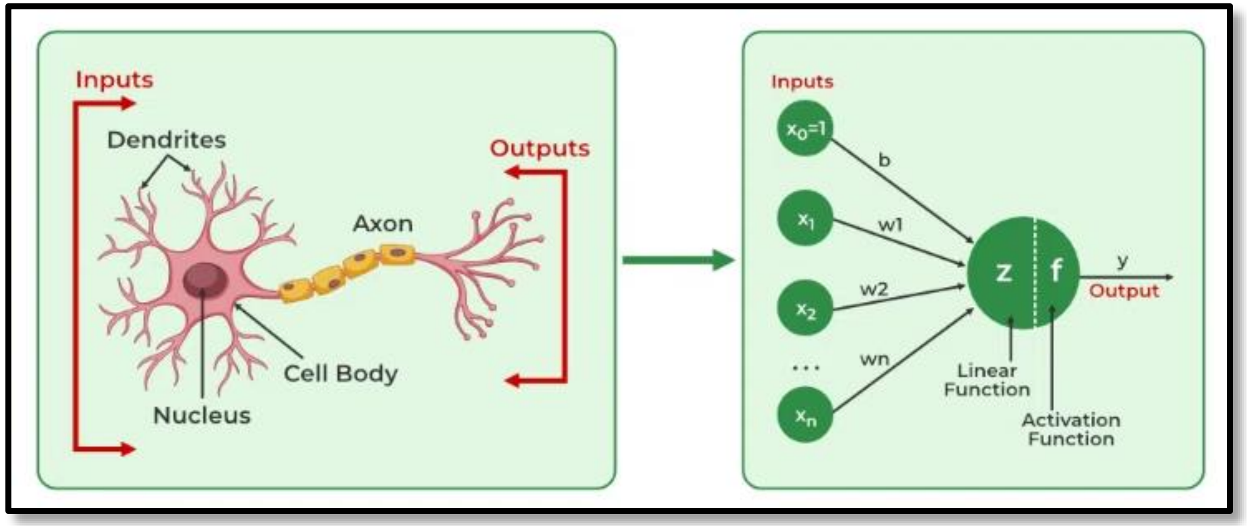
What is an artificial neural network?

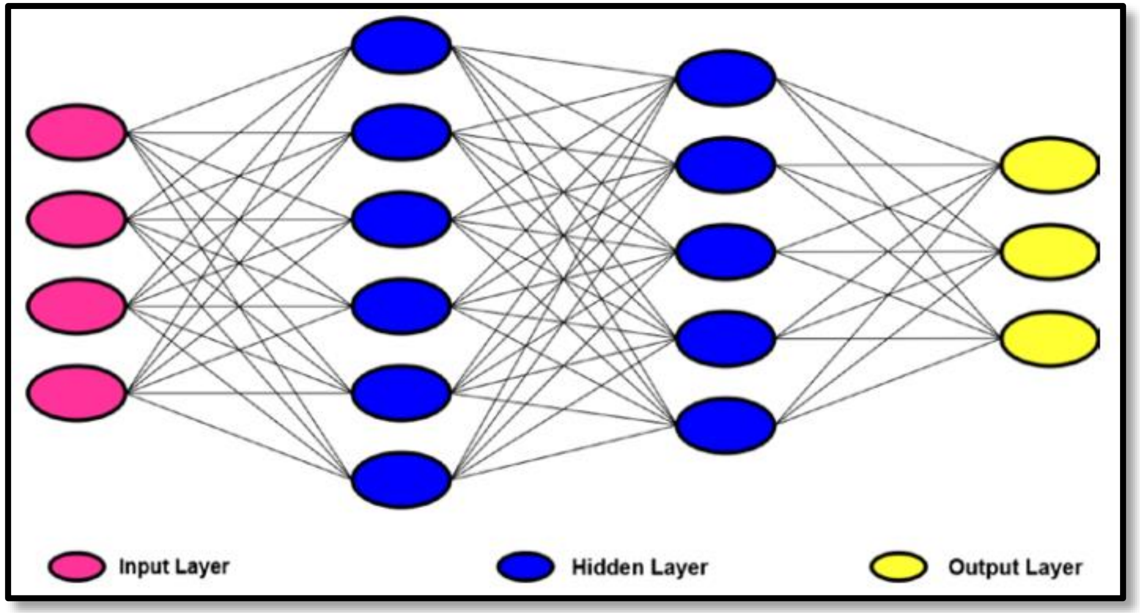
An artificial neural network (ANN) is a computational model inspired by the way biological neural networks in the human brain operate. It consists of interconnected nodes, or artificial neurons, organized in layers, which process information by responding to external inputs.
Each neuron in the network receives inputs, applies a weighted sum, and passes the result through an activation function to determine its output.



