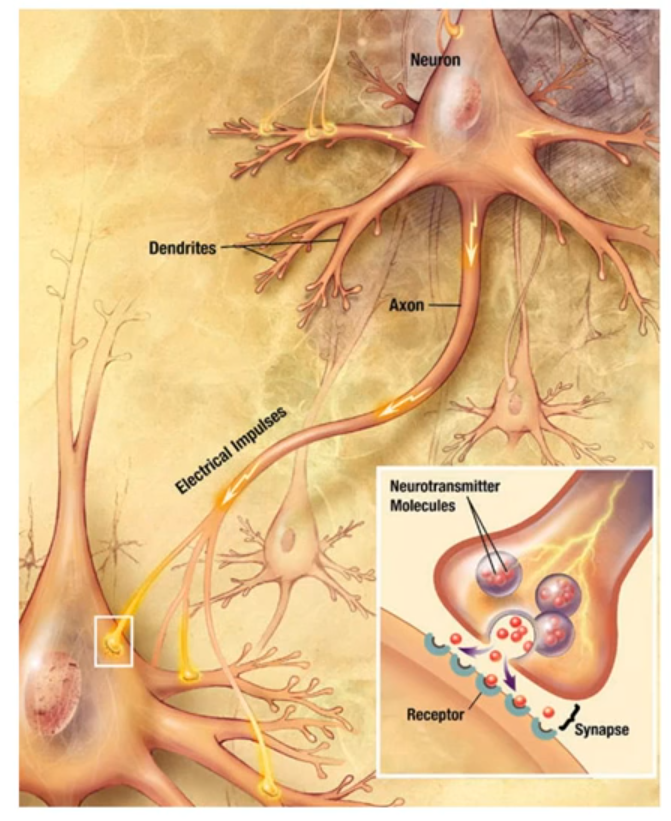


Here, Dendrites are the receiver’s of signal where as Axons are Transmitter of Signal.

Following Figure shows, How actually Neurons looks like,



Axon transmits the signal from one neuron to another neuron. It is partially connected to the other Neurons and the gap region between the neurons are known as Synapse.

Here, Neurotransmitter are the last end of Axon whereas, Receptors are the starting region of Dendrites.

In our human body. We receive/acknowledge every information through our senses and in terms of signals in brain.

Ex :- When we touch any fire by our hands. Then, actually our hands starts burning but we feel pain in brain as a signal. So, Every where in our body neurons are available with the help of nervous system. When we touch fire then, Neurons existing at hand receives heat as a signal from Dendrites and send the same signal to Axon for further signal transfer to next neurons till it reaches to the brain .

That’s why any normal human being can feel touch on his face faster than the touch at his/her legs due to less distance and less time taken to transfer the signal + If the same pressure of touch is applied to face and at leg then, It may happen that person may response to Face touch but he/she may not even get aware of slight touch at his/her leg.

