Day 1 - Introduction to Artificial Neural Network

Tags: Artificial Neuron · Dense Layer · Fully Connected Layer · Perceptron · Step function · TLU

Introduction to Artificial Neural Network

Author: **Chandan Kumar**

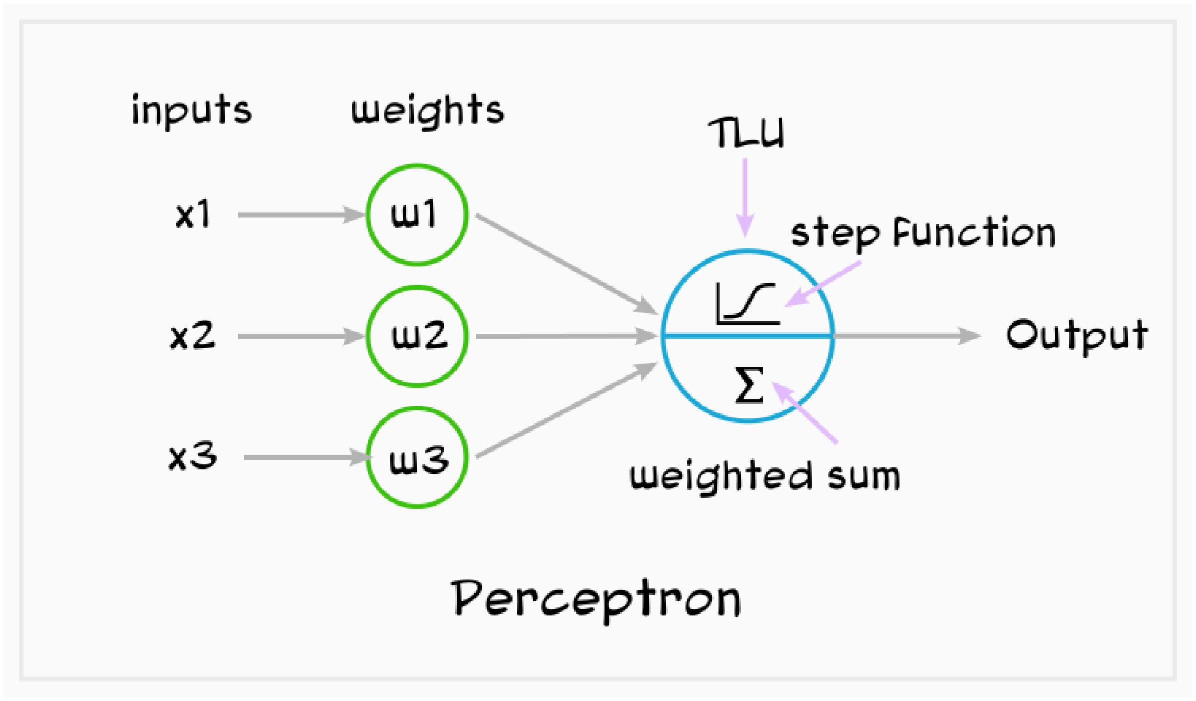
`enchandan.com`

Artificial Neurons

- Introduced in 1943 by the neurophysiologist Warren McCulloch and the mathematician Walter Pitts.
- ANNs looked promising but couldn't perform as expected, so funding went somewhere else.
- It faded during 1980s. Other powerful ML algorithms like Support Vector Machine were invented.
- ANN is back; It is here to stay because of:
    - Huge quantity of data available to train neural networks
    - Increase in computing power
    - Accessible to everyone with cloud platforms
- ANNs play role of input neuron, TLU (decision unit neuron)

Perceptron

- One of the simplest ANN architecture, invented in 1957 by Frank Rosenblatt.
- It is based on Threshold Logic Unit (TLU) (also called Linear Threshold Unit - LTU)
    - TLU is an artificial neuron which computes a weighted sum of its inputs and then applies a step function (to decide if the weights should be activated to go further in network or not)
    - There are other artificial neuron units like TLU such as ReLU (Rectified Linear Unit)
- A Perceptron is composed of a single layer of TLUs.
- **A Simple Perceptron Architecture**



- Perceptron have one or more input and output connections. These connections have weights.
- TLU computes a weighted sum of it's inputs ($z = w_1x_1 + w_2x_2 + ... + w_nx_n = x^Tw$), it's a matrix multiplication (really quick, thanks to GPUs ⚡). Then applies a step function to that sum and outputs the result

$$h_w(x) = step(X^TW)$$

- Output of the TLU (Artificial Neuron) is activated if the activation function (here step function) allows it.