The connections between neurons have associated weights that adjust during the learning process, allowing the network to learn from data.

Different layers in artificial neural networks

Input Layer: The first layer in ANNs that receives the raw input data and serves as the entry point for information into the network
Hidden Layers: Intermediate layers between the input and the output layer which extract features and learn complex representations from the input data
Output Layer: The final layer in the neural networks that produces the final output or prediction



What is a Perceptron?

A perceptron is a fundamental component of artificial neural networks and serves as a simple model of a biological neuron.

The perceptron operates by taking multiple input values, applying weights to these inputs, and producing a single binary output based on a decision rule.

The main components of a perceptron include the following:

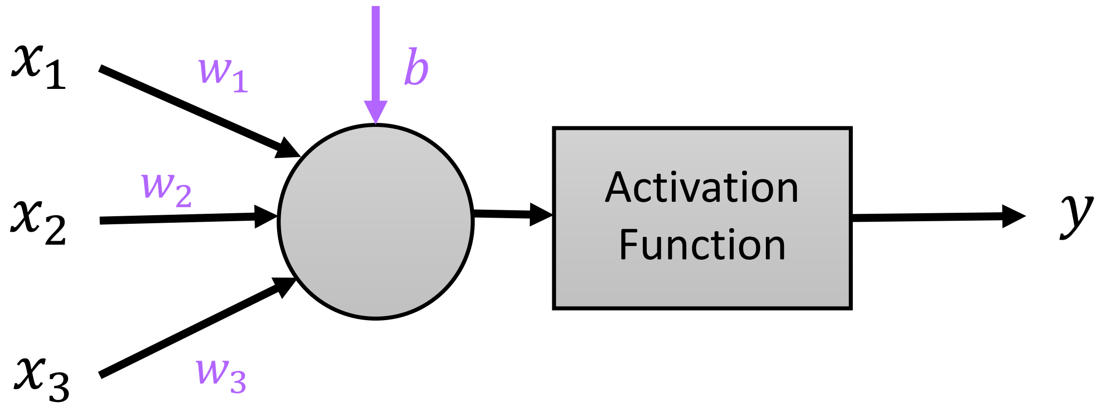
Inputs: The perceptron receives one or more input values, which represent features from the dataset.

Weights: Each input is associated with a weight that signifies its importance in the decision-making process. The weights are adjusted during the training phase to optimize the perceptron's performance.

Bias: A bias term is added to the weighted sum of inputs, allowing the model to shift the activation function, which enhances its flexibility in classification tasks.

What is a Perceptron?

Activation Function: The perceptron uses an activation function to determine the output. If the weighted sum of inputs plus the bias exceeds a certain threshold, the perceptron outputs a 1 (indicating one class); otherwise, it outputs a 0 (indicating the other class).
Output: The final output is a binary value (0 or 1) that indicates the class to which the input data belongs.



What is a Perceptron?

A perceptron computes it's output using the following formula:

$$output = f(\sum_{i=1}^{n} w_i . x_i + b)$$

Where:
n: number of features (inputs)
x_i: the i^{th} input value
w_i: the weight corresponding to each feature
b: the bias term
f: the activation function

A perceptron learns through a process called the perceptron learning rule, which adjusts the weights based on the difference between the predicted output and the actual output. This involves updating the weights and bias to minimize classification errors.
In conclusion, a perceptron is a basic yet essential building block of neural networks, enabling simple binary classification tasks.

Single-Layer Perceptron (SLP)

A single-layer perceptron (SLP) is a type of artificial neural network that consists of a single layer of output nodes connected directly to input features. It represents one of the most elementary forms of neural networks and is primarily used for binary classification tasks.