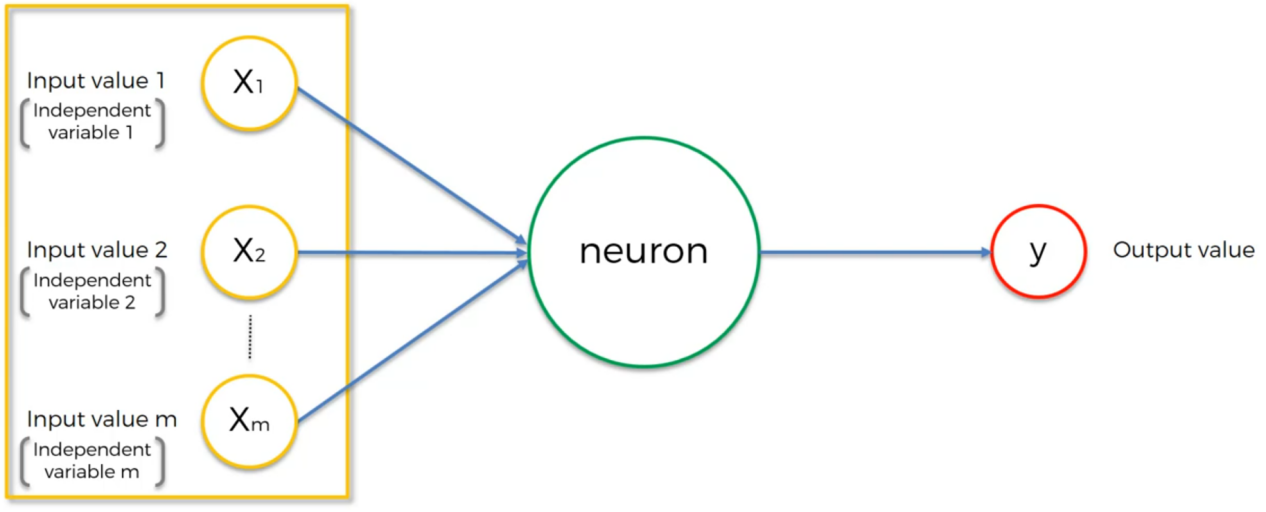
Conclusion of above statement,

Signal Strength is inversely proportional to distance.

If distance is min Then, Signal Strength can be High.

If distance is max Then, Signal Strength can be low.

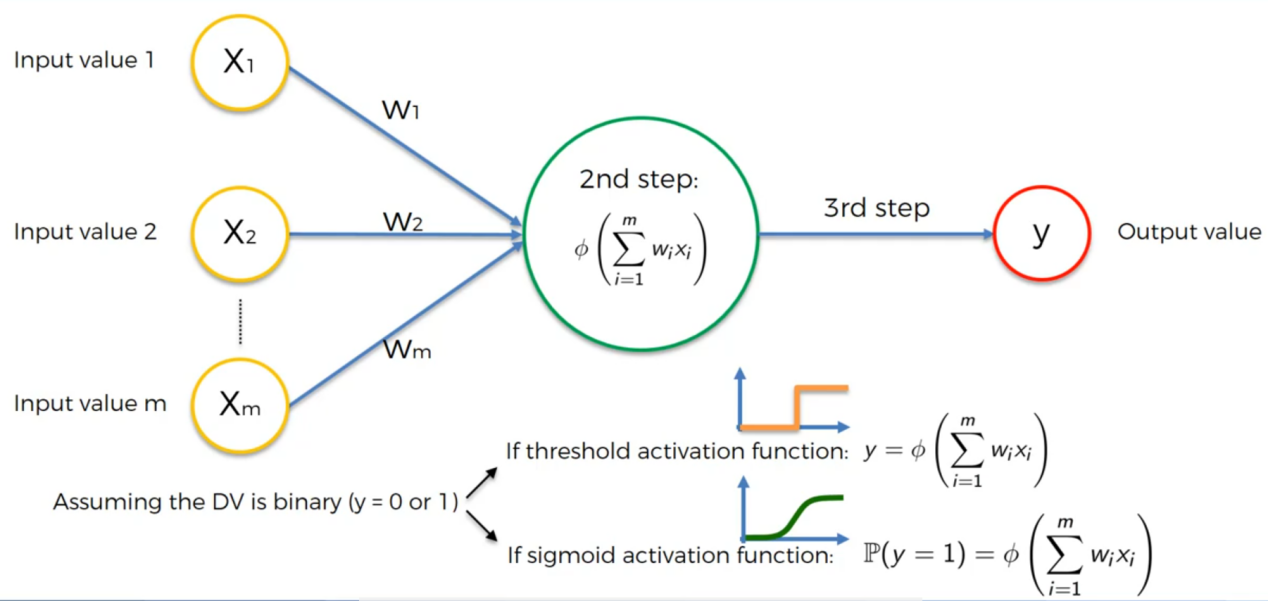
Technically, Neuron Works in machine learning in a following way,



