SNS COLLEGE OF TECHNOLOGY

An Autonomous Institution

Coimbatore-35

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A++’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ECT303-ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

III YEAR/ V SEMESTER

UNIT 4 – NEURONS AND NEURAL NETWORKS

4.5 Multilayer Perceptron

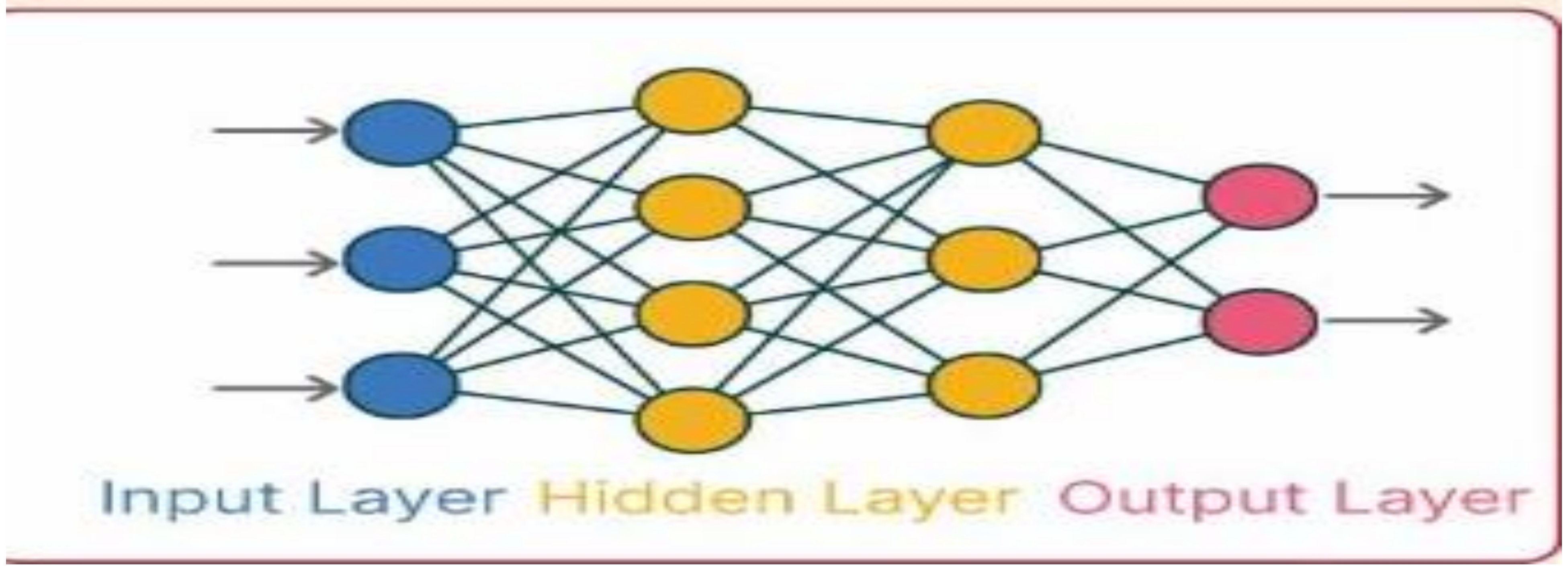
MULTILAYER PERCEPTRON

Introduction

MultiLayer Perceptron Neural Network is a Neural Network with multiple layers, and all its layers are connected.

It uses a **BackPropagation algorithm** for training the model.

Multilayer Perceptron is a class of Deep Learning, also known as MLP.



MULTILAYER PERCEPTRON

What is MultiLayer Perceptron Neural Network?

- It belongs to the **feedforward neural network.**
    - The Multilayer Perceptron (MLP) Neural Network works only in the forward direction.
    - Each node passes its value to the coming node only in the forward direction.
- It is an Artificial Neural Network in which all nodes are interconnected with nodes of different layers.
- First defined by ***Frank Rosenblatt*** in his perceptron program.
- Defines the artificial neuron in the neural network.
- It is a **supervised learning algorithm** that contains **nodes' values, activation functions, inputs, and node weights to calculate the output.**
- All nodes are fully connected to the network.
- It uses a **Back propagation algorithm** to increase the accuracy of the training model.

