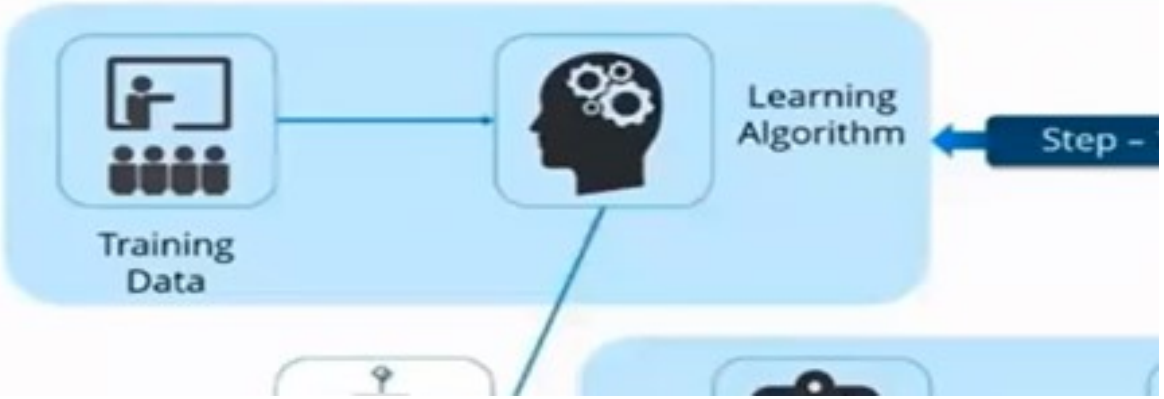
Deep Learning


Artificial Intelligence
Machine Learning
We'll learn more about De
networks and Neural
Machine Learning is a
subset of AI

Types Of Machine Learning – Supervis

Supervised Learning is where you have input variables (x) and an o
an algorithm to learn the mapping function from the input to the o



Types Of Machine Learning – Unsupe

- Unsupervised Learning is the training of a model using information that is neither classified nor labelled.
- This model can be used to cluster the input data in classes on the basis of their statistical properties.

Types Of Machine Learning – Reinfor

- Reinforcement Learning (RL) is learning by interacting with a s
- An RL agent learns from the consequences of its actions, rathe explicitly. It selects its actions on basis of its past experiences (choices (exploration).

Limitations Of Machine Learning

- One of the big challenges with traditional Machine Learning m
  feature extraction.
- For complex problems such as object recognition or handwrit
  challenge.

Deep Learning To The Rescue

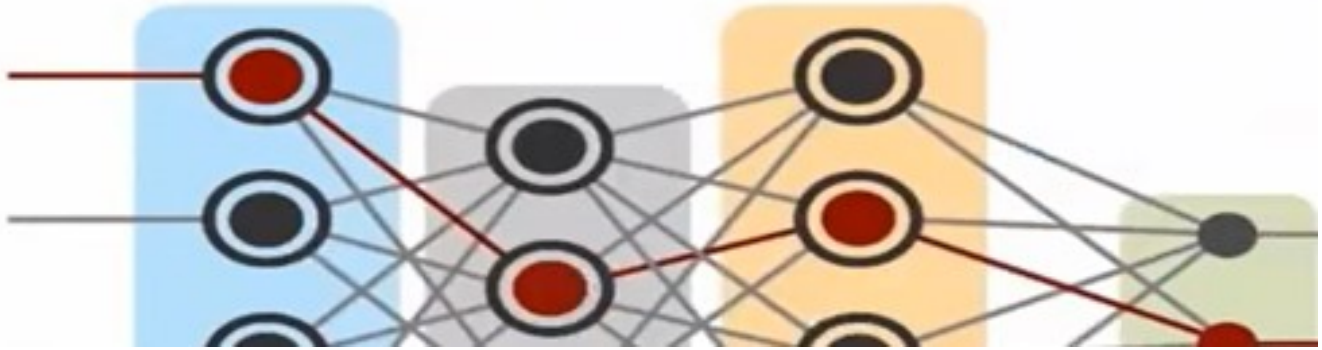
- Deep Learning models are capable to focus on the right featur
  little guidance from the programmer.
- These models also partially solve the dimensionality problem.

Deep Learning To The Rescue

- Deep Learning is implemented through Neural Networks.
- Motivation behind Neural Networks is the biological Neuron.

What Is Deep Learning?

A collection of statistical machine learning techniques used t
often based on artificial neural netwo

