ASSIGNMENT - 01

EAI501

ARTIFICIAL NEURAL NETWORK

Submitted To:

Mr. Nripesh Chauhan
Asst. Professor
CCSIT, TMU

Submitted By:

Samrat Singh
TCA2359180
B.Tech.(CSE-AI), 5th Sem
Section - C

TABLE OF CONTENTS

Introduction to Neural Network

A **Neural Network** is a computational approach that attempts to model the way the human brain processes information. It is a fundamental concept in **Artificial Intelligence (AI)** and **Machine Learning (ML)**, and it has gained significant attention due to its effectiveness in solving complex tasks that traditional algorithms struggle with, such as image recognition, language translation, and predictive analytics.

Biological Inspiration

Neural networks are inspired by the structure and functioning of the **biological brain**, particularly the way **neurons** communicate with each other. In the human brain, neurons receive signals from other neurons, process them, and pass on the result to yet other neurons. This communication is what enables the brain to perceive, learn, and make decisions. Artificial neural networks (ANNs) mimic this process using mathematical functions and data structures.

Basic Building Blocks

A neural network is made up of **layers of nodes (neurons)**:

- Each **neuron** is a simple computing unit that takes one or more input values, processes them, and produces an output.
- Neurons are organized in **layers**, starting with an **input layer**, followed by one or more **hidden layers**, and ending with an **output layer**.
- Every connection between neurons is assigned a **weight**, which determines the strength or influence of one neuron on another.
- A **bias** is often added to the weighted sum to increase the model's flexibility.

History of Artificial Neural Network

1. Early Foundations (1940s – 1950s)

The roots of ANNs lie in attempts to model the human brain computationally.

- **1943 – McCulloch & Pitts Model**
 Warren McCulloch (neurophysiologist) and Walter Pitts (logician) proposed the first mathematical model of a **neuron**. Their artificial

neuron accepted inputs, applied weights, and produced a binary output (0 or 1). This was the **foundation of neural computing**.

- **1949 – Hebbian Learning**
 Donald Hebb introduced the idea that **connections between neurons strengthen when they are activated together**. This is famously summarized as *"neurons that fire together, wire together"*. It became a key principle of learning in neural networks.

- **1957 – Perceptron**
 Frank Rosenblatt developed the **Perceptron**, the first neural network model capable of learning from data. It could classify inputs into two categories by adjusting weights based on errors. It was seen as a step toward machine intelligence.

2. The First AI Winter (1960s – 1970s)

Excitement quickly turned into skepticism.

- **1969 – Minsky & Papert's Criticism**
 In the book *Perceptrons*, Marvin Minsky and Seymour Papert showed that the perceptron was limited—it could not solve simple problems like the XOR function.
 Funding for neural network research declined, leading to the **first AI winter**.

- During this time, **symbolic AI** (rule-based systems) became more popular than neural networks.

3. Revival with Multilayer Networks (1980s)

Neural networks re-emerged when researchers found ways to overcome perceptron limitations.

- **1970s – 1980s – Backpropagation Algorithm**
 Paul Werbos (1974), and later Rumelhart, Hinton, and Williams (1986), introduced **backpropagation**, an efficient method to train **multilayer perceptrons (MLPs)**. This allowed networks to learn complex,

non-linear functions like XOR.

- **Hopfield Networks (1982)**
  John Hopfield created networks for associative memory, showing neural networks could solve optimization problems.

- **Boltzmann Machines (1985)**
  Geoffrey Hinton and others developed probabilistic neural models that could learn internal representations.

- Neural networks began to be applied in **speech recognition, character recognition, and control systems**.

4. The Second AI Winter (1990s – early 2000s)

Despite progress, ANNs faced challenges.

- **Computational Limitations**: Neural networks required large computational resources and data, which were not widely available.

- **Rise of Support Vector Machines (SVMs) & Statistical Methods**: These models often outperformed ANNs on smaller datasets.

- Neural networks were seen as difficult to train and prone to overfitting.

5. The Deep Learning Era (2006 – Present)

The true breakthrough came with **deep neural networks (DNNs)**.

- **2006 – Hinton's Deep Belief Networks**
  Geoffrey Hinton introduced deep learning techniques, showing that neural networks could learn hierarchical representations.

- **2012 – AlexNet & ImageNet Competition**
  Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton built **AlexNet**, a deep convolutional neural network (CNN). It drastically outperformed traditional methods in image recognition, sparking the deep learning revolution.