Here,

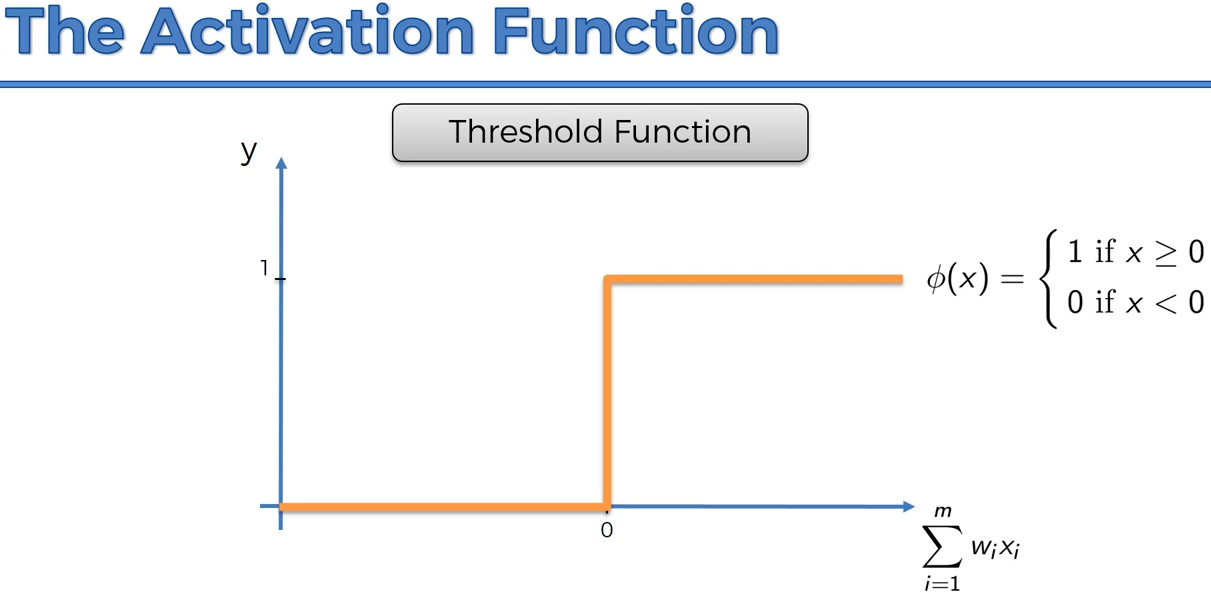
**X1, X2, …, Xm are the input columns Where are neuron is the processing Part and Y is the Final Output.**

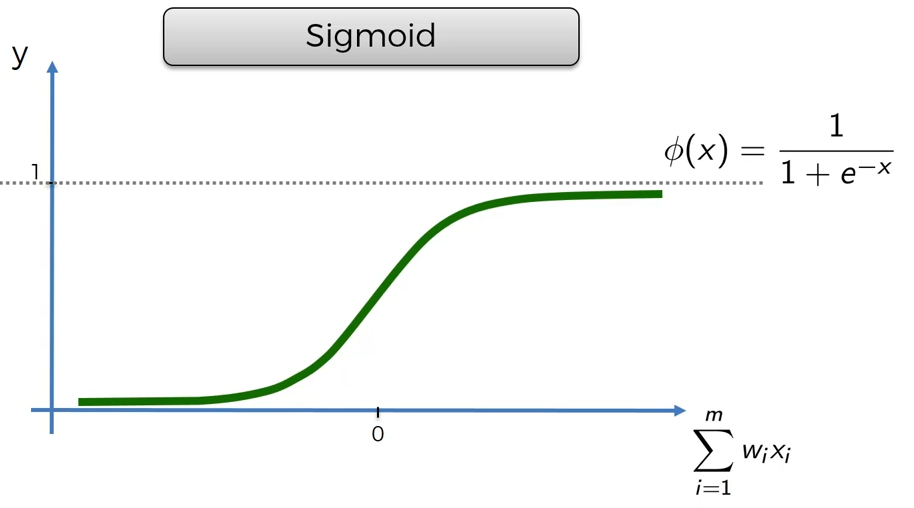
Now, We will discuss about the weight in following figures,