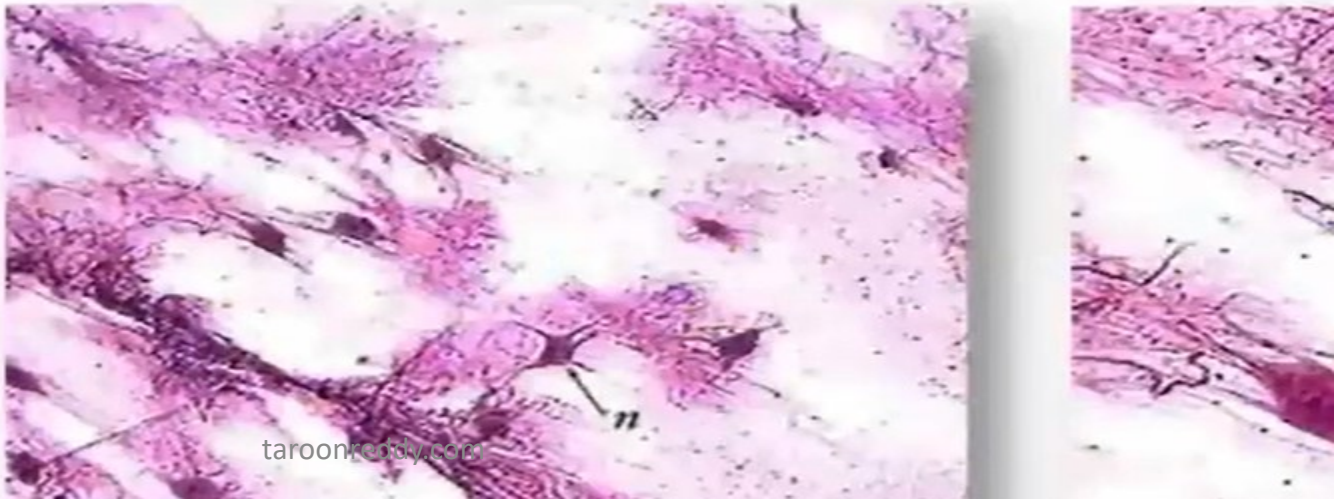


Neural Network

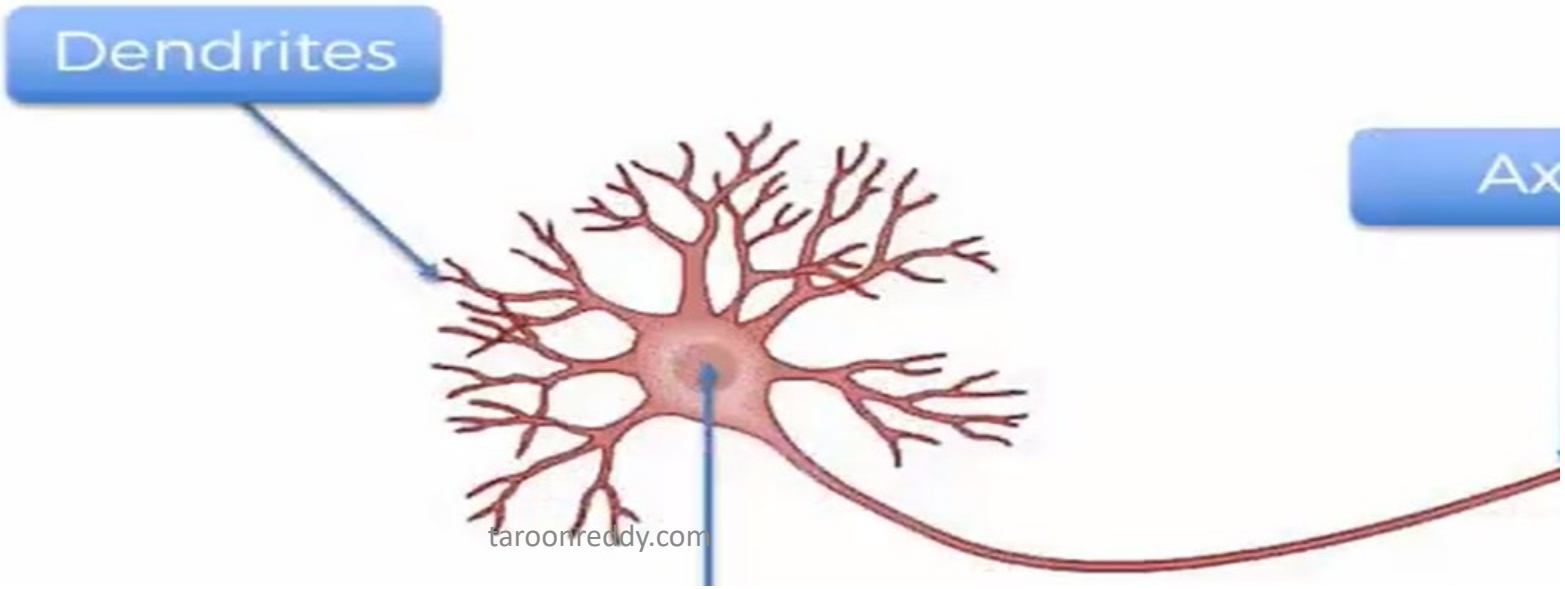
The Neur

taroonreddy.com

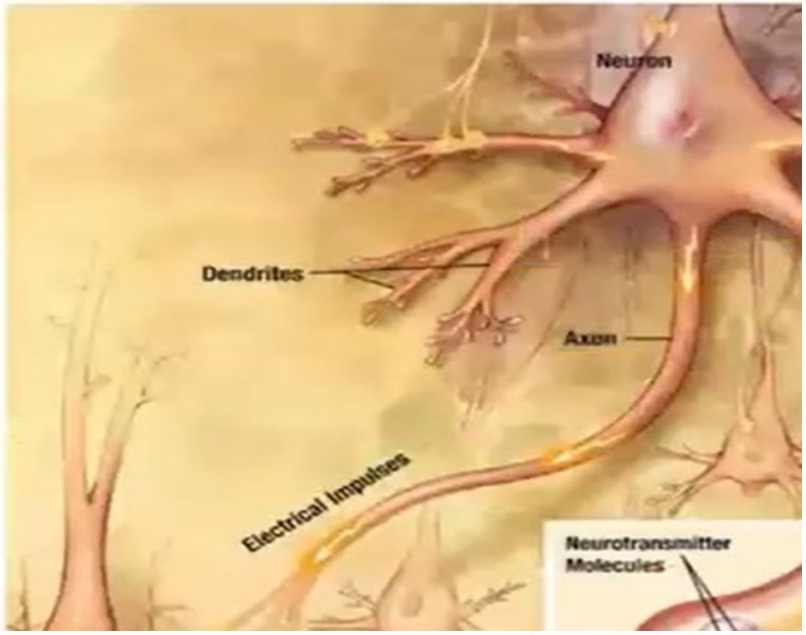