MULTILAYER PERCEPTRON

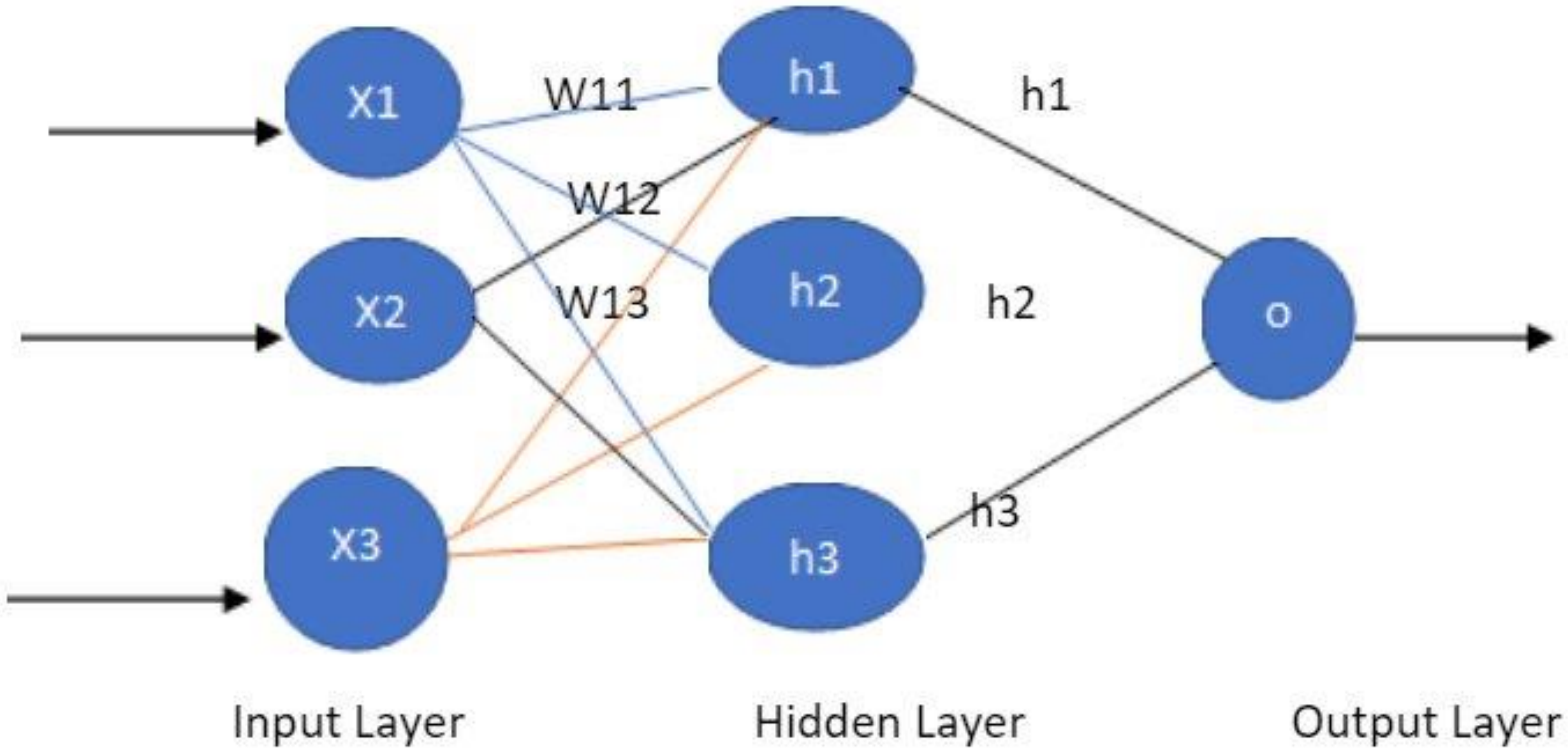


Diagram Of MultiLayer Perceptron Neural Network

MULTILAYER PERCEPTRON

Structure of MultiLayer Perceptron Neural Network -3

Input Layer

- It is the initial or starting layer of the Multilayer perceptron.
- It takes input from the training data set and forwards it to the hidden layer.
- There are n input nodes in the input layer.
- The number of input nodes depends on the number of dataset features.
- Each input vector variable is distributed to each of the nodes of the hidden layer.

Hidden Layer

- Heart of all Artificial neural networks and comprises all computations of the neural network.
- The edges of the hidden layer have weights multiplied by the node values.
- This layer uses the activation function.
- There can be one or two hidden layers in the model.
- Several hidden layer nodes should be accurate as few nodes in the hidden layer make the model unable to work efficiently with complex data.
- More nodes will result in an over fitting problem.

MULTILAYER PERCEPTRON

Output Layer

It gives the estimated output of the Neural Network.

The number of nodes in the output layer depends on the type of problem.