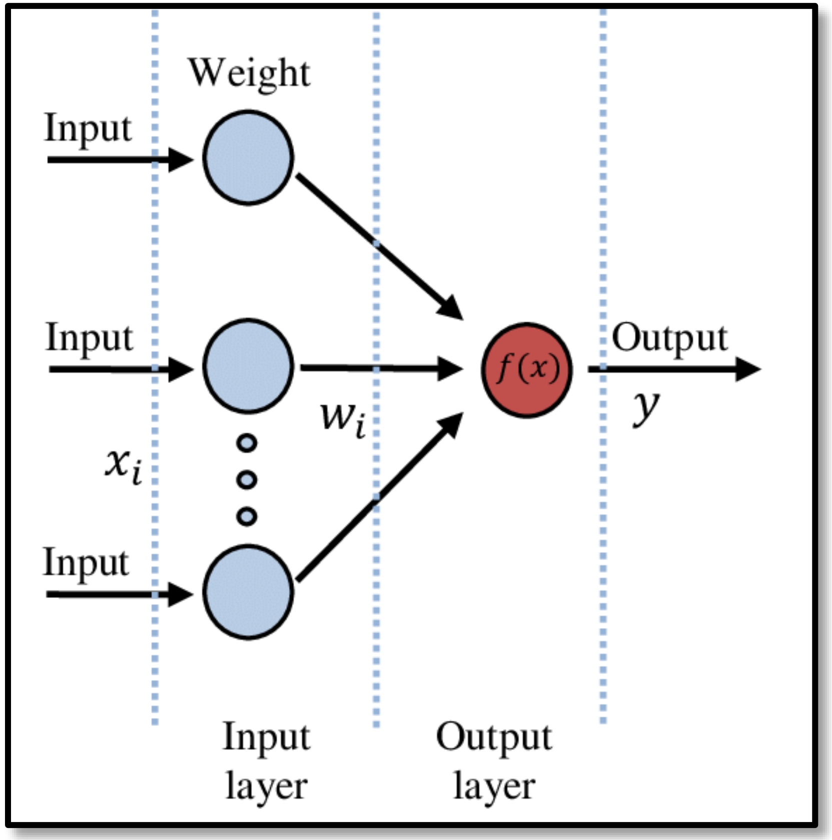
A single-layer perceptron can be defined as a linear classifier that maps input features to a binary output.

It operates by taking multiple inputs,- applying weights to these inputs, summing them up, and passing the result through an activation function. The output is determined based on whether the weighted sum exceeds a certain threshold.

Single-Layer Perceptron (SLP)

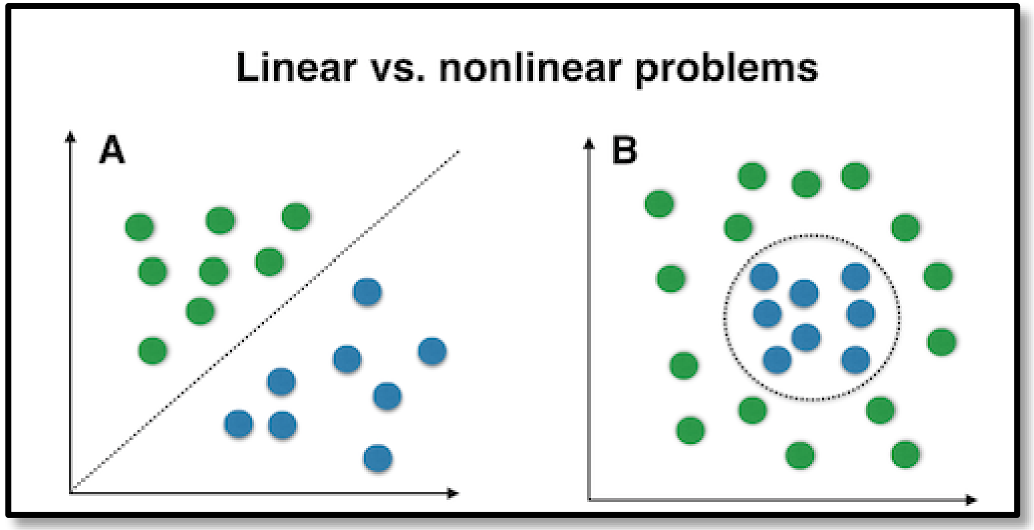
The architecture of a single-layer perceptron includes the input layer, weights, the output node and the activation function.