The Neuron



The Neuron

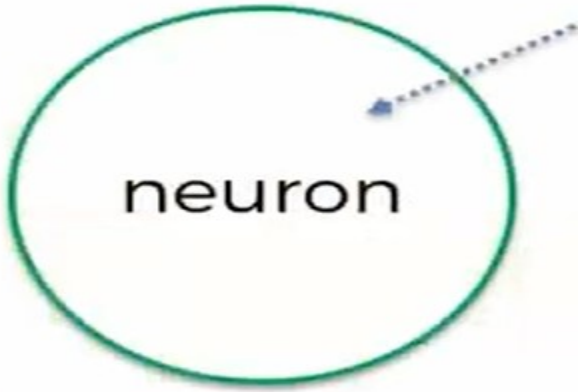


The Neuron

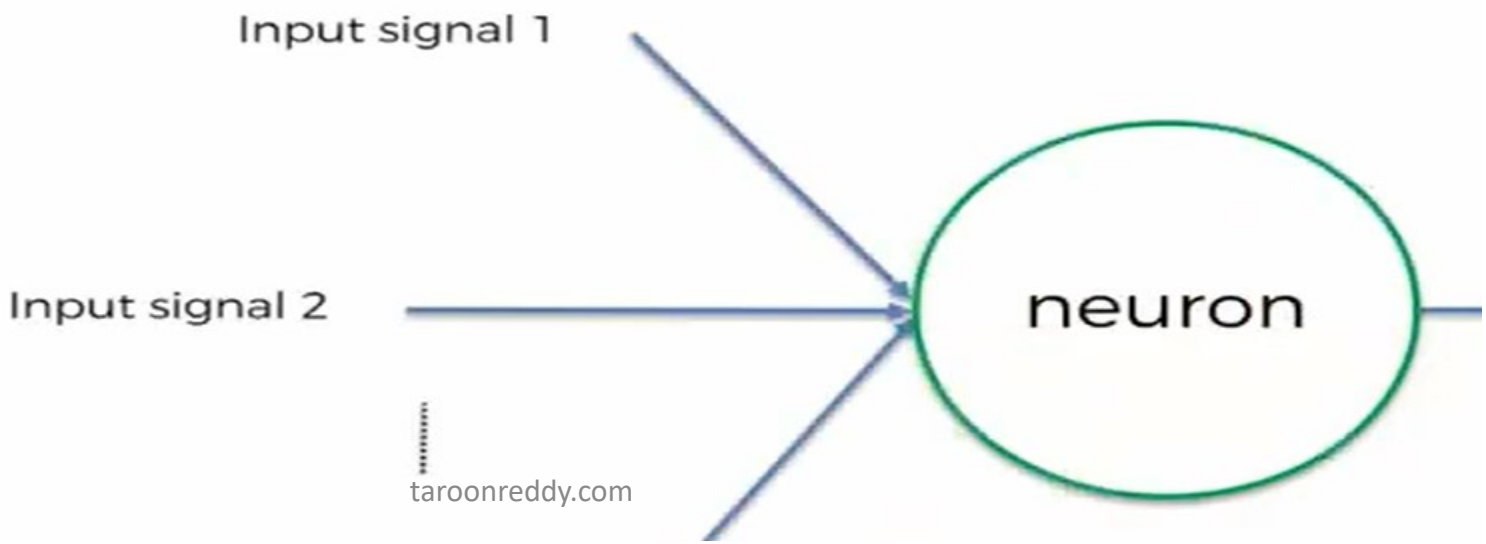


taroonreddy.com

