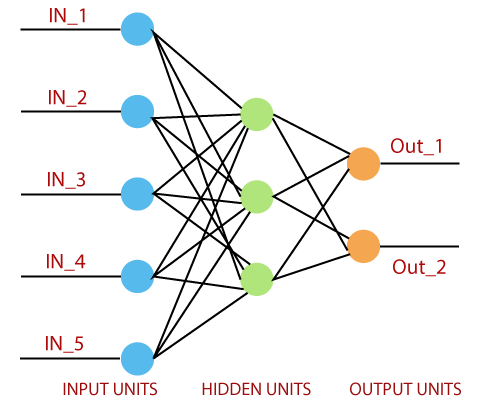
Single Layer Perceptron

The perceptron is a single processing unit of any neural network. **Frank Rosenblatt** first proposed in **1958** is a simple neuron which is used to classify its input into one or two categories. Perceptron is a linear classifier, and is used in supervised learning. It helps to organize the given input data.

A perceptron is a neural network unit that does a precise computation to detect features in the input data. Perceptron is mainly used to classify the data into two parts. Therefore, it is also known as **Linear Binary Classifier**.



Perceptron uses the step function that returns +1 if the weighted sum of its input 0 and -1.

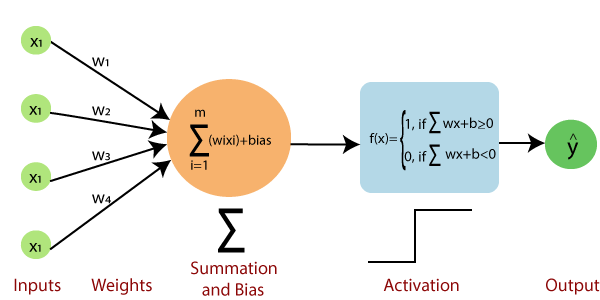
The activation function is used to map the input between the required value like (0, 1) or (-1, 1).

16.3M

300

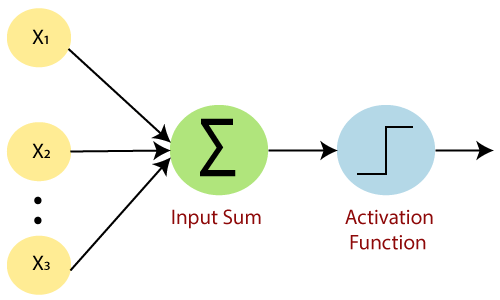
Prime Ministers of India | List of Prime Minister of India (1947-2020)

A regular neural network looks like this:

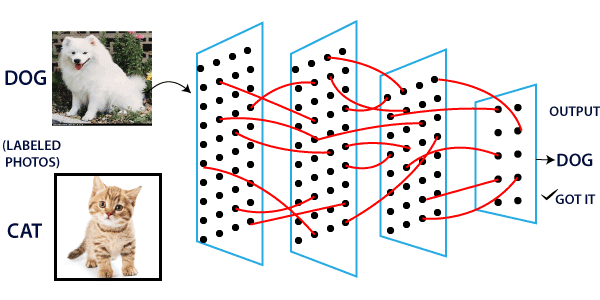


The perceptron consists of 4 parts.

* **Input value or One input layer:** The input layer of the perceptron is made of artificial input neurons and takes the initial data into the system for further processing.
* **Weights and Bias:**  
  **Weight:** It represents the dimension or strength of the connection between units. If the weight to node 1 to node 2 has a higher quantity, then neuron 1 has a more considerable influence on the neuron.  
  **Bias:** It is the same as the intercept added in a linear equation. It is an additional parameter which task is to modify the output along with the weighted sum of the input to the other neuron.
* **Net sum:** It calculates the total sum.
* **Activation Function:** A neuron can be activated or not, is determined by an activation function. The activation function calculates a weighted sum and further adding bias with it to give the result.



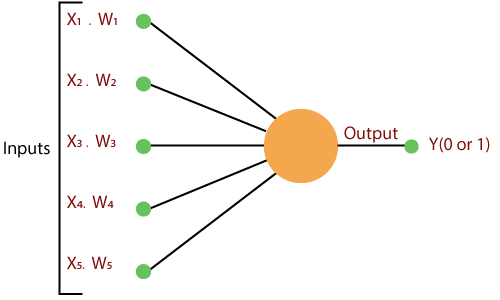
A standard neural network looks like the below diagram.



