Experiment 2: Design of Logic Gates using Neural Network

AIM: The aim of this experiment is to understand the working of the basic Neural Networks and design the basic logic gates using neural network

Problem Definition: Design and implement basic logic gates (AND, OR, NOT, XOR) using artificial neural networks.

Theory: Neural Networks are networks of interconnected neurons, for example in human brains. Artificial Neural Networks are highly connected to other neurons, and performs computations by combining signals from other neurons. Outputs of these computations may be transmitted to one or more other neurons. The neurons are connected together in a specific way to perform a particular task. A neural network is a function. It consists of basically:

a. Neurons: which pass input values through functions and output the result.

b. Weights: which carry values (real-number) between neurons.

Neurons can be categorized into layers:

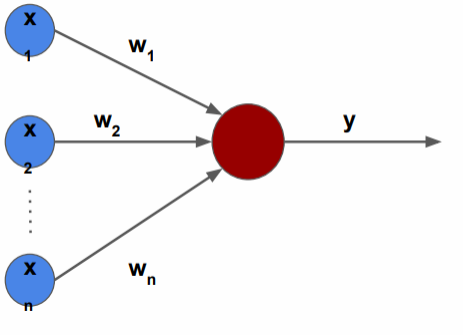
a. Input Layer

b. Hidden Layer

c. Output Layer

Perceptron Neural Network is motivated by the biological neuron. A perceptron is a computing element where inputs are associated with the weights and the cell having a

threshold value.



$$y = \begin{cases} 1, & \text{if } \sum w_i x_i > \text{threshold} \\ 0 & \text{otherwise} \end{cases}$$

Activation functions determine whether a neuron should "activate" by processing the input data. They act as filters, allowing only relevant information to pass through. Common functions include:

- **Sigmoid:** Smooth curve used for binary activations.
- **ReLU (Rectified Linear Unit):** Popular in modern networks due to its simplicity and effectiveness of passing only positive values.

During training, neural networks make predictions based on the input data. When these predictions are incorrect, the network adjusts the connections (weights) between neurons. This iterative process helps the network improve over time, similar to learning from mistakes.

Neural networks come in various architectures, each tailored for specific types of tasks. Here is a quick overview:

Type	Use Case	Description
Feed-Forward Neural Network	Standard processing from input to output	Data flows in one direction without looping
Convolutional Neural Network (CNN)	Image analysis and pattern recognition	Excels at processing grid-like topology such as images
Recurrent Neural	Sequential data	Ideal for handling time-series or

Network (RNN)	processing (e.g., text, audio)	sequence-dependent data