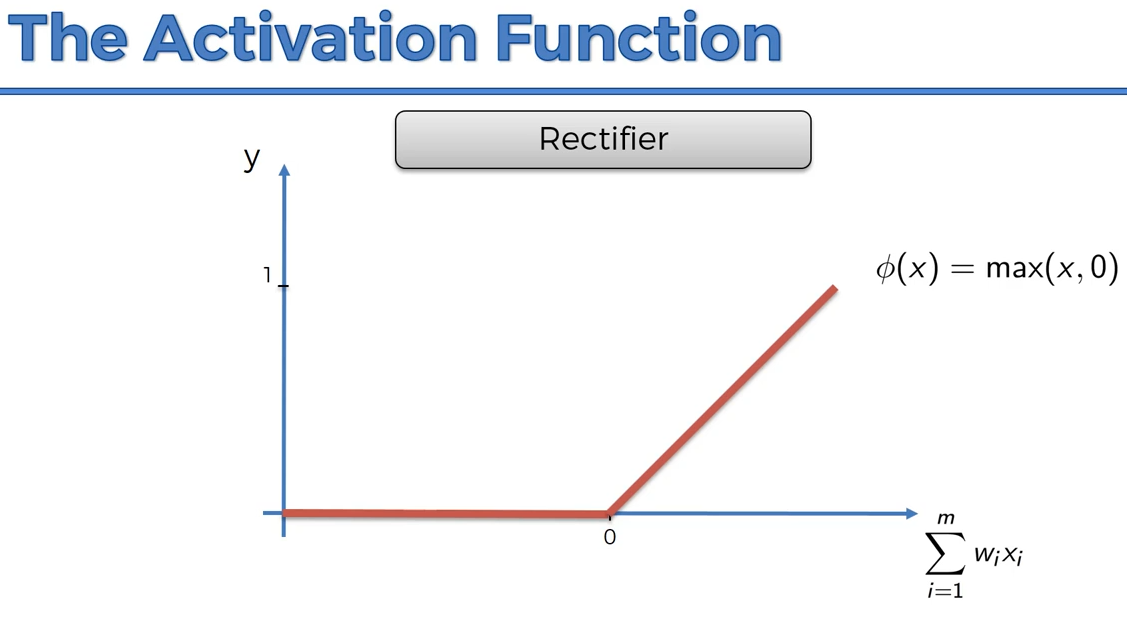


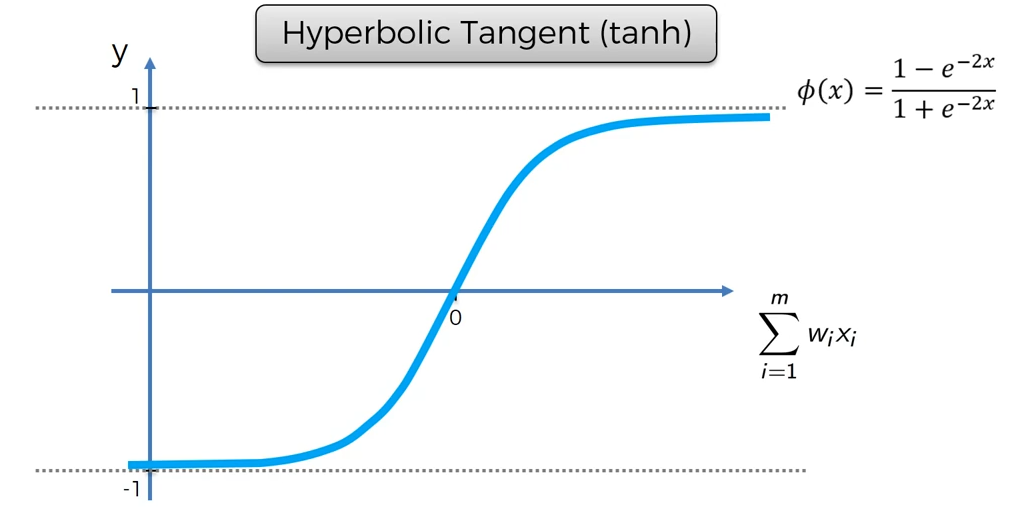
Here, Weights are the Co-efficient of all the inputs . It helps in adjusting the inputs.

0<= weight <= 1.









Ex :- X1 = 5, X2 = 9 and X3 = 2 are the 3 inputs. Where W1 = 0.6, W2 = 0.2 and W3 = 0.4.

Sum = 5+9+2 = 16

Then,

X1 =X1 \* W1 = 5 \* 0.6 = 3

X2 = X2 \*W2 = 9 \* 0.2 = 1.8

X3 = X3 \*W3 = 2 \* 0.4 = 0.8

Sum = 3+1.8+0.8 = 5.6

Relu Function -> max( 5.6, 0 ) -> 5.6 ( Pass )

So, What happens inside neuron,

After doing the summation of all the X input(s), sum = X1 + X2 + …+ Xm .

We check whether the sum is more than the threeshold value Then, Values of sum passes to next neuron.

Understanding the Neural Network Concept with real life example,

Consider a Data Set having columns as **Area**, No. Of **Bedrooms**, **Distance from City**, **Age**( Life Span of House ) as input, our job is to find the price of house as Output ( y ).



Now, For few people Area of House and Distance from city can be highest priority factor. For few, Bedrooms and Distance from city can be the highest property.