Step function

- Step functions works as a Activation function in a neural network.
- Activation function are decision making units of neural network.
- Some common step functions used in perceptrons:
  - **heaveside step function**:
    - Most common activation function in neural network
    - Produces binary output, so also called binary step function

- **sign step function**

$$\text{heaveside}(z) \begin{cases} 0 \text{ if } z < 0 \\ 1 \text{ if } z >= 0 \end{cases}$$

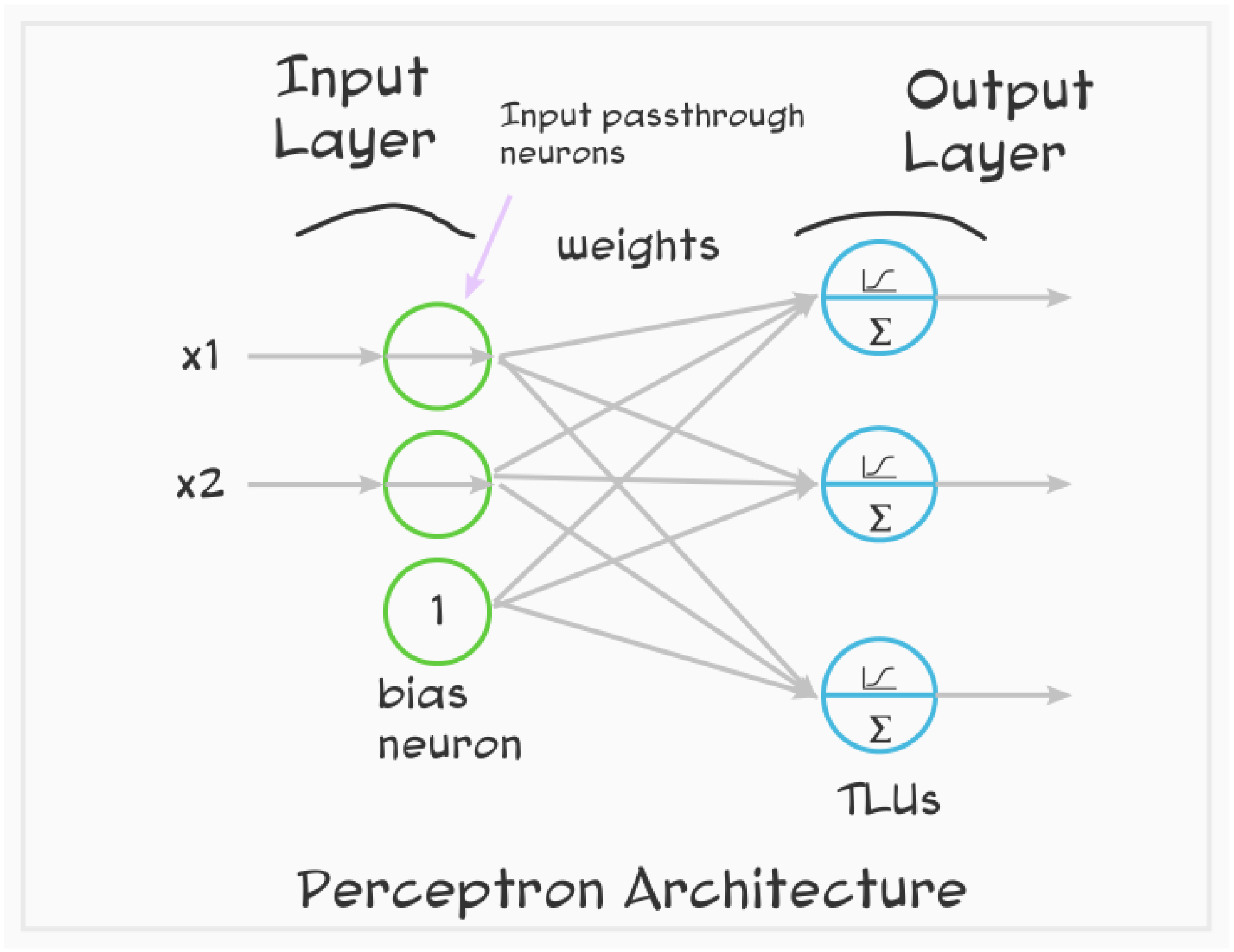
$$\text{sign}(z) \begin{cases} -1 \text{ if } z < 0 \\ 0 \text{ if } z = 0 \\ +1 \text{ if } z > 0 \end{cases}$$

Training Perceptrons

- Training a TLU means finding the right values for w1, w2, w3...
- Perceptrons are trained using a rule which considers error made by the network while prediction.
  Reinforces connection weights which contributed to the correct prediction.