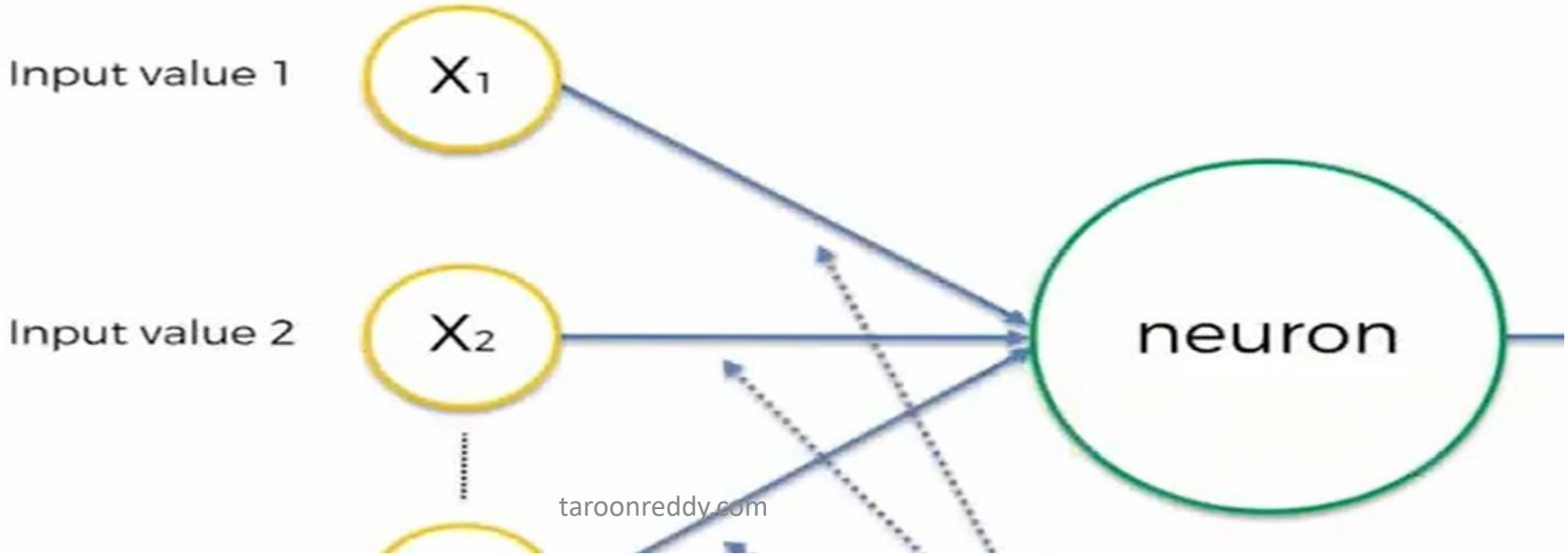
The Neuron



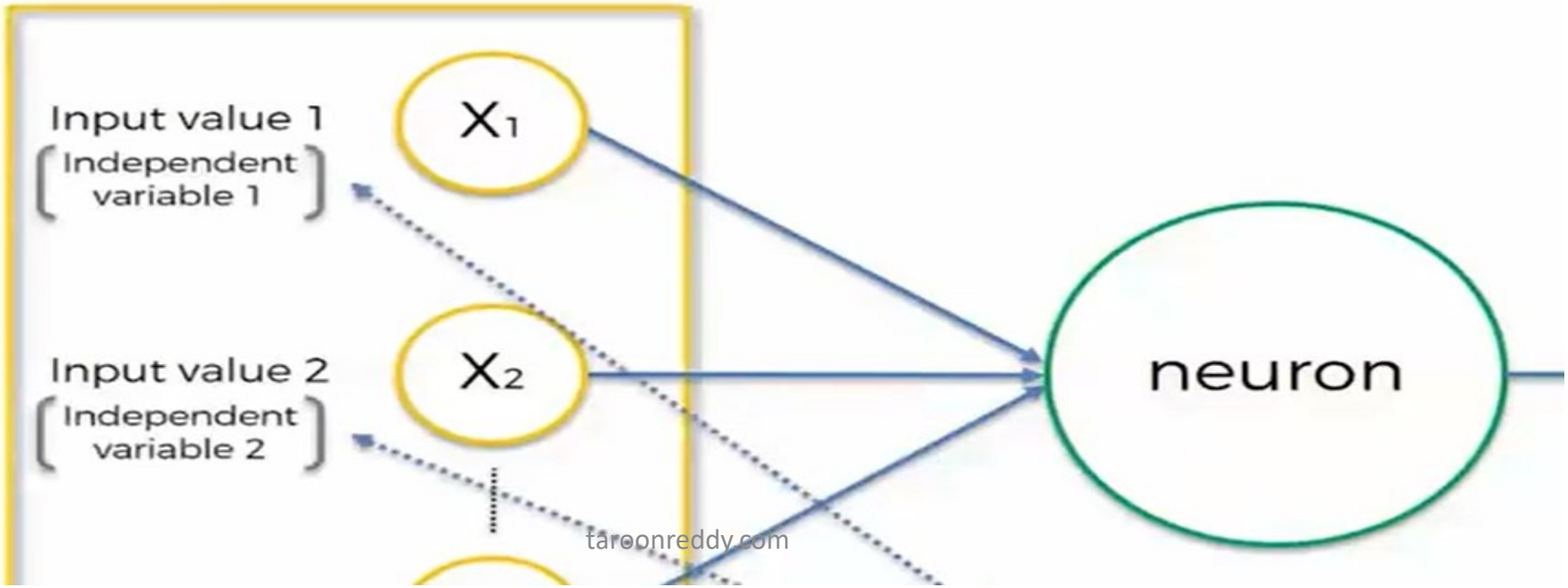
The Neuron



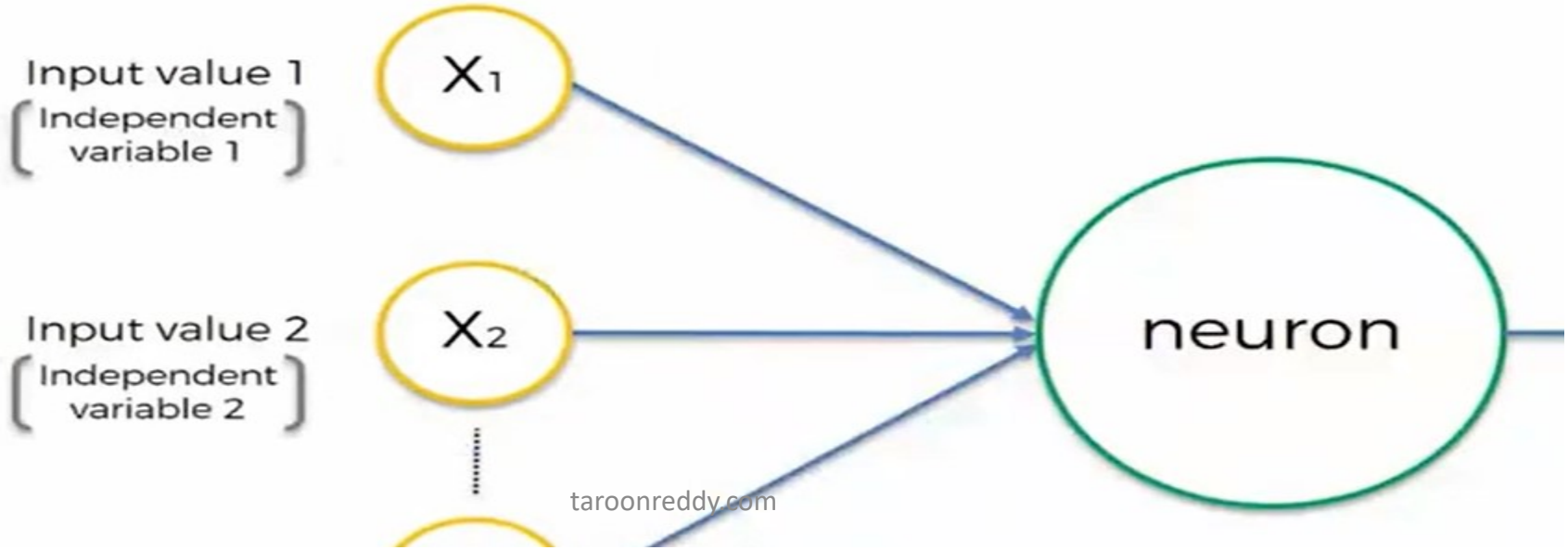
The Neuron