Here, Priority are the weight.

Where are weights are for each person, which is the highest priority.

Suppose, For person 1( P1 ), Lets Assume weight is W1.

So, There is a large variety of choices.

So, It is good to keep all the priority and assign each priority to each node.

Ex :-

Person Budget for flat is 40 Lakh

1st Time,

Area = 2000 ft2 ( 45 lakh ) , Distance = 1 km( 10 lakh ), Age = 50 yrs ( 20 lakh ) and Bedroom = 3BHK ( 15 lakh )

W1Area = 0.8, W1Distance = 0.9, W1Age = 0.6 and W1Bedrooms = 0.7

Now Output of Person is,

Y = W1Area \* Area+ W1Distance \* Distance + W1Age \* Age + W1Bedrooms \* Bedroom .

Y = 0.8\*45,00,000 + 0.9\*10,00,000 + 0.6\*20,00,000 + 0.7 \* 15,00,000

Y = 34,00,000 + 9,00,000 + 12,00,000 + 11,50,000

Y = 66,50,000

By Analyzing the above calculation and following Figure we can understand, how neural network works.

2nd Time,

Area = 2000 ft2 ( 45 lakh ) , Distance = 1 km( 10 lakh ), Age = 50 yrs ( 20 lakh ) and Bedroom = 3BHK ( 15 lakh )

W1Area = 0.2( reducing 0.8 to 0.2 ) , W1Distance = 0.9, W1Age = 0.6 and W1Bedrooms = 0.7

Now Output of Person is,

Y = W1Area \* Area+ W1Distance \* Distance + W1Age \* Age + W1Bedrooms \* Bedroom .

Y = 0.2\*45,00,000 + 0.9\*10,00,000 + 0.6\*20,00,000 + 0.7 \* 15,00,000

Y = 9,00,000 + 9,00,000 + 12,00,000 + 11,50,000

Y = 41,50,000

3rd Time,

Area = 2000 ft2 ( 45 lakh ) , Distance = 1 km( 10 lakh ), Age = 50 yrs ( 20 lakh ) and Bedroom = 3BHK ( 15 lakh )

W1Area = 0.2 , W1Distance = 0.5( reducing 0.9 to 0.5 ) , W1Age = 0.6 and W1Bedrooms = 0.7

Now Output of Person is,

Y = W1Area \* Area+ W1Distance \* Distance + W1Age \* Age + W1Bedrooms \* Bedroom .

Y = 0.2\*45,00,000 + 0.5\*10,00,000 + 0.6\*20,00,000 + 0.7 \* 15,00,000

Y = 9,00,000 + 5,00,000 + 12,00,000 + 11,50,000

Y = 37,50,000

4th Time,

Area = 2000 ft2 ( 45 lakh ) , Distance = 1 km( 10 lakh ), Age = 50 yrs ( 20 lakh ) and Bedroom = 3BHK ( 15 lakh )

W1Area = 0.2 , W1Distance = 0.5( reducing 0.9 to 0.5 ) , W1Age = 0.6( increasing 0.6 to 0.7 ) and W1Bedrooms = 0.7

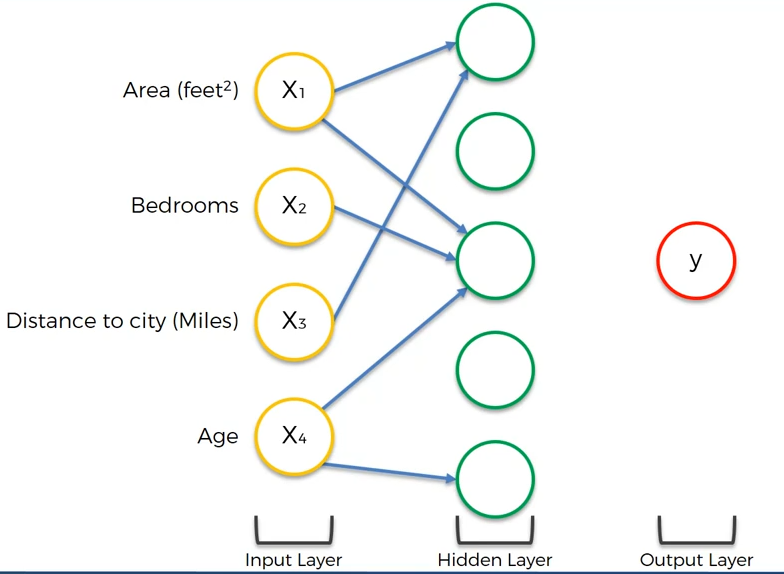
Now Output of Person is,

Y = W1Area \* Area+ W1Distance \* Distance + W1Age \* Age + W1Bedrooms \* Bedroom .

