SNS COLLEGE OF TECHNOLOGY

19ECT303- ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

PERCEPTRON AND THE TRAINING OF PERCEPTRON

Perceptron is one of the simplest Artificial neural network architectures. It is the simplest type of feedforward neural network, consisting of a single layer of input nodes that are fully connected to a layer of output nodes. It can learn the linearly separable patterns. it uses slightly different types of artificial neurons known as threshold logic units (TLU).

Types of Perceptron

- **Single-Layer Perceptron:** This type of perceptron is limited to learning linearly separable patterns. effective for tasks where the data can be divided into distinct categories through a straight line.
- **Multilayer Perceptron**: Multilayer perceptrons possess enhanced processing capabilities as they consist of two or more layers, adept at handling more complex patterns and relationships within the data.

Basic Components of Perceptron

A perceptron, the basic unit of a neural network, comprises essential components that collaborate in information processing.

- **Input Features:** The perceptron takes multiple input features, each input feature represents a characteristic or attribute of the input data.
- **Weights**: Each input feature is associated with a weight, determining the significance of each input feature in influencing the perceptron's output. During training, these weights are adjusted to learn the optimal values.
- **Summation Function**: The perceptron calculates the weighted sum of its inputs using the summation function. The summation function combines the inputs with their respective weights to produce a weighted sum.
- **Activation Function**: The weighted sum is then passed through an activation function. Perceptron uses Heaviside step function functions. which take the summed values as input and compare with the threshold and provide the output as 0 or 1.
- **Output**: The final output of the perceptron, is determined by the activation function's result. For example, in binary classification problems, the output might represent a predicted class (0 or 1).
- **Bias**: A bias term is often included in the perceptron model. The bias allows the model to make adjustments that are independent of the input. It is an additional parameter that is learned during training.

- **Learning Algorithm (Weight Update Rule)**: During training, the perceptron learns by adjusting its weights and bias based on a learning algorithm. A common approach is the perceptron learning algorithm, which updates weights based on the difference between the predicted output and the true output.

These components work together to enable a perceptron to learn and make predictions. While a single perceptron can perform binary classification, more complex tasks require the use of multiple perceptrons organized into layers, forming a neural network.

How does Perceptron work?

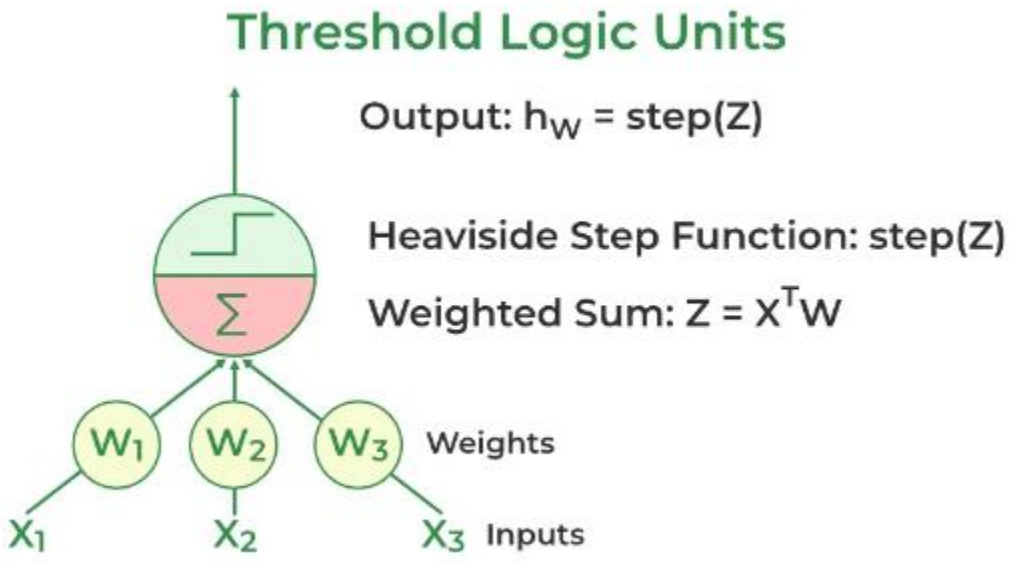
A weight is assigned to each input node of a perceptron, indicating the significance of that input to the output. The perceptron's output is a weighted sum of the inputs that have been run through an activation function to decide whether or not the perceptron will fire. it computes the weighted sum of its inputs as:

$z = w_1x_1 + w_1x_2 + \ldots + w_nx_n = X^TW$

The step function compares this weighted sum to the threshold, which outputs 1 if the input is larger than a threshold value and 0 otherwise, is the activation function that perceptrons utilize the most frequently. The most common step function used in perceptron is the Heaviside step function:

$$h(z) = \begin{cases} 0 & \text{if } z < Threshold \\ 1 & \text{if } z \geq Threshold \end{cases}$$

A perceptron has a single layer of **threshold logic units** with each TLU connected to all inputs.



When all the neurons in a layer are connected to every neuron of the previous layer, it is known as a fully connected layer or dense layer.

The output of the fully connected layer can be:

$$f_{W,b}(X) = h(XW + b)$$

where X is the input W is the weight for each inputs neurons and b is the bias and h is the step function.

During training, The perceptron's weights are adjusted to minimize the difference between the predicted output and the actual output. Usually, supervised learning algorithms like the delta rule or the perceptron learning rule are used for this.

$$w_{i,j} = w_{i,j} + \eta(y_j - \hat{y}_j)x_i$$

Here $w_{i,j}$ is the weight between the ith input and jth output neuron, x_i is the ith input value, and y_j and is the jth actual and predicted value is the learning rate.