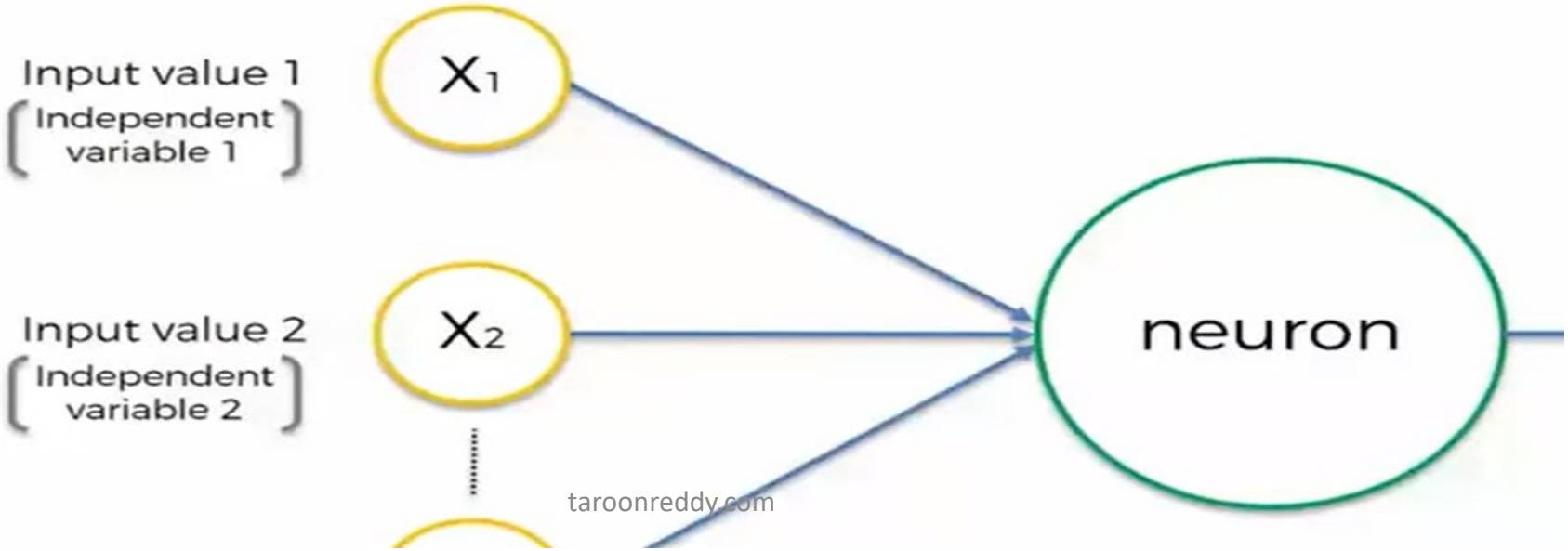
The Neuron


Input value 1
Independent variable 1
X1
Input value 2
Independent variable 2
X2
neuron
taroonreddy.com