Y = 0.2\*45,00,000 + 0.5\*10,00,000 + 0.7\*20,00,000 + 0.7 \* 15,00,000

Y = 9,00,000 + 5,00,000 + 14,00,000 + 11,50,000

Y = 38,50,000

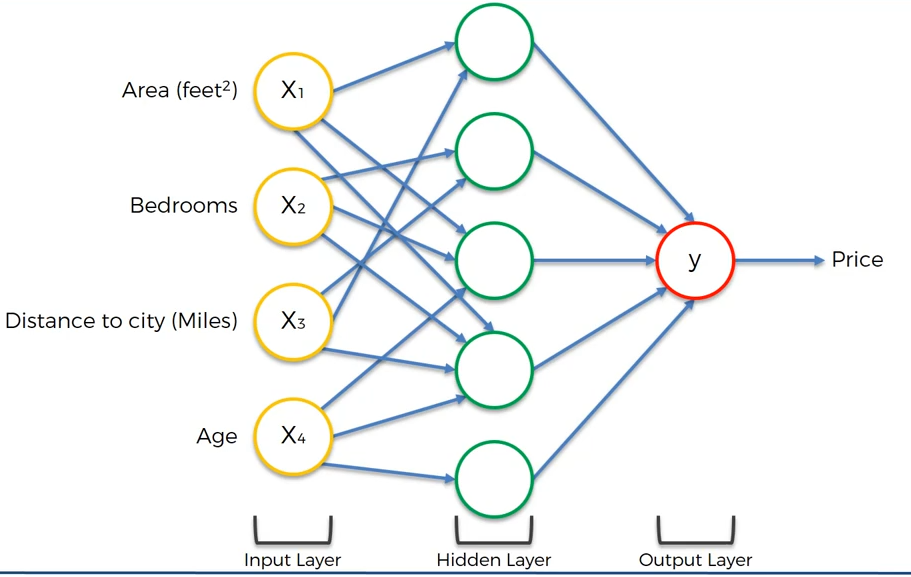


We can See in above Figure,

Input are the First Nodes, Middle Nodes/ Hidden Layer Nodes are Persons and Y is the Output.

* Person1( P1 ) has interest in Area and Distance From City. Here, Person 1 is Hidden Layer Top 1st Node.
* Similarly, Person2( P2 ) has interest in Area, Bedrooms and Age. Here, Person 2 is Hidden Layer Top 3rd Node.
* Then, Person3( P3 ) has interest in only Age. Here, Person 3 is Hidden Layer Last Node.

We can analyze that, we can have multiple Hidden Layer Nodes depending on the Number of persons and their requirement and based on the each hidden layer nodes. We can predict the price of house.



Now, If a person has Gone to a house agent**( Our Neural Network Algorithm )** and his budget is 40 lakh**( Actual Y )** for the house but he is unaware of the locality housing cost. When he started giving priority to house agent then, Agent gave a price of 50 Lakh**( Predicted Y )**. Which is obviously out of budget for him.

Now, Person will have to adjust his priorities**( Weights )** and in order to take the house in that locality, he will have to reduce/ Increases his priorities**( Weights ).**

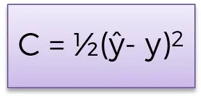
**Or else,**

Agent will have to update the price of house in that locality.

So, Reducing/Increasing the priorities**( Adjusting Weights )** and rechecking the price is happening simultaneously.

Now, In order to stop the rechecking( algorithm ). Agent will have to find the house for client at his desired cost with slight increment/decrements in priorities. So, Every time till the re-checking iteration stops, Cost Diffrence( Error Checking ) will also Occur.

So, For this we have a following **Cost Function** which is used to check the Error,



Ex :- In above calculation we saw, Y = 5.2 but Our desired Y is 4.