- **Advancements in Architectures**
  - CNNs (for vision tasks like object detection, face recognition).
  - RNNs & LSTMs (for sequential data like speech and text).
  - GANs (2014, for image generation).
  - Transformers (2017, revolutionized NLP with models like BERT, GPT).
- **Enablers of Success**
  - Availability of **big data**.
  - **GPUs and TPUs** for fast computation.
  - Improved **training algorithms** and **regularization techniques**.

6. Current Trends (2020s – Future)

- **Foundation Models & Large Language Models (LLMs)** like GPT, BERT, and PaLM use **transformers** with billions of parameters.
- Applications include **autonomous vehicles, healthcare, finance, natural language processing, robotics, and generative AI**.
- Research now explores **explainability, efficiency, ethical AI, neuromorphic computing, and brain-inspired architectures**.

Components of Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) is composed of several fundamental components that work together to process information, learn patterns from data, and make decisions or predictions. Each component plays a specific role in the data flow, learning, and performance of the network. Below are the core components of an ANN:

1. Neurons (Nodes or Units)

A **neuron** is the basic processing unit of a neural network, modeled loosely after a biological neuron. Each neuron receives inputs, processes them, and generates an output.

- **Input**: Receives one or more input signals (numerical values).
- **Processing**: Applies a mathematical operation — usually a **weighted sum** of inputs plus a **bias**.
- **Output**: The result is passed through an **activation function** and sent to the next layer or output.

2. Layers

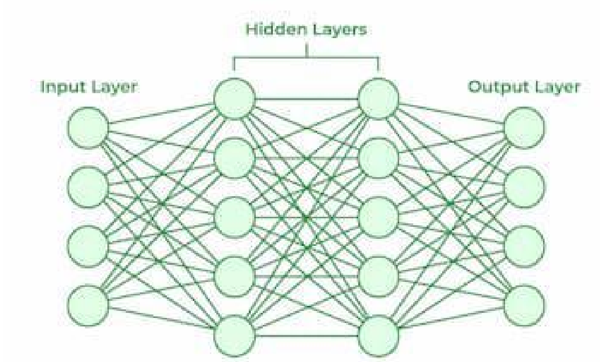
ANNs are organized into layers of neurons:

- **Input Layer**: The first layer that receives raw data (features of the dataset). It doesn't perform computations—only passes inputs to the next layer.
- **Hidden Layers**: One or more layers between the input and output layers. They perform computations using neurons, weights, and activation functions. More hidden layers often mean the model can learn more complex patterns.
- **Output Layer**: Produces the final output of the network (e.g., a classification label or a numerical prediction).

3. Weights

Each connection between neurons carries a **weight**, a real number that determines the importance of the input signal. Weights are the primary **learnable parameters** in a neural network.

- Initially assigned randomly.
- Updated during training to reduce error.



- Higher weights indicate stronger influence on the neuron's output.

4. Bias

Bias is an additional learnable parameter added to the neuron's input. It helps shift the activation function and improves the model's flexibility.

- Allows the network to fit the data better by adjusting the threshold at which neurons activate.
- Bias is added to the weighted sum before applying the activation function.

5. Activation Function

The **activation function** introduces **non-linearity** into the network, enabling it to learn complex relationships in data.

Common activation functions include:

- **Sigmoid**: Outputs between 0 and 1; good for binary classification.
- **ReLU (Rectified Linear Unit)**: Efficient and commonly used in deep networks.
- **Tanh**: Outputs between -1 and 1; zero-centered.
- **Softmax**: Used in multi-class classification to produce probability distribution.

6. Loss Function (Cost Function)

The **loss function** quantifies the difference between the network's predicted output and the actual target values.

- It guides how the model learns.
- The smaller the loss, the better the model's performance.
- Common loss functions:
    - **Mean Squared Error (MSE)** for regression
    - **Cross-Entropy Loss** for classification

7. Optimizer

An **optimizer** is an algorithm used to update the weights and biases of the network to minimize the loss function.

Popular optimizers include:

- Gradient Descent
- Stochastic Gradient Descent (SGD)
- Adam (Adaptive Moment Estimation)

Learning Process of Artificial Neural Network (ANN)

The learning process in an Artificial Neural Network (ANN) refers to how the network **adjusts its internal parameters (weights and biases)** to minimize the error between its predictions and the actual outcomes. This process allows the ANN to recognize patterns, make predictions, and generalize knowledge from training data to unseen data.

The learning process typically involves **four major stages**:

1. Forward Propagation

Objective: To compute the output of the neural network given a set of input data.