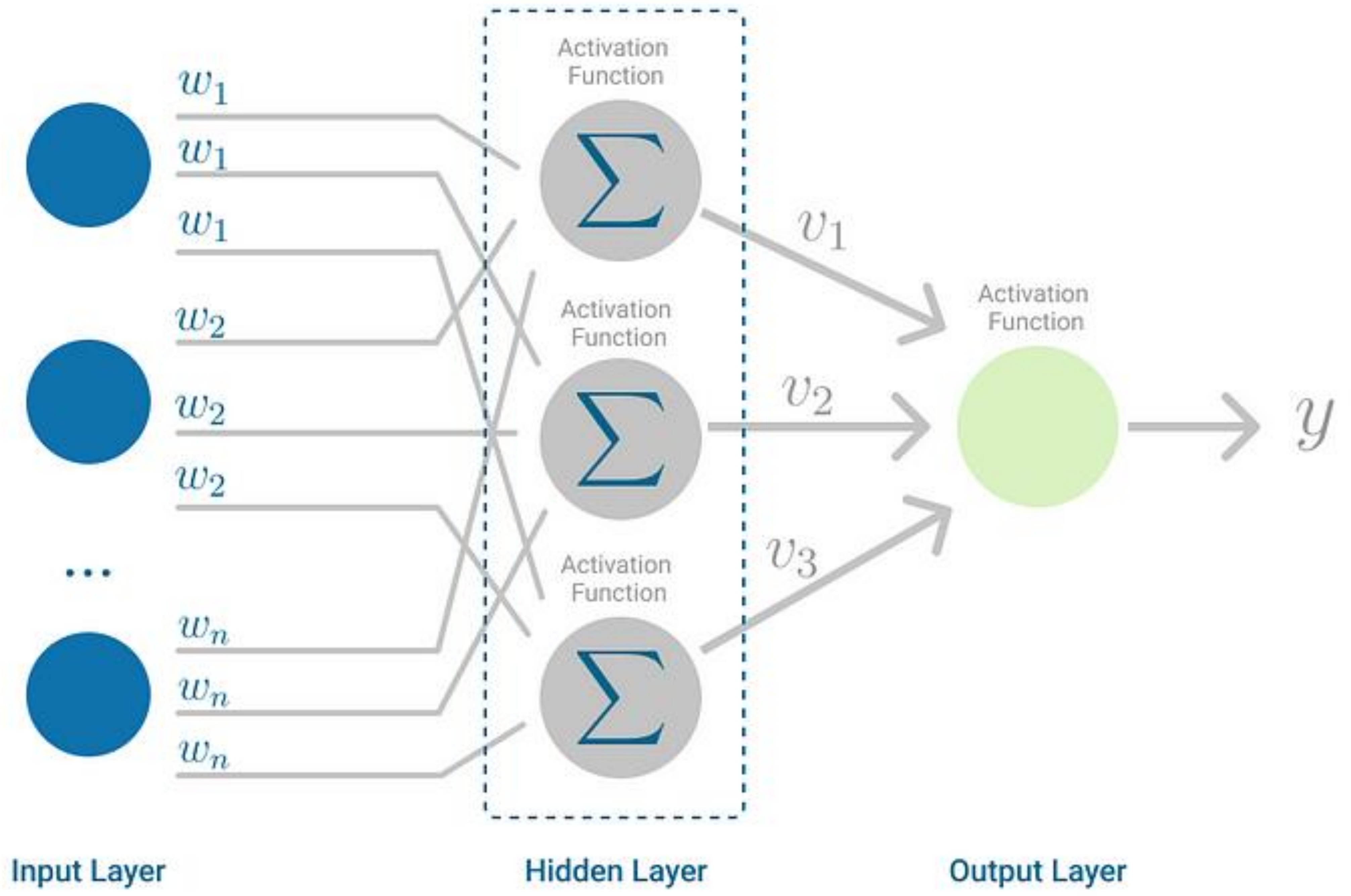
For a single targeted variable, use one node.

N classification problem, ANN uses N nodes in the output layer.

Working of MultiLayer Perceptron Neural Network

1. The input node represents the feature of the dataset.
2. Each input node passes the vector input value to the hidden layer.
3. **In the hidden layer, each edge has some weight multiplied by the input variable. All the production values from the hidden nodes are summed together. To generate the output**
4. **The activation function is used in the hidden layer to identify the active nodes.**
5. The output is passed to the output layer.
6. Calculate the difference between predicted and actual output at the output layer.
7. The model uses back propagation after calculating the predicted output.

MULTILAYER PERCEPTRON



MULTILAYER PERCEPTRON

It is a neural network where the **mapping between inputs and output is non-linear.**

- A Multilayer Perceptron has input and output layers, and one or more **hidden layers** with many neurons stacked together.
- neuron must have an **activation function** that imposes a threshold, like ReLU or sigmoid, neurons in a Multilayer Perceptron can use any **arbitrary activation function.**
- category of feedforward algorithms,because inputs are combined with the initial weights in a weighted sum and subjected to the activation function, just like in the Perceptron.
- But the difference is that each linear combination is propagated to the next layer.
- Each layer is *feeding* the next one with the result of their computation, their internal representation of the data.
- This goes all the way through the hidden layers to the output layer.
- If the algorithm only computed the weighted sums in each neuron, propagated results to the output layer, and stopped there, it wouldn’t be able to *learn* the weights that minimize the cost function.
- If the algorithm only computed one iteration, there would be no actual learning.