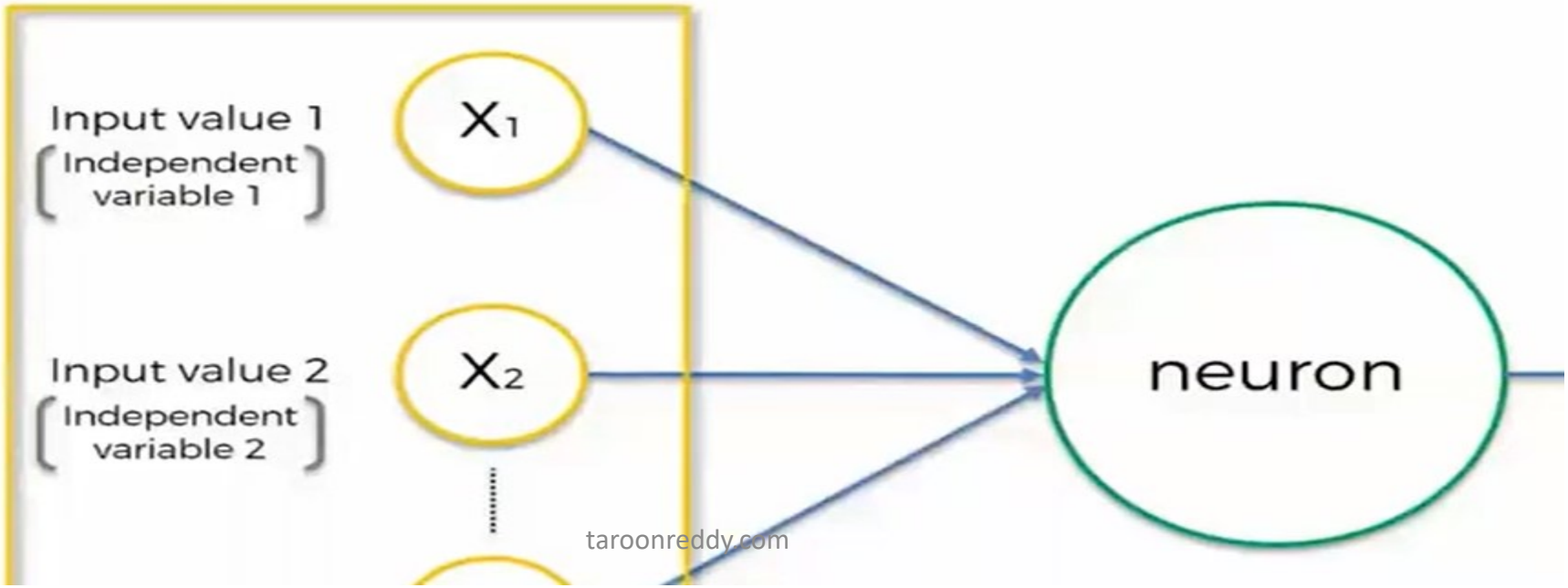
The Neuron



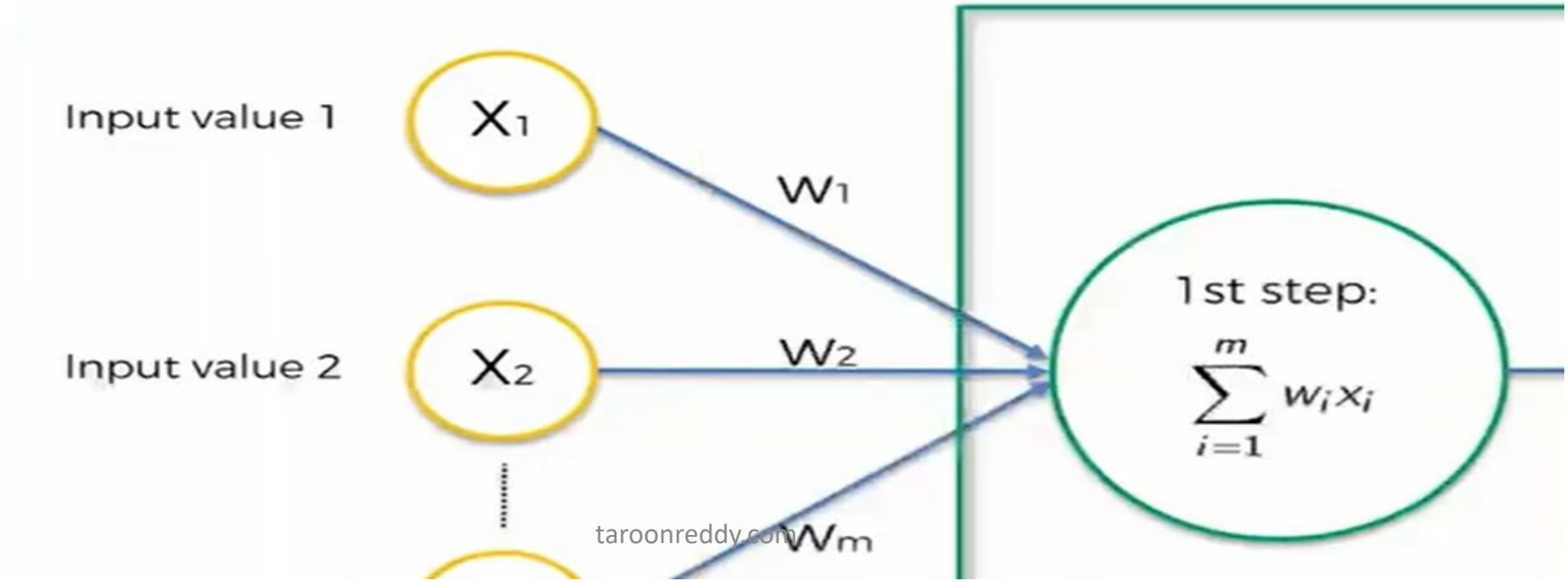
The Neuron



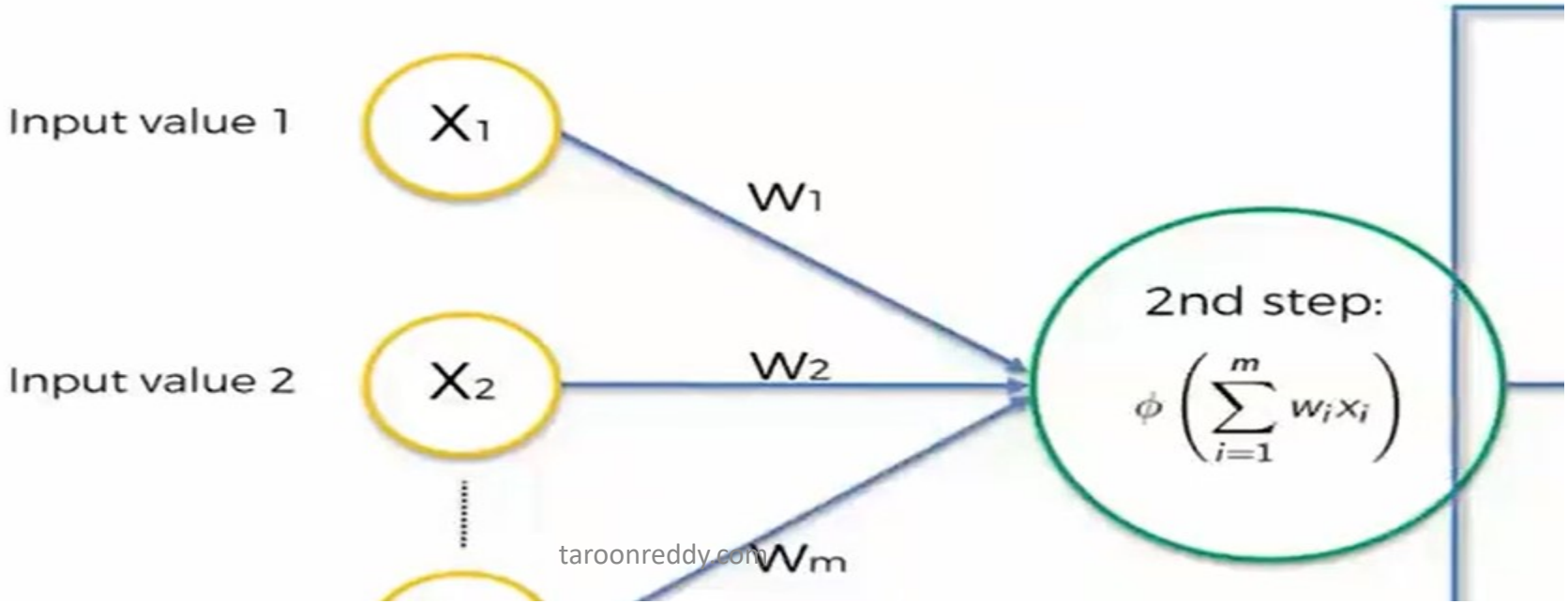
The Neuron


Input value 1
Independent variable 1
X1
Input value 2
Independent variable 2
X2
neuron
taroonreddy.com