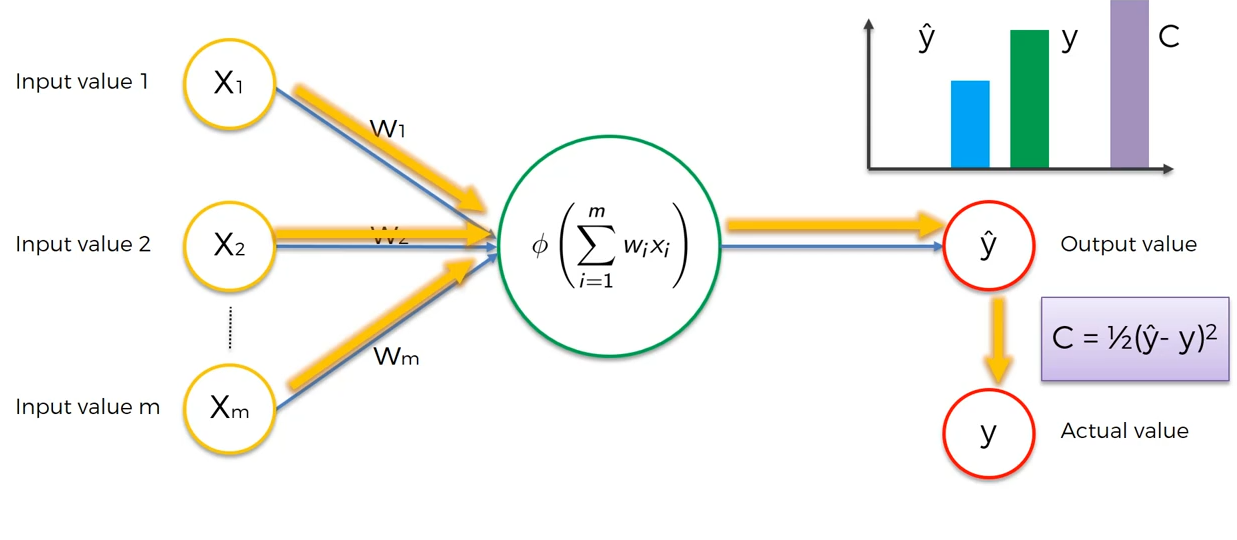
So, Error( Cost Function Value ) = ( 5.2 - 4 )2 / 2 = (1.2)2 / 2 = 1.44/2 = 0.72 is the error.

Note :- ( Actual Y value - Predicted Y value ) and Error both are different. Error denotes the change in weight where as, Y Difference denotes, Difference in Output.

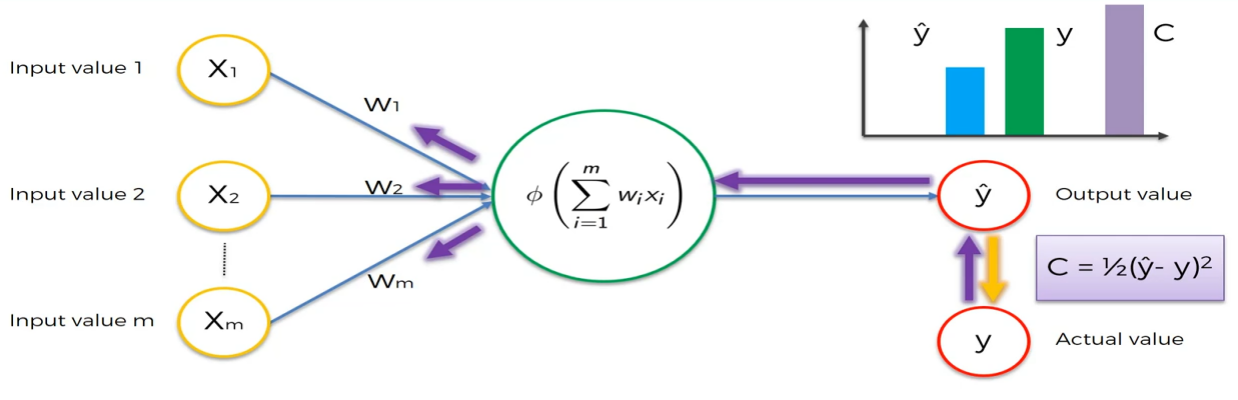
Through Error, we check whether we have to stop training iteration or not.

So, Step1 :-

Find the Error.