- When all the neurons in a layer is connected to all the neurons in the previous layer, it's called Fully connected layer (or Dense layer).

Another Perceptron Architecture



A perceptron architecture with 2 input neurons, 1 bias neuron and 3 output neurons

Input layer

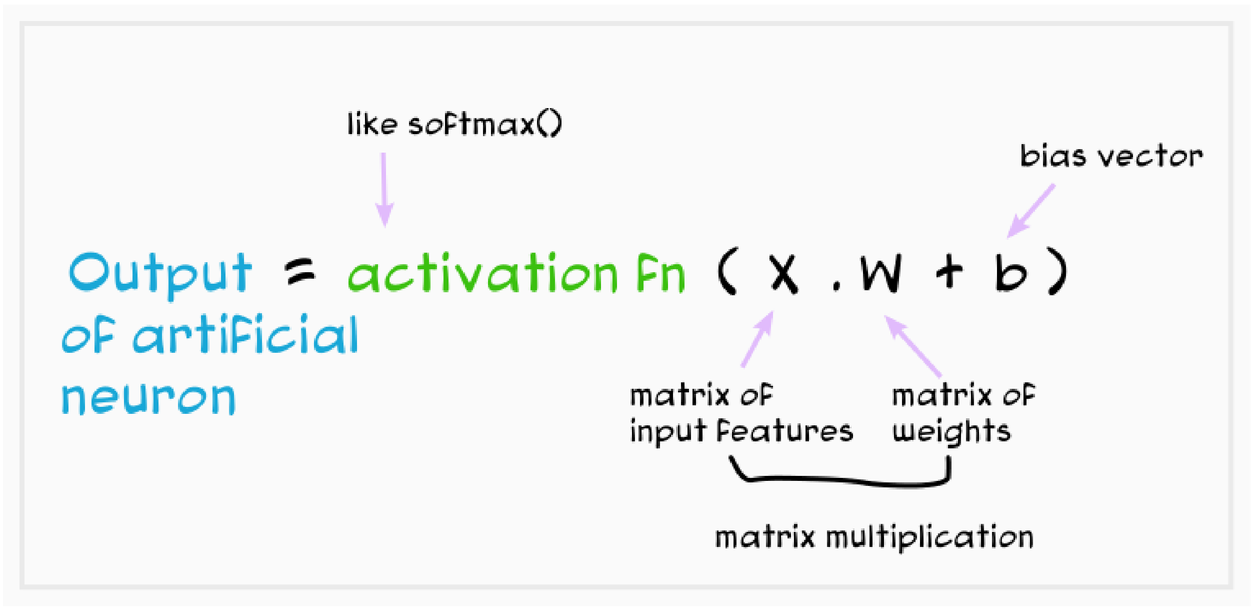
- Input of a perceptron go through a special pass-through neuron called Input neurons.
- They output whatever values they are fed.
- All input neurons form input layer.

Bias neuron

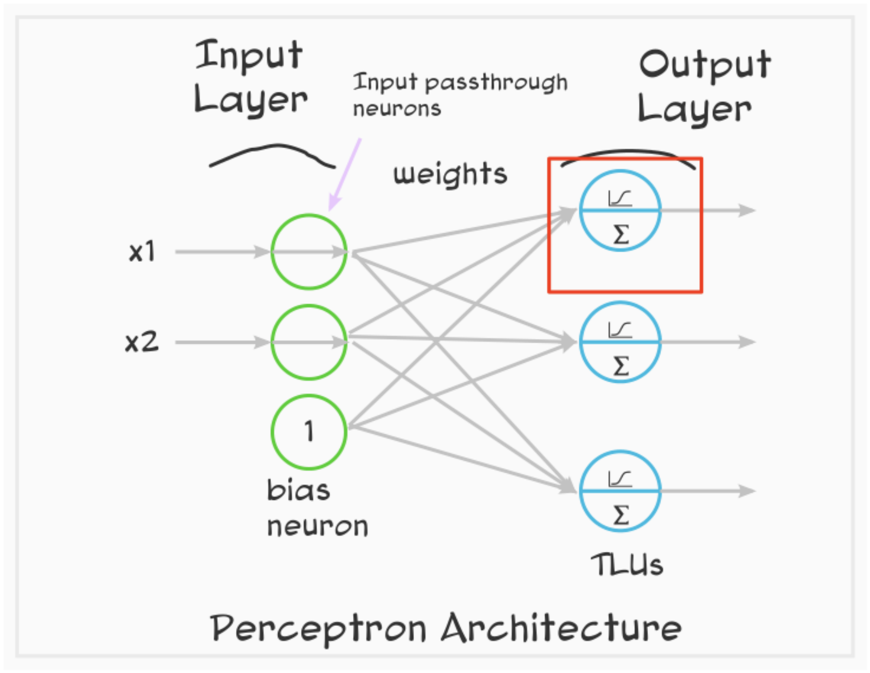
- It is a special neuron added to each layer of the neural network and it always outputs 1.