The activation function commonly utilized in a single-layer perceptron is a step function which determines the output based on the weighted sum and bias. If the result exceeds a threshold, the output is activated; otherwise, it is set to 0.



Single-Layer Perceptron (SLP)

Despite its simplicity, a single-layer perceptron has several limitations.
SLPs can only solve problems that are linearly separable, meaning they can only classify data points that can be separated by a straight line.
Additionally, The single-layer architecture restricts the model's ability to learn complex patterns or relationships in the data. As a result, it cannot capture intricate decision boundaries.



Lastly, The absence of hidden layers means that the perceptron cannot perform feature extraction or transformation, limiting its effectiveness in more complex tasks.

Multi-Layer Perceptron (MLP)

A multi-layer perceptron (MLP) is a type of artificial neural network that consists of multiple layers of neurons, enabling it to learn complex patterns in data. MLPs are particularly notable for their ability to handle non-linearly separable data, which single-layer perceptrons cannot manage effectively.

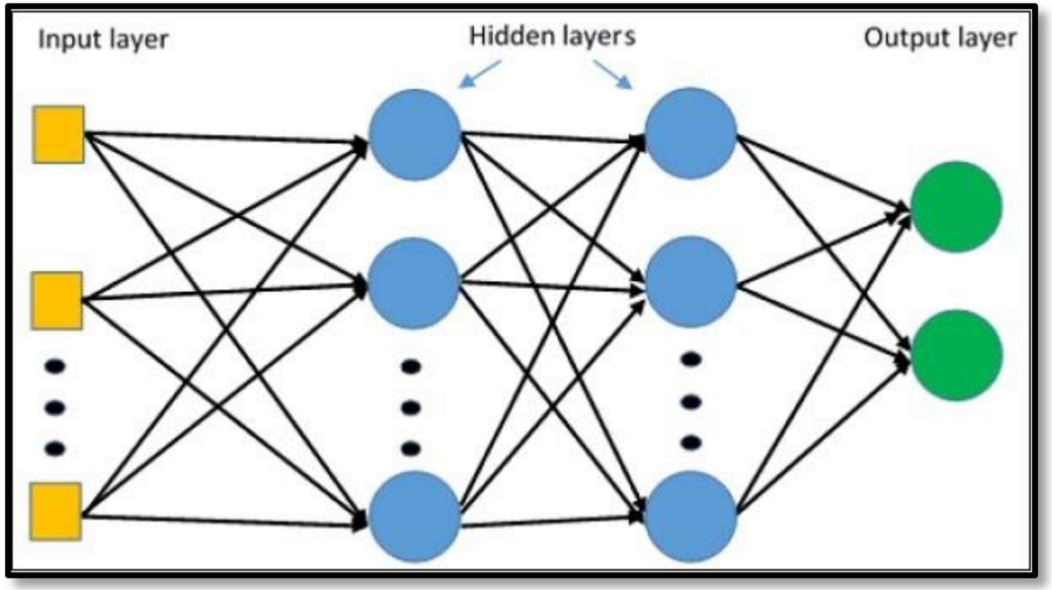
In MLPs, each neuron in the network applies a non-linear activation function (such as sigmoid, hyperbolic tangent, or ReLU) to its input, allowing the- network to learn complex relationships in the data.

MLPs have a Feedforward architecture; this means that information moves in one direction; from the input layer, through one or more hidden layers, to the output layer.

Each neuron in one layer is connected to every neuron in the next layer, which facilitates comprehensive learning across the network.

Multi-Layer Perceptron (MLP)

The architecture of a multi-layer perceptron includes the input layer, hidden layers and the output layer.
The input layer consists of neurons that represent the features of the input data.
One or more hidden layers exist between the input and output layers. Each neuron in a hidden layer receives inputs from all neurons in the previous layer and applies an activation function to produce an output. The number of-



hidden layers and neurons is a design choice that can significantly affect the model's performance.
The output layer produces the output of the network.