The Neuron



The Neuron

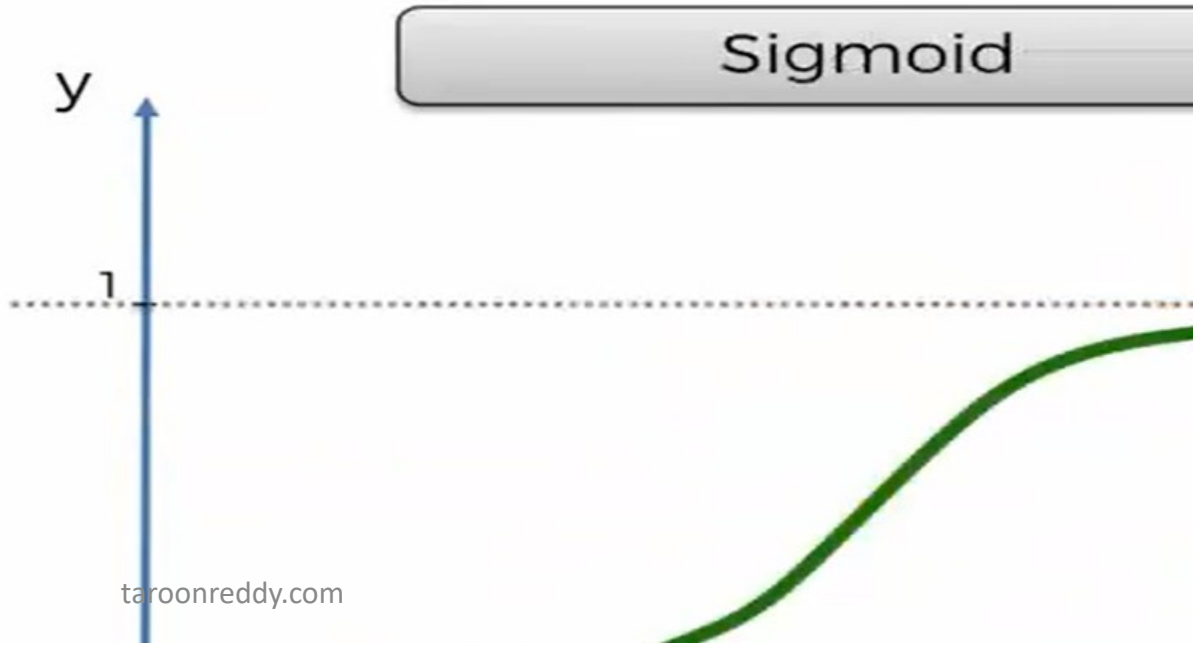


Activation Functions

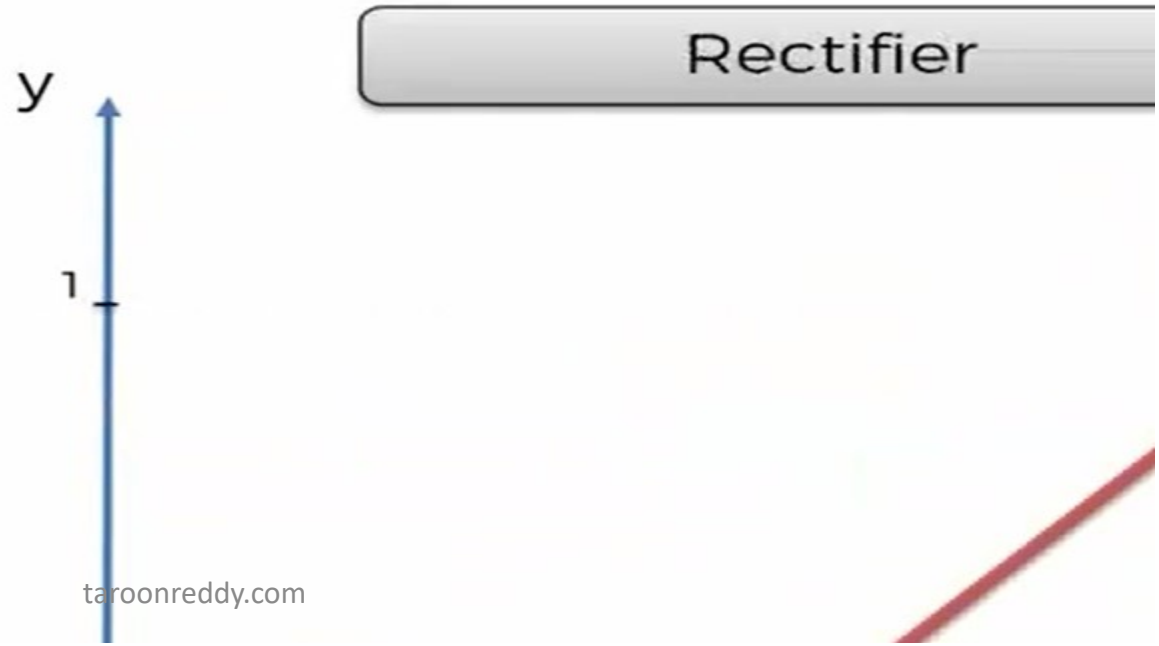
The Activation Functio