- Incoming weights increases the steepness of the activation function. This means weight decide how fast the activation function will trigger whereas bias is used to delay the triggering of the activation function.
- It can be thought as similar to the constant **c** available in linear function $y = mx + c$.

Output of an Artificial neuron



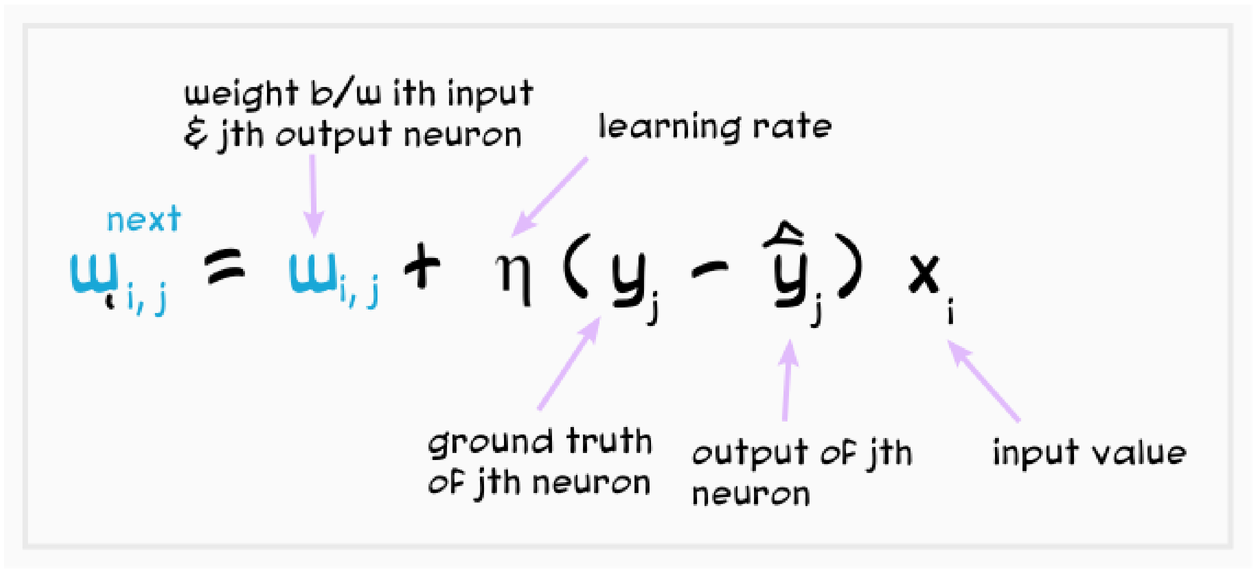
Output of a single artificial neuron, highlighted 'Red'



- Perceptron learning algorithm reinforces weight connections that helps reduce the error (loss function).

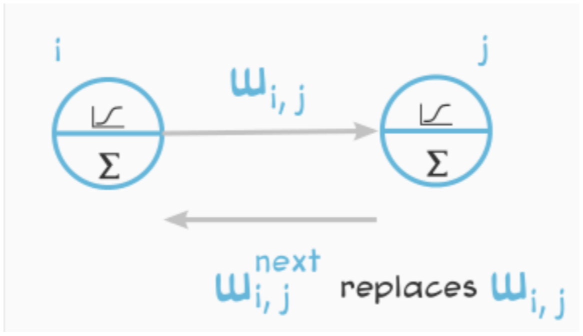
Perceptron learning rule (weight update)

- Perceptron is fed one training instance at a time and for each instance it makes its predictions.
- For every output neuron that produced the wrong prediction, It reinforces the connection weights from the inputs that would have contributed to the correct prediction.


