initialize weights randomly
2. Perform forward propagation
3. Calculate error at output layer
4. Propagate error backward
5. Update weights using gradient descent
6. Repeat until convergence

Error Function (Example: Mean Squared Error)

$$E = \frac{1}{2}(t - y)^2$$

Weight Update Rule

$$w_{new} = w_{old} - \eta \frac{\delta E}{\delta w}$$