DEEP LEARNING

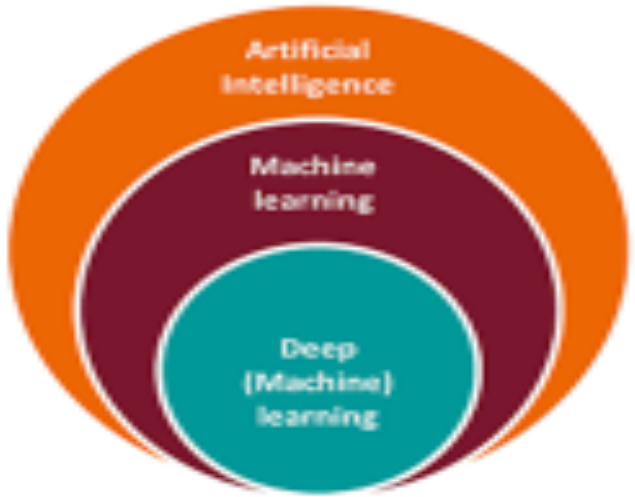
MODULE 1

Deep learning

- Artificial Intelligence - mimic the human behaviour.

- Machine Learning - Branch of artificial intelligence.
  - Subset of AI techniques which uses statistical methods to enable machine to improve performances.

- Deep Learning - is a subset of machine learning which has a set of algorithms to learn to represent the data.



Neuron

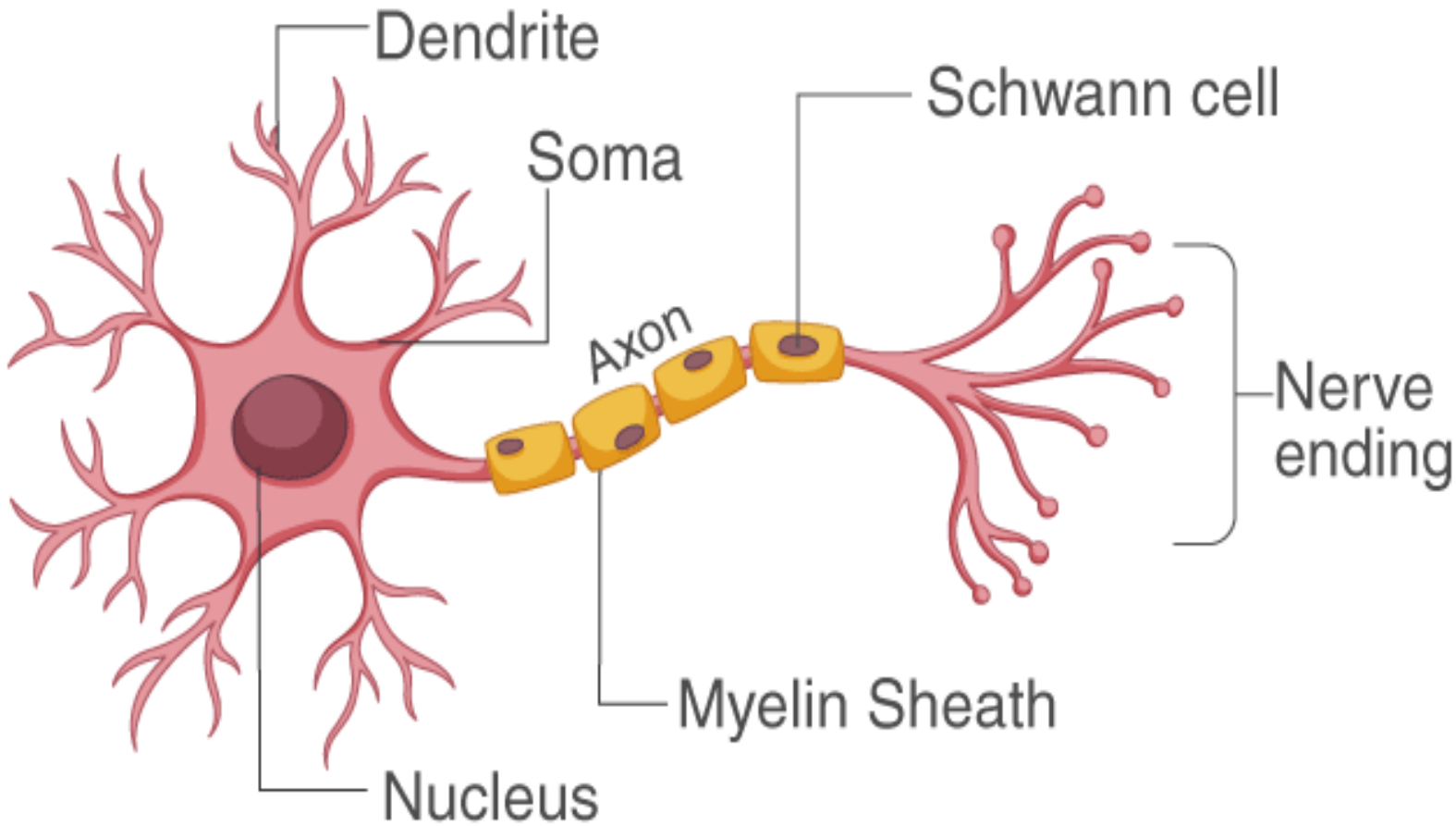
- Neurons are the fundamental unit of the nervous system specialized to transmit information to different parts of the body

What is a Neuron?