Steps:

- The input data is fed into the **input layer**.
- Each neuron in the hidden and output layers computes a **weighted sum** of its inputs, adds a **bias**, and applies an **activation function** to produce its output.
- This process continues layer-by-layer until the final **output** is produced by the **output layer**.

2. Loss Calculation (Error Computation)

Objective: To measure how far the network's output is from the expected (true) output.

- A **loss function** (or cost function) is used to compute the difference between the actual output and the predicted output.
- The choice of loss function depends on the type of problem:
    - **Mean Squared Error (MSE)** for regression tasks.
    - **Cross-Entropy Loss** for classification tasks.

3. Backward Propagation (Backpropagation)

Objective: To compute the **gradient of the loss function** with respect to each weight and bias, so we know how to adjust them to reduce the error.

How it works:

- **Backpropagation** applies the **chain rule of calculus** to propagate the error backward from the output layer to the input layer.
- It calculates how much each **weight and bias** contributed to the error.
- The computed gradients are then used to update the parameters.

Key Concepts:

- **Gradient**: Derivative of loss with respect to weights/biases.
- **Partial derivatives**: Measure how small changes in weights affect the loss.

4. Weight and Bias Update (Optimization)

Objective: To update the weights and biases using the gradients calculated during backpropagation to minimize the loss.

- This step uses an **optimization algorithm**, the most basic being **Gradient Descent**.

Gradient Descent Formula:

$$w = w - \eta \frac{\partial L}{\partial w}$$

$$b = b - \eta \frac{\partial L}{\partial b}$$

Where:

- w, b: weights and biases,

- η : learning rate,

- $\frac{\partial L}{\partial w}$: gradient of the loss with respect to the weight.

Variants of Optimization Algorithms:

- Stochastic Gradient Descent (SGD)
- Adam (Adaptive Moment Estimation)
- RMSprop, etc.

Applications of Artificial Neural Networks

1. **Computer Vision** – Face recognition, medical imaging, handwriting recognition.

2. **Natural Language Processing (NLP)** – Translation, chatbots, speech recognition.

3. **Healthcare** – Disease diagnosis, drug discovery, patient monitoring.

4. **Finance** – Stock prediction, fraud detection, credit scoring.

5. **Robotics & Autonomous Systems** – Self-driving cars, drones, industrial robots.

6. **Manufacturing** – Predictive maintenance, process optimization, quality control.

7. **Security** – Cybersecurity, biometrics, surveillance.

8. **Entertainment** – Recommendation systems (Netflix, YouTube), gaming AI, AI art.

9. **Agriculture** – Crop disease detection, precision farming, drone monitoring.

10. **Emerging Areas** – Smart cities, personalized education, climate modeling.

Biological Neural Network (BNN)

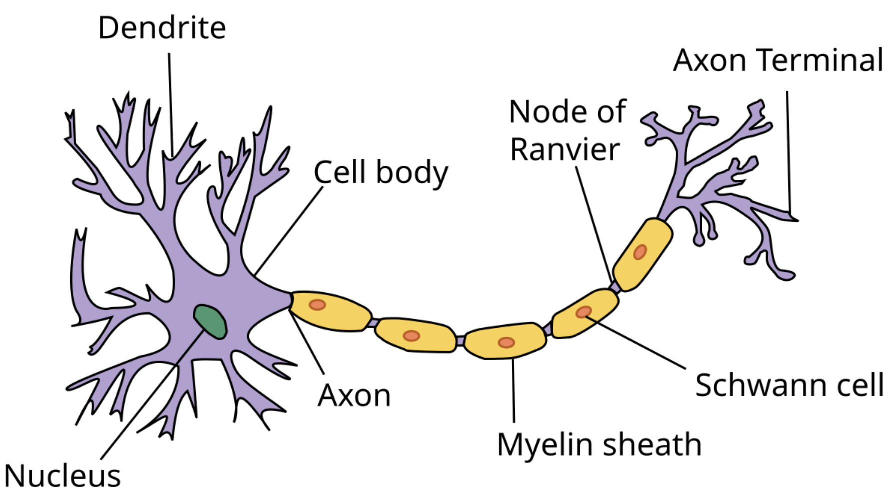
A **Biological Neural Network (BNN)** refers to the **network of neurons in a living brain or nervous system**. It is the natural system on which Artificial Neural Networks (ANNs) are inspired.

In simple words, **BNNs are the brain's information-processing systems**, made of billions of interconnected **neurons (nerve cells)** that communicate through electrical and chemical signals.