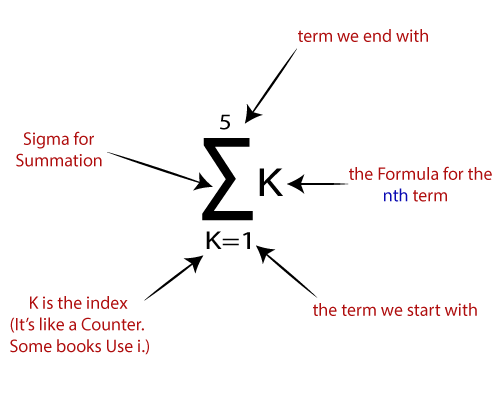
How does it work?

The perceptron works on these simple steps which are given below:

**a.** In the first step, all the inputs x are multiplied with their weights **w**.



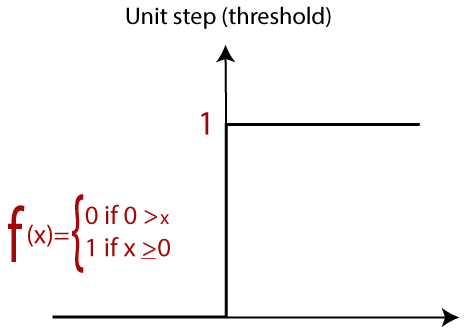
**b.** In this step, add all the increased values and call them the **Weighted sum**.



**c.** In our last step, apply the weighted sum to a correct **Activation Function**.

**For Example:**

A Unit Step Activation Function



There are two types of architecture. These types focus on the functionality of artificial neural networks as follows-

* Single Layer Perceptron
* Multi-Layer Perceptron

Single Layer Perceptron

The single-layer perceptron was the first neural network model, proposed in 1958 by Frank Rosenbluth. It is one of the earliest models for learning. Our goal is to find a linear decision function measured by the weight vector w and the bias parameter b.

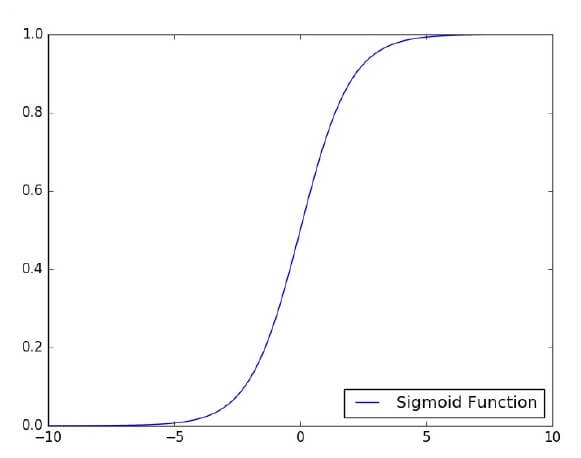
To understand the perceptron layer, it is necessary to comprehend artificial neural networks (ANNs).

The artificial neural network (ANN) is an information processing system, whose mechanism is inspired by the functionality of biological neural circuits. An artificial neural network consists of several processing units that are interconnected.

This is the first proposal when the neural model is built. The content of the neuron's local memory contains a vector of weight.

The single vector perceptron is calculated by calculating the sum of the input vector multiplied by the corresponding element of the vector, with each increasing the amount of the corresponding component of the vector by weight. The value that is displayed in the output is the input of an activation function.

Let us focus on the implementation of a single-layer perceptron for an image classification problem using TensorFlow. The best example of drawing a single-layer perceptron is through the representation of "**logistic regression**."



Now, We have to do the following necessary steps of training logistic regression-

* The weights are initialized with the random values at the origination of each training.
* For each element of the training set, the error is calculated with the difference between the desired output and the actual output. The calculated error is used to adjust the weight.
* The process is repeated until the fault made on the entire training set is less than the specified limit until the maximum number of iterations has been reached.