weight b/w ith input & jth output neuron
learning rate
next
$w_{i,j}^{next} = w_{i,j} + \eta(y_j - \hat{y}_j) x_i$
ground truth of jth neuron
output of jth neuron
input value

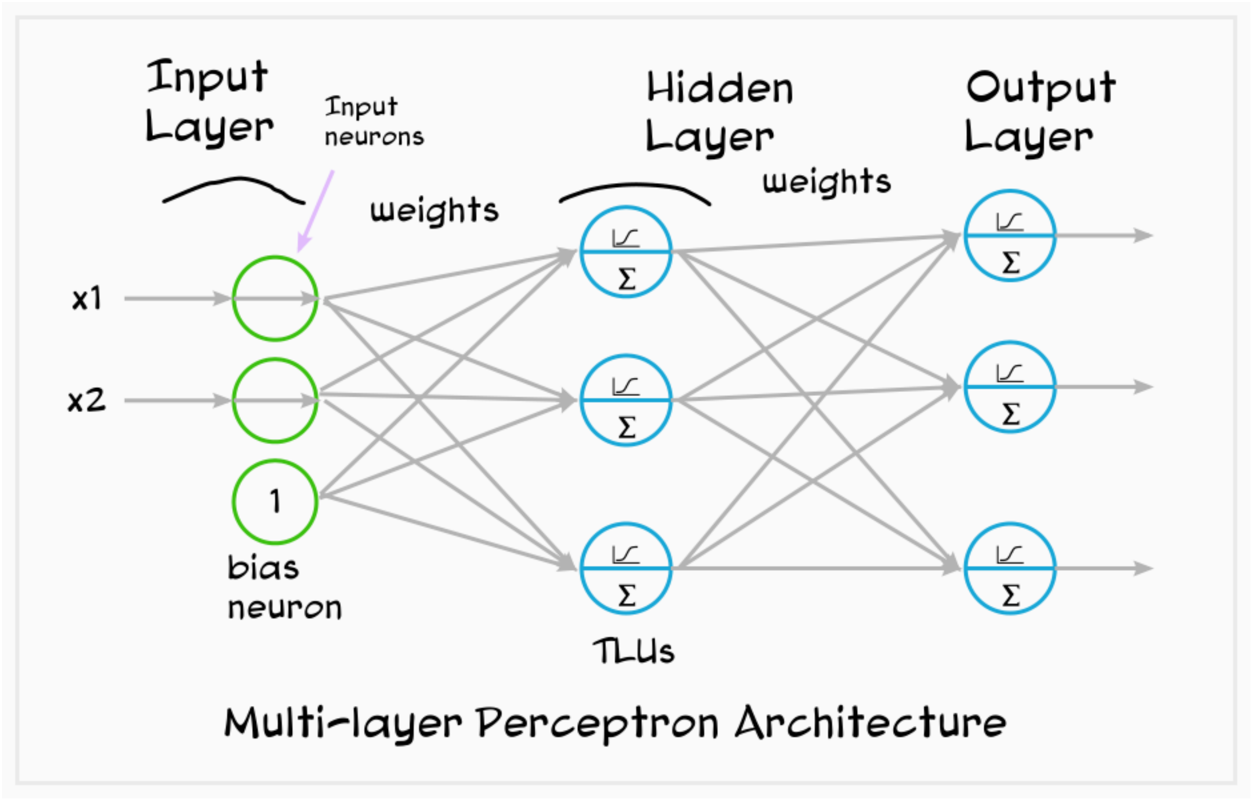
- When you differentiate the error, you get $(y_j - y_j')x_i$
  It is similar to the weight update method used during backpropagation algorithm (coming in next section):

  $$w_x^{new} = w_x - \eta(\frac{\delta Error}{\delta w_x})$$



Multi-layer Perceptron

- Addition of Hidden layer between Input and Output layer



- An MLP is composed of:
  - one input layer (passthrough)
  - one or more TLUs, hidden layers
  - one final layer of TLUs, output layer
  - Each layer except output layer has bias neuron and fully connected to next layer
- Signal flows in one direction - feed forward neural network
- When Artificial neural networks (ANNs) keeep a deep stack of hidden layers - it's called Deep Neural Network (DNN).
- For many years, researchers struggled to train MLPs.
  In 1986, Geoffry Hinton & team introduced ground-breaking Backpropagation algorithm which made training MLPs easier.