- Neurons are the building blocks of the nervous system.
- They receive and transmit signals to different parts of the body.
- This is carried out in both physical and electrical forms.
- There are several different types of neurons that facilitate the transmission of information.

- *The **sensory neurons** carry information from the sensory receptor cells present throughout the body to the brain.*
- *The **motor neurons** transmit information from the brain to the muscles.*
- *The **interneurons** transmit information between different neurons in the body.*

STRUCTURE OF NEURON



Neuron Structure

- **All neurons have three different parts – dendrites, cell body and axon.**

- **Parts of Neuron**
- Following are the different parts of a neuron:

- Dendrites
- These are branch-like structures that receive messages from other neurons and allow the transmission of messages to the cell body.

- Cell Body
- Each neuron has a cell body with a nucleus, Golgi body, endoplasmic reticulum, mitochondria and other components.

- Axon
- Axon is a tube-like structure that carries electrical impulse from the cell body to the axon terminals that pass the impulse to another neuron.

- **Synapse**
- It is the chemical junction between the terminal of one neuron and the dendrites of another neuron.

Neuron Functions

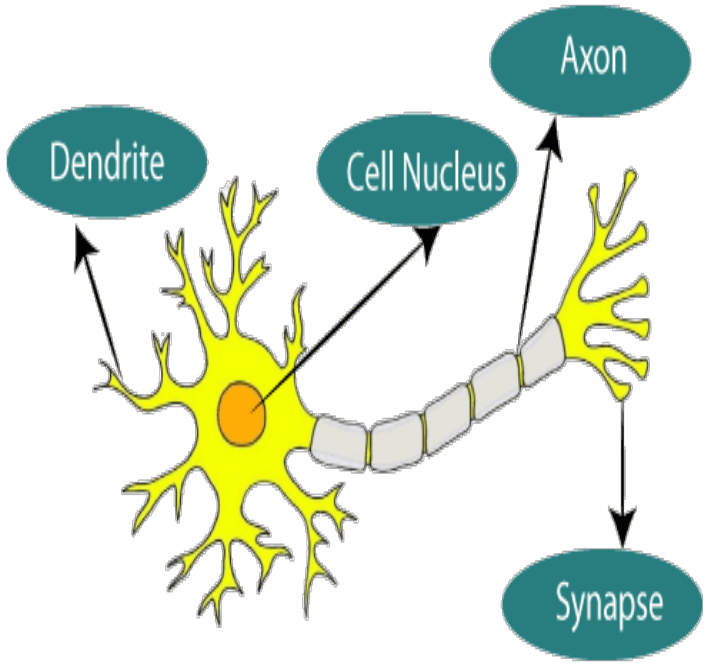
- The important functions of a neuron are:

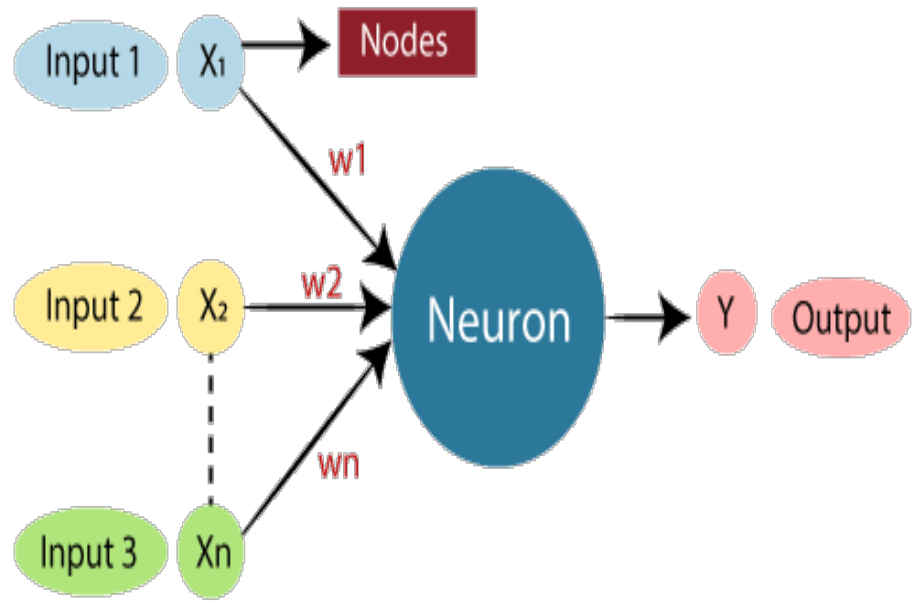
- **Chemical Synapse**
- In chemical synapses, the action potential affects other neurons through a gap present between two neurons known as the synapse. The action potential is carried along the axon to a postsynaptic ending that initiates the release of chemical messengers known as neurotransmitters. These neurotransmitters excite the postsynaptic neurons that generate an action potential of their own.

- **Electrical Synapse**
- When two neurons are connected by a gap junction, it results in an electrical synapse. These gaps include ion channels that help in the direct transmission of a positive electrical signal. These are much faster than chemical synapses.