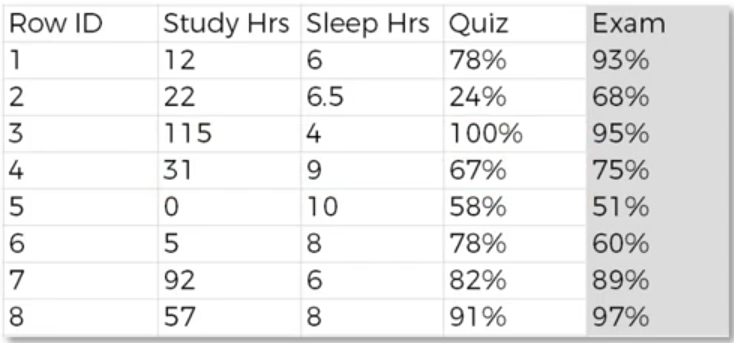
Step 2 : - Send the Error report to input nodes again via activation functions to re-adjust the weights( Priorities )



Note :- Lower the Cost Function means Predicted Output( YPred ) near to our desired Output( YActual ).

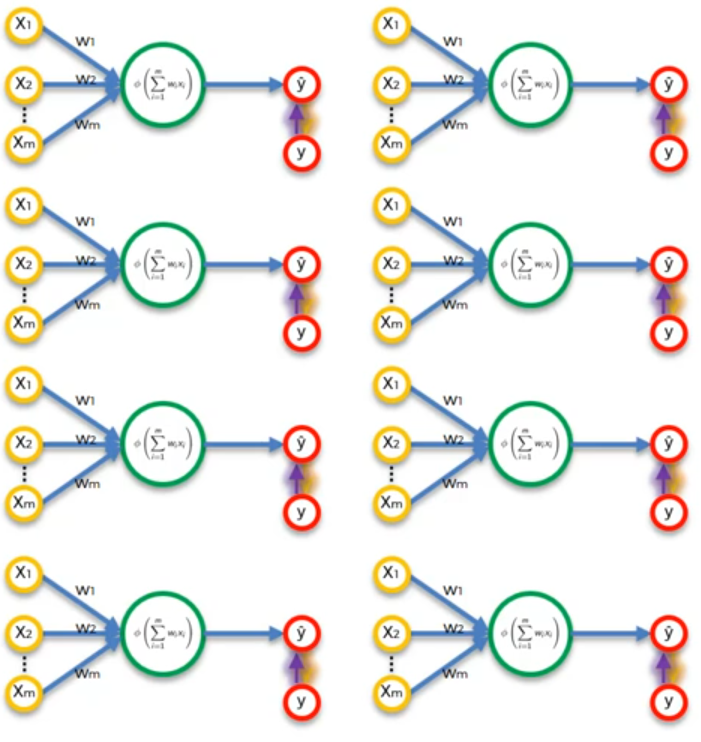
Again, This Flow will Continue till Our Cost Functions stop Changing or leads to 0.

Now, Let Consider a data set which explains the students sleeping time and studying time and their pass fail result.

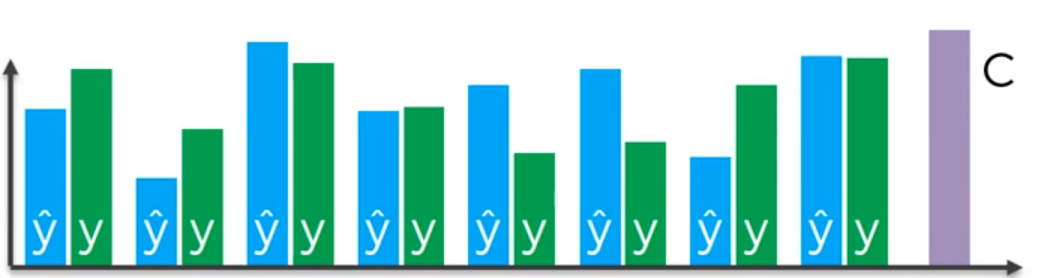


Here, Quiz Column is predicted value and Exam Column is Actual value.

Now, For each row we will we running the training iteration. We have 8 rows, we are running training iterating 1 for all the rows then, adjust the weight then, iteration 2 for all the row and so on.



Let’s Draw Actual Y and Predicted Y on Graph for Each Row and at the end of last row, we find C( Cost Function Error ) which will be same for the rows.



Then, After Iteration 1 Again, we are send C value as feedback and the weights will get adjusted.