Multi-Layer Perceptron (MLP)

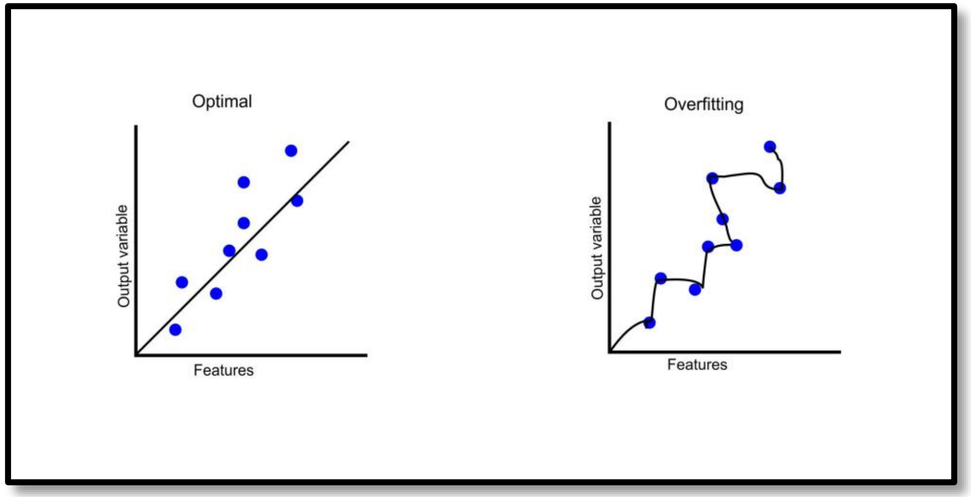
Here are some applications of MLPs:

Image Recognition: MLPs are commonly used in image classification tasks, such as recognizing objects within images and handwritten digit recognition. They can effectively learn features from pixel data to distinguish between different categories.

Speech Recognition: MLPs are applied in converting spoken language into-text, enabling voice-activated systems and virtual assistants. They help in recognizing patterns in audio signals to improve accuracy in transcription.

MLPs also have applications is Natural Language Processing (NLP), forecasting and prediction, medical diagnosis and game development.

Multi-Layer Perceptron (MLP)



Despite their versatility, MLPs have several limitations.
MLPs can be complex to implement and require significant computational resources for training, especially with many layers and neurons.
Additionally, they are prone to overfitting, particularly when trained on small datasets or when the model is overly complex relative to the amount of available data.
Vanishing gradients are an additional-challenge when using MLPs. During training, especially in deep networks, gradients may become very small, leading to slow convergence or failure to learn effectively.
Lastly, the optimization landscape for MLPs can have many local minima, making it challenging to find the global minimum during training.

Logistic Regression Model

Logistic regression is a supervised machine learning algorithm used for binary classification. It models the probability of a binary outcome as a function of predictor variables using the logistic (sigmoid) function.

The logistic regression model assumes a linear combination of the predictor variables and estimates the parameters using maximum likelihood estimation.

While logistic regression and neural networks are both supervised learning-algorithms, they differ in their complexity. Neural networks have a more complex architecture with multiple hidden layers, while logistic regression has a simple structure with only an input layer and an output. Additionally, neural networks use various activation functions in their hidden layers, while logistic regression specifically uses the sigmoid function in the output layer.

Logistic Regression Model

Logistic regression enables the assessment of the relevance or appropriateness of independent variables through the magnitude of their coefficients, while also elucidating the directionality of the relationship or association, whether positive or negative, between the predictor variables and the outcome of interest.

The logistic regression model is suitable for linearly separable data sets. A linearly separable data set refers to a-graph where a straight line separates the two data classes.

We are already familiar with the sigmoid activation function. Here is a brief review of its formula:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Where e is the base of natural logarithms.

Logistic Regression Model

Here is the equation representing logistic regression:

$$y = \frac{e^{(b_0+b_1.x)}}{1 + \ e^{(b_0+b_1.x)}}$$

Where:

x: input value
y: predicted value
b_0: bias term
b_1: coefficient for input x