Artificial Neural Network

- An Artificial neural network is usually a computational network based on biological neural networks that construct the structure of the human brain.
- Similar to a human brain, it has neurons interconnected to each other, artificial neural networks also have neurons that are linked to each other in various layers of the networks.
- These neurons are known as nodes.



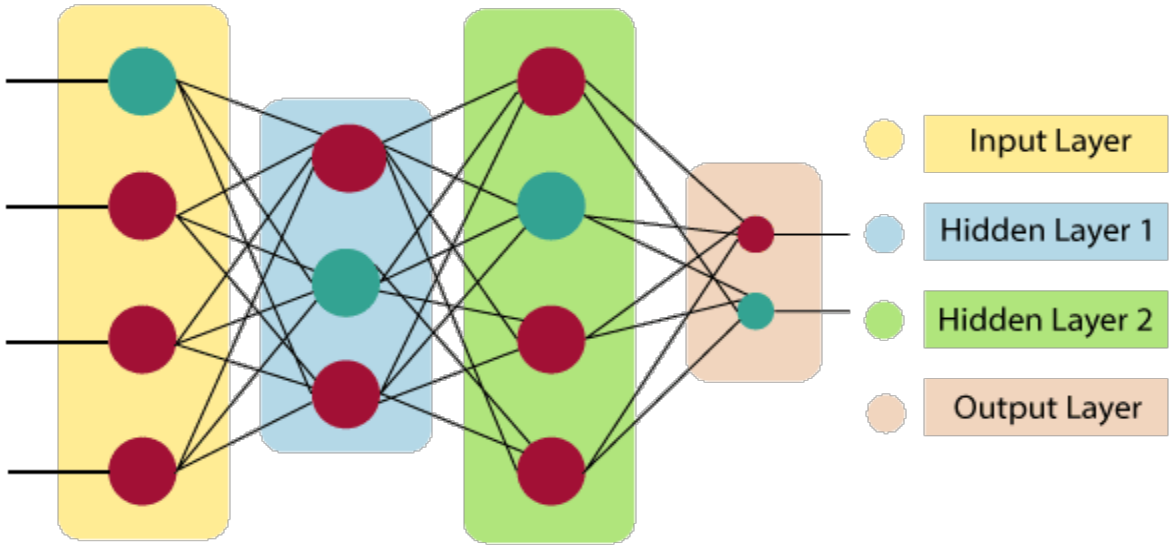


Dendrites from Biological Neural Network represent inputs in Artificial Neural Networks, cell nucleus represents Nodes, synapse represents Weights, and Axon represents Output.

Biological Neural Network	Artificial Neural Network
Dendrites	Inputs
Cell nucleus	Nodes
Synapse	Weights
Axon	Output

Artificial Neural Network

- Artificial Neural Network primarily consists of three layers:

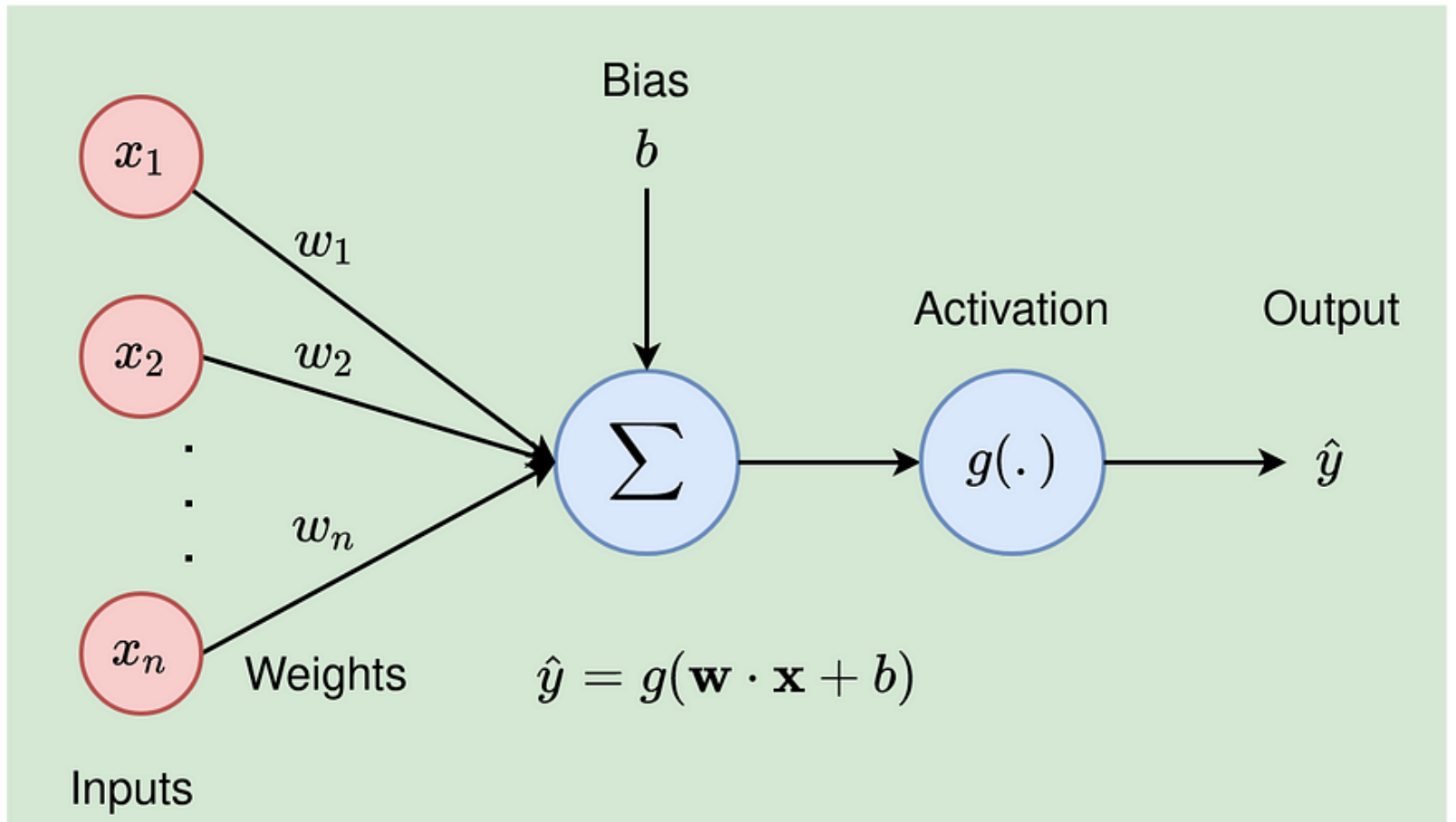


- Input Layer:
- As the name suggests, it accepts inputs in several different formats provided by the programmer.
- Hidden Layer:
- The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.
- Output Layer:
- The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.

- The artificial neural network takes input and computes the weighted sum of the inputs and includes a bias. This computation is represented in the form of a transfer function.

$$\sum_{i=1}^{n} Wi * Xi + b$$

- **What is Artificial Neural Network**
- It determines weighted total is passed as an input to an activation function to produce the output. Activation functions choose whether a node should fire or not. Only those who are fired make it to the output layer. There are distinctive activation functions available that can be applied upon the sort of task we are performing.


Bias
b
x_1
w_1
x_2
w_2
w_n
x_n
$\sum$
Activation
$g(.)$
Output
$\hat{y}$
Weights
$\hat{y} = g(\mathbf{w} \cdot \mathbf{x} + b)$
Inputs

Advantages of Artificial Neural Network (ANN)

- **Parallel processing capability:**

- Artificial neural networks have a numerical value that can perform more than one task simultaneously.

- **Storing data on the entire network:**

- Capability to work with incomplete knowledge:
- After ANN training, the information may produce output even with inadequate data. The loss of performance here relies upon the significance of missing data.
- Having a memory distribution:
- For ANN is to be able to adapt, it is important to determine the examples and to encourage the network according to the desired output by demonstrating these examples to the network. produce false output.
- Having fault tolerance:
- Extortion of one or more cells of ANN does not prohibit it from generating output, and this feature makes the network fault-tolerance.

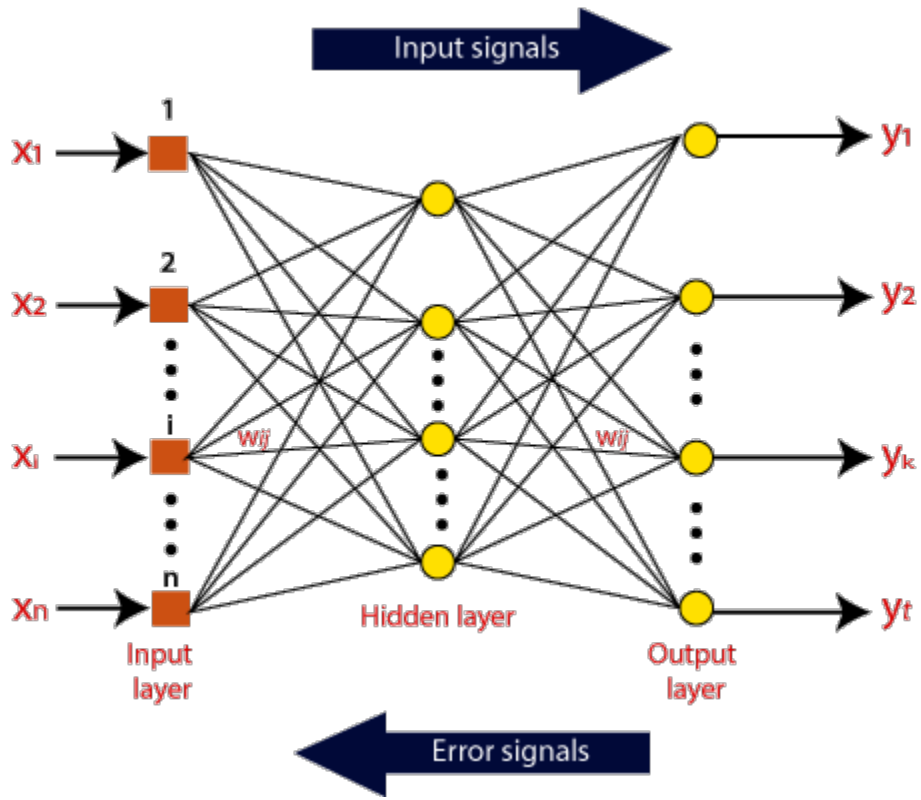
Disadvantages of Artificial Neural Network:

- Assurance of proper network structure:
- There is no particular guideline for determining the structure of artificial neural networks. The appropriate network structure is accomplished through experience, trial, and error.
- Unrecognized behavior of the network:
- It is the most significant issue of ANN. When ANN produces a testing solution, it does not provide insight concerning why and how. It decreases trust in the network.
- Hardware dependence:
- Artificial neural networks need processors with parallel processing power, as per their structure. Therefore, the realization of the equipment is dependent.

- Difficulty of showing the issue to the network:
- ANNs can work with numerical data. Problems must be converted into numerical values before being introduced to ANN. The presentation mechanism to be resolved here will directly impact the performance of the network. It relies on the user's abilities.
- The duration of the network is unknown:
- The network is reduced to a specific value of the error, and this value does not give us optimum results.

How do artificial neural networks work?

- Artificial Neural Network can be best represented as a weighted directed graph, where the artificial neurons form the nodes.
- The association between the neurons outputs and neuron inputs can be viewed as the directed edges with weights.
- The Artificial Neural Network receives the input signal from the external source in the form of a pattern and image in the form of a vector.
- These inputs are then mathematically assigned by the notations x(n) for every n number of inputs.


Input signals
1
2
i
n
x1
x2
xi
xn
wij
wij
y1
y2
yk
yt
Input layer
Hidden layer
Output layer
Error signals

- Afterward, each of the input is multiplied by its corresponding weights (these weights are the details utilized by the artificial neural networks to solve a specific problem).
- In general terms, these weights normally represent the strength of the interconnection between neurons inside the artificial neural network.
- All the weighted inputs are summarized inside the computing unit.
- If the weighted sum is equal to zero, then bias is added to make the output non-zero or something else to scale up to the system's response.
- Bias has the same input, and weight equals to 1.
- Here the total of weighted inputs can be in the range of 0 to positive infinity.

- The activation function refers to the set of transfer functions used to achieve the desired output.
- There is a different kind of the activation function, but primarily either linear or non-linear sets of functions.
- Some of the commonly used sets of activation functions are the Binary, linear, and Tan hyperbolic sigmoidal activation functions.

- Binary:
- In binary activation function, the output is either a one or a 0. Here, to accomplish this, there is a threshold value set up. If the net weighted input of neurons is more than 1, then the final output of the activation function is returned as one or else the output is returned as 0.
- Sigmoidal Hyperbolic:
- The Sigmoidal Hyperbola function is generally seen as an "S" shaped curve. Here the tan hyperbolic function is used to approximate output from the actual net input. The function is defined as:
- F(x) = (1/1 + exp(-x))

Types of Artificial Neural Network:

- Feedforward ANN

Its flow is uni-directional, meaning that the information in the model flows in only one direction—forward—from the input nodes, through the hidden nodes (if any) and to the output nodes, without any cycles or loops.

- Feedback ANN

In this type of ANN, the output returns into the network to accomplish the best-evolved results internally

. The feedback networks feed information back into itself and are well suited to solve optimization issues.

The Internal system error corrections utilize feedback ANN

PERCEPTRON

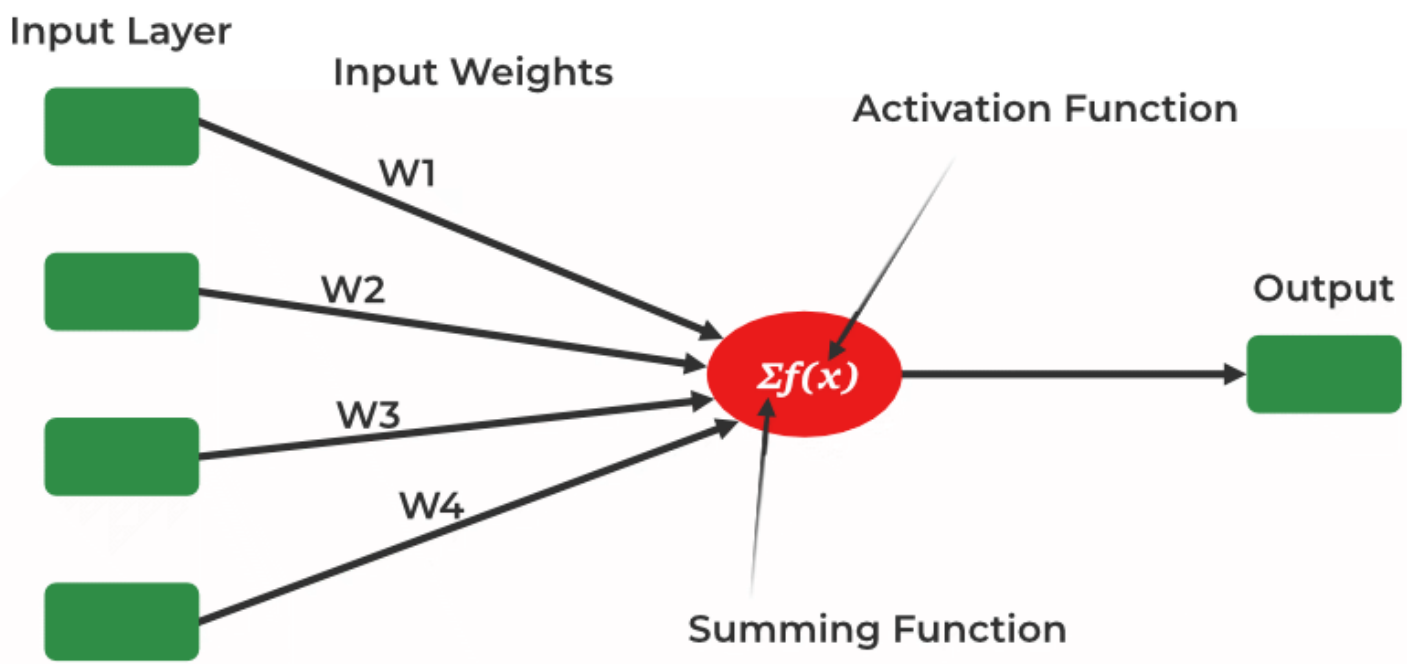
- It is one of the oldest and first introduced neural networks.
- It was proposed by **Frank Rosenblatt** in **1958**.
- Perceptron is also known as an artificial neural network.
- Perceptron is mainly used to compute the logical gate like **AND, OR, and NOR** which has binary input and binary output.
- Perceptron is a building block of an Artificial Neural Network.
- Perceptron is a linear Machine Learning algorithm used for supervised learning for various binary classifiers. This algorithm enables neurons to learn elements and processes them one by one during preparation.

- **The main functionality of the perceptron is:-**
- Takes input from the input layer
- Weight them up and sum it up.
- Pass the sum to the function to produce the output.

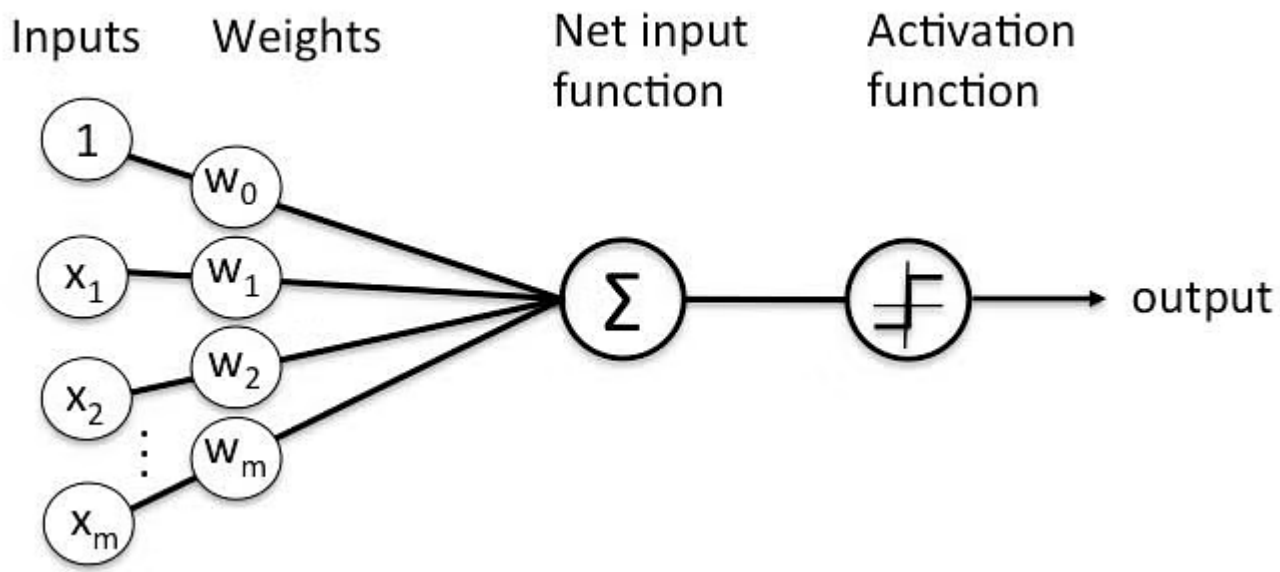
What is Binary classifier in Machine Learning?

- In Machine Learning, binary classifiers are defined as the function that helps in deciding whether input data can be represented as vectors of numbers and belongs to some specific class.

- Binary classifiers can be considered as linear classifiers. In simple words, we can understand it as a classification algorithm that can predict linear predictor function in terms of weight and feature vectors.


Input Layer
Input Weights
Activation Function
W1
W2
W3
W4
Σf(x)
Output
Summing Function

- Activation functions can be anything like **sigmoid, tanh, relu**

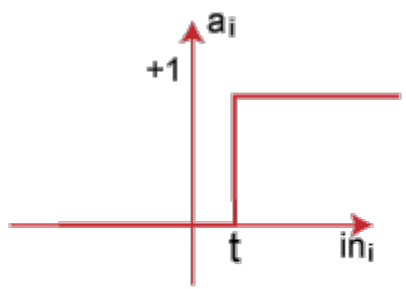


- **Input Nodes or Input Layer:**
- This is the primary component of Perceptron which accepts the initial data into the system for further processing. Each input node contains a real numerical value.
- Wight and Bias:
- Weight parameter represents the strength of the connection between units.
- Weight is directly proportional to the strength of the associated input neuron in deciding the output.
- Bias can be considered as the line of intercept in a linear equation.

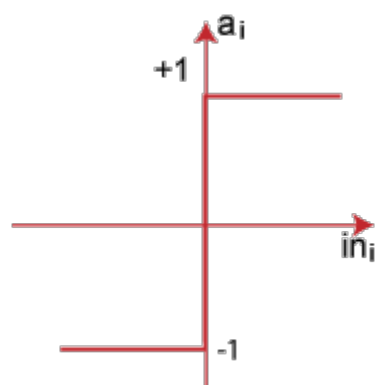
- **Activation Function:**
- These are the final and important components that help to determine whether the neuron will fire or not. Activation Function can be considered primarily as a step function.

- Types of Activation functions:

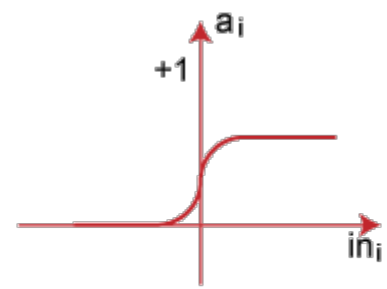
- Sign function
- Step function, and
- Sigmoid function



Step Function



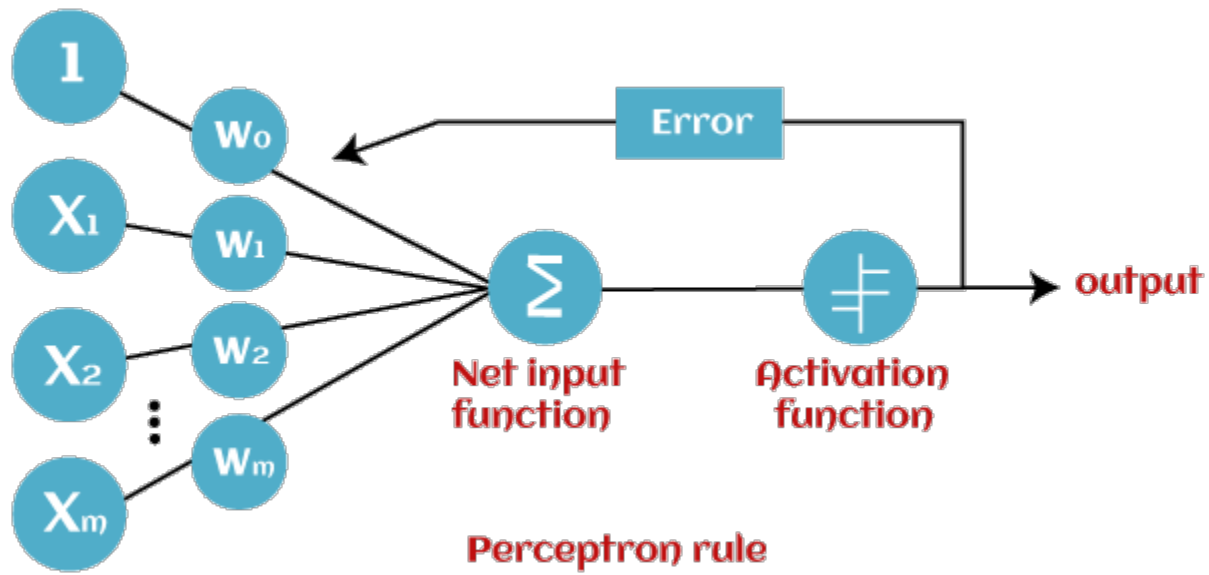
Sign Function



Sigmoid Function

How does Perceptron work?

- In Machine Learning, Perceptron is considered as a single-layer neural network that consists of four main parameters named input values (Input nodes), weights and Bias, net sum, and an activation function. The perceptron model begins with the multiplication of all input values and their weights, then adds these values together to create the weighted sum.
- Then this weighted sum is applied to the activation function 'f' to obtain the desired output. This activation function is also known as the step function and is represented by 'f'.


1
X1
X2
Xm
W0
W1
W2
Wm
Error
Σ
output
Net input function
Activation function
Perceptron rule

Perceptron model works in two important steps as follows:

- Step-1
- In the first step first, multiply all input values with corresponding weight values and then add them to determine the weighted sum. Mathematically, we can calculate the weighted sum as follows:
- $\sum wi*xi = x1*w1 + x2*w2 + \ldots wn*xn$
- Add a special term called bias 'b' to this weighted sum to improve the model's performance.
- $\sum wi*xi + b$
- Step-2
- In the second step, an activation function is applied with the above-mentioned weighted sum, which gives us output either in binary form or a continuous value as follows:
- $Y = f(\sum wi*xi + b)$

Types of Perceptron Models

- Based on the layers, Perceptron models are divided into two types. These are as follows:

- Single-layer Perceptron Model
- Multi-layer Perceptron model

Single Layer Perceptron Model:

- This is one of the easiest Artificial neural networks (ANN) types.
- A single-layered perceptron model consists feed-forward network and also includes a threshold transfer function inside the model.
- The main objective of the single-layer perceptron model is to analyze the linearly separable objects with binary outcomes.
- In a single layer perceptron model, its algorithms do not contain recorded data, so it begins with inconstantly allocated input for weight parameters.
- Further, it sums up all inputs (weight).
- After adding all inputs, if the total sum of all inputs is more than a pre-determined value, the model gets activated and shows the output value as +1.
- If the outcome is same as pre-determined or threshold value, then the performance of this model is stated as satisfied, and weight demand does not change.
- Hence, to find desired output and minimize errors, some changes should be necessary for the weights input.
- "Single-layer perceptron can learn only linearly separable patterns."

Multi-Layered Perceptron Model:

- Like a single-layer perceptron model, a multi-layer perceptron model also has the same model structure but has a greater number of hidden layers.
- The multi-layer perceptron model is also known as the Backpropagation algorithm, which executes in two stages as follows:
- Forward Stage: Activation functions start from the input layer in the forward stage and terminate on the output layer.
- Backward Stage: In the backward stage, weight and bias values are modified as per the model's requirement. In this stage, the error between actual output and demanded originated backward on the output layer and ended on the input layer.
- Hence, a multi-layered perceptron model has considered as multiple artificial neural networks having various layers in which activation function does not remain linear, similar to a single layer perceptron model. Instead of linear, activation function can be executed as sigmoid, TanH, ReLU, etc., for deployment.
- A multi-layer perceptron model has greater processing power and can process linear and non-linear patterns. Further, it can also implement logic gates such as AND, OR, XOR, NAND, NOT, XNOR, NOR.

Advantages of Multi-Layer Perceptron:

- Advantages of Multi-Layer Perceptron:

- A multi-layered perceptron model can be used to solve complex non-linear problems.
- It works well with both small and large input data.
- It helps us to obtain quick predictions after the training.
- It helps to obtain the same accuracy ratio with large as well as small data.

IMPLEMENTATION OF SINGLE-LAYER PERCEPTRON

- Let us now implement a single-layer perceptron using the “MNIST” dataset using the TensorFlow library.

- **Step1: Import necessary libraries**

- Numpy – Numpy arrays are very fast and can perform large computations in a very short time.
- Matplotlib – This library is used to draw visualizations.
- TensorFlow – This is an open-source library that is used for Machine Learning and Artificial intelligence and provides a range of functions to achieve complex functionalities with single lines of code.

- import numpy as np
- import tensorflow as tf
- from tensorflow import keras
- import matplotlib.pyplot as plt
- %matplotlib inline

- Step 2: Now load the dataset using “Keras” from the imported version of tensor flow.

- (x_train, y_train),\
- (x_test, y_test) = keras.datasets.mnist.load_data()

- Step 3: Now display the shape and image of the single image in the dataset. The image size contains a 28*28 matrix and length of the training set is 60,000 and the testing set is 10,000.

- len(x_train)
- len(x_test)
- x_train[0].shape
- plt.matshow(x_train[0])

- Step 4: Now normalize the dataset in order to compute the calculations in a fast and accurate manner.

- # Normalizing the dataset
- x_train = x_train/255
- x_test = x_test/255
-
- # Flatting the dataset in order
- # to compute for model building
- x_train_flatten = x_train.reshape(len(x_train), 28*28)
- x_test_flatten = x_test.reshape(len(x_test), 28*28)

Step 5: Building a neural network with single-layer perception. Here we can observe as the model is a single-layer perceptron that only contains one input layer and one output layer there is no presence of the hidden layers.

```
model = keras.Sequential([
    keras.layers.Dense(10, input_shape=(784,),
                       activation='sigmoid')
])
model.compile(
    optimizer='adam',
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy'])

model.fit(x_train_flatten, y_train, epochs=5)
```

- Step 6: Output the accuracy of the model on the testing data.

- model.evaluate(x_test_flatten, y_test)

Types of Neural Network Architectures

- **Standard neural networks**
- • Perceptron - A neural network that applies a mathematical operation to an input value, providing an output variable.
- • Feed-Forward Networks - A multi-layered neural network where the information moves from left to right, or in other words, in a forward direction. The input values pass through a series of hidden layers on their way to the output layer.
- • Residual Networks (ResNet) - A deep feed-forward network with hundreds of layers.

- **Recurrent neural networks**
- Recurrent neural networks (RNNs) remember previously learned predictions to help make future predictions with accuracy.
- Long short term memory network (LSTM) - LSTM adds extra structures, or gates, to an RNN to improve memory capabilities.
- Echo state network (ESN) - A type of RNN hidden layers that are sparsely connected.

- **Convolutional neural networks**
- Convolutional neural networks (CNNs) are a type of feed-forward network that are used for image analysis and language processing. There are hidden convolutional layers that form ConvNets and detect patterns. CNNs use features such as edges, shapes, and textures to detect patterns. Examples of CNNs include:
- AlexNet - Contains multiple convolutional layers designed for image recognition.
- Visual geometry group (VGG) - VGG is similar to AlexNet, but has more layers of narrow convolutions.
- Capsule networks - Contain nested capsules (groups of neurons) to create a more powerful CNN.

- **Generative adversarial networks**
- Generative adversarial networks (GAN) are a type of unsupervised learning where data is generated from patterns that were discovered from the input data. GANs have two main parts that compete against one another:
- • Generator - creates synthetic data from the learning phase of the model. It will take random datasets and generate a transformed image.
- • Discriminator - decides whether or not the images produced are fake or genuine.
- GANs are used to help predict what the next frame in a video might be, text to image generation, or image to image translation.

- **Transformer neural networks**
- Unlike RNNs, transformer neural networks do not have a concept of timestamps. This enables them to pass through multiple inputs at once, making them a more efficient way to process data.