Threshold Functio

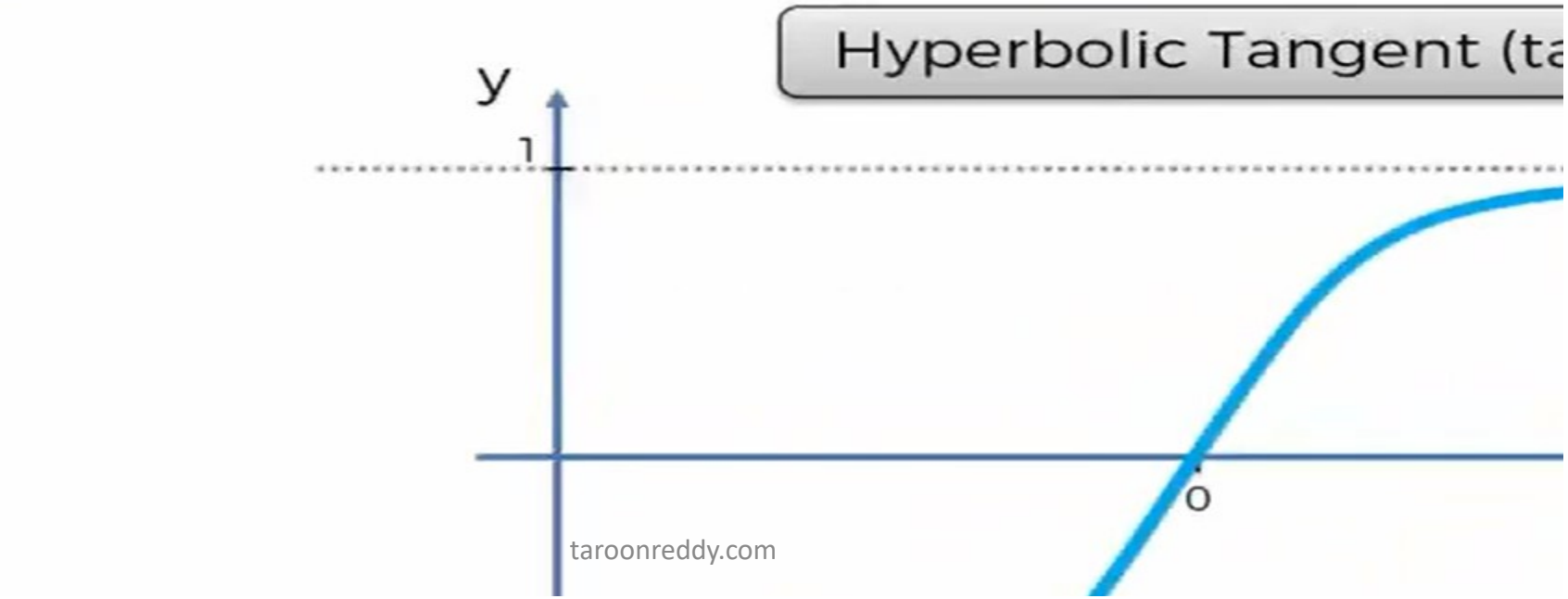
The Activation Functio



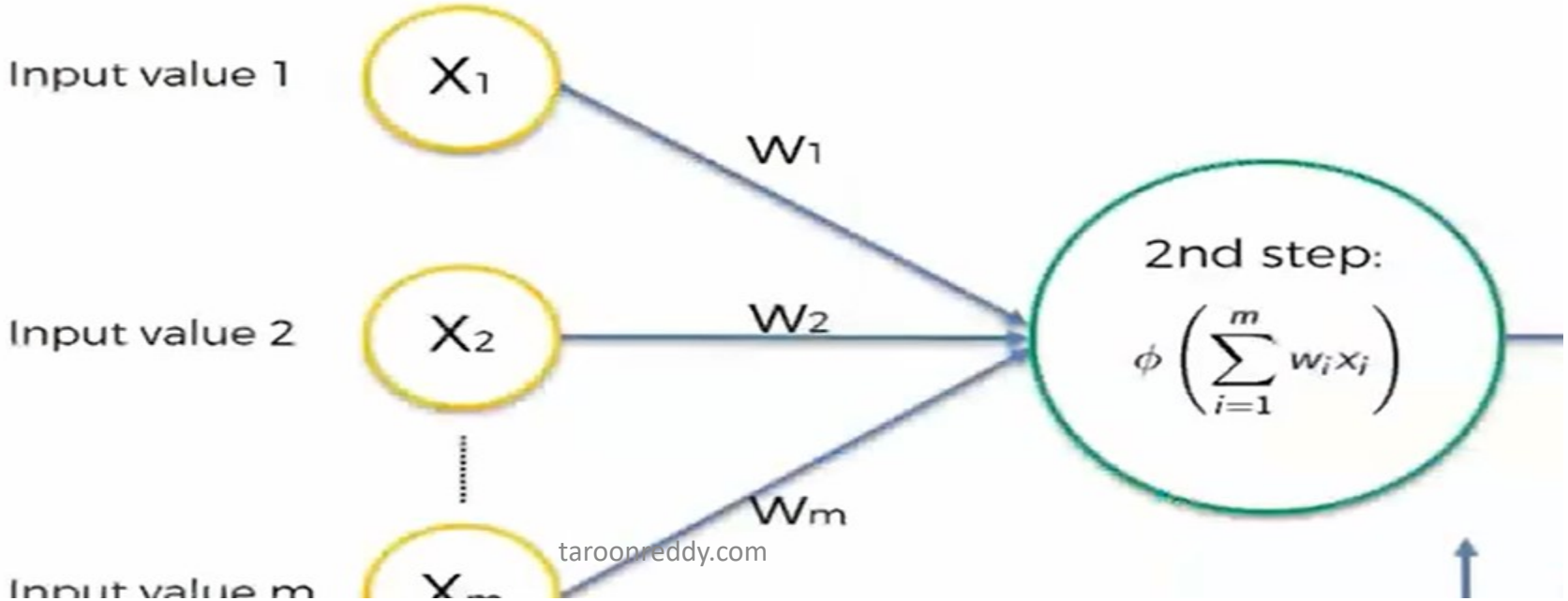
The Activation Functio