Structure of BNN

1. Neuron (Basic Unit)

- Has three main parts:
  - **Dendrites** – receive signals from other neurons.



        - **Cell Body (Soma)** – processes the information.
        - **Axon** – transmits signals to other neurons.

2. ### Synapse (Connection Point)

    - Neurons communicate via synapses using **neurotransmitters** (chemical messengers).
    - Strength of a synapse changes with learning (synaptic plasticity).

3. ### Signal Transmission

    - Information passes as **electrical impulses (action potentials)**.
    - When the impulse reaches the axon terminal, neurotransmitters are released to activate the next neuron.

Working of Biological Neural Network (BNN)

A **Biological Neural Network** refers to the complex web of interconnected neurons in **living organisms**, especially in the **human brain**. These networks are responsible for sensing, processing, and responding to stimuli. Here's how they work:

1. Signal Transmission Process

The functioning of a biological neural network involves **electrochemical signaling**:

a) Reception (Input)

- Dendrites receive **chemical signals** (neurotransmitters) from other neurons.
- These chemicals bind to receptors on the dendrites and trigger **electrical changes** in the neuron.

b) Integration

- The **cell body** integrates all incoming signals.
- If the combined electrical signal (membrane potential) crosses a certain **threshold**, the neuron fires an **action potential**.

c) Firing (Action Potential)

- An action potential is a rapid electrical impulse that travels down the **axon**.
- This is an **all-or-nothing** event—either the neuron fires or it doesn't.

d) Transmission

- When the action potential reaches the axon terminal:
  - It triggers the release of **neurotransmitters** into the **synaptic cleft**.
  - These chemicals travel to the next neuron’s dendrites, continuing the chain.

2. Synaptic Plasticity (Learning Mechanism)

Biological learning is based on changes in synapses:

- **Hebbian Learning**: "Cells that fire together, wire together." If two neurons are repeatedly active together, the synaptic connection between them strengthens.
- **Long-Term Potentiation (LTP)**: Strengthening of synapses based on frequent activity.
- **Long-Term Depression (LTD)**: Weakening of synapses due to inactivity or certain patterns.

These changes form the **basis of memory and learning**.

3. Parallel and Distributed Processing

- The brain contains ~86 billion neurons.
- Signals are processed **in parallel** and distributed across different brain regions.
- There's **redundancy**, allowing for fault tolerance and robustness.

4. Key Features of Biological Neural Networks

Feature	Description

Analog Signals	Operate with continuous values, not just binary.
Learning via Synaptic Changes	Adapt based on experience and environment.
Massive Parallelism	Multiple processes occur simultaneously.
Noise Tolerance	Functions reliably even with imperfect signals.
Energy Efficient	Much more efficient than artificial systems.

ANN vs BNN

Aspect	Artificial Neural Network (ANN)	Biological Neural Network (BNN)
Definition	A computational model inspired by the brain, built using algorithms and mathematics	The actual network of neurons in biological brains, like those in humans or animals

Structure	Layers of artificial neurons (nodes) arranged in input, hidden, and output layers	Complex, irregular network of billions of neurons connected via synapses
Neurons	Abstract units performing mathematical operations	Real biological cells transmitting electrochemical signals
Signal Type	Digital or numeric values (e.g., floating-point numbers)	Electrochemical impulses (action potentials and neurotransmitters)
Learning Mechanism	Gradient descent, backpropagation, weight optimization	Synaptic plasticity, Hebbian learning, long-term potentiation/depression
Data Flow	Usually unidirectional (feedforward) or bidirectional (in recurrent nets)	Highly parallel and complex, with feedback loops and inhibitory/excitatory effects
Speed	Extremely fast in computation (nanoseconds to milliseconds)	Slower (milliseconds per signal transmission)
Adaptability	Requires training on large datasets	Learns from few examples and experiences; adapts to new environments
Fault Tolerance	Prone to error if weights are corrupted or data is missing	Highly fault-tolerant and robust due to redundancy in neural pathways

Energy Efficiency	Consumes significant computational power and energy	Very efficient; the human brain uses ~20 watts to function
Memory	External memory systems (RAM, disk, etc.)	Memory encoded in neural and synaptic connections
Hardware Dependency	Runs on silicon chips, GPUs, TPUs	Based in organic brain tissue
Interpretability	Can be opaque ("black box"); difficult to explain decisions	Still not fully understood, but evidence of pattern-based and experience-based reasoning
Learning Speed	Requires multiple training epochs	Can learn quickly through one-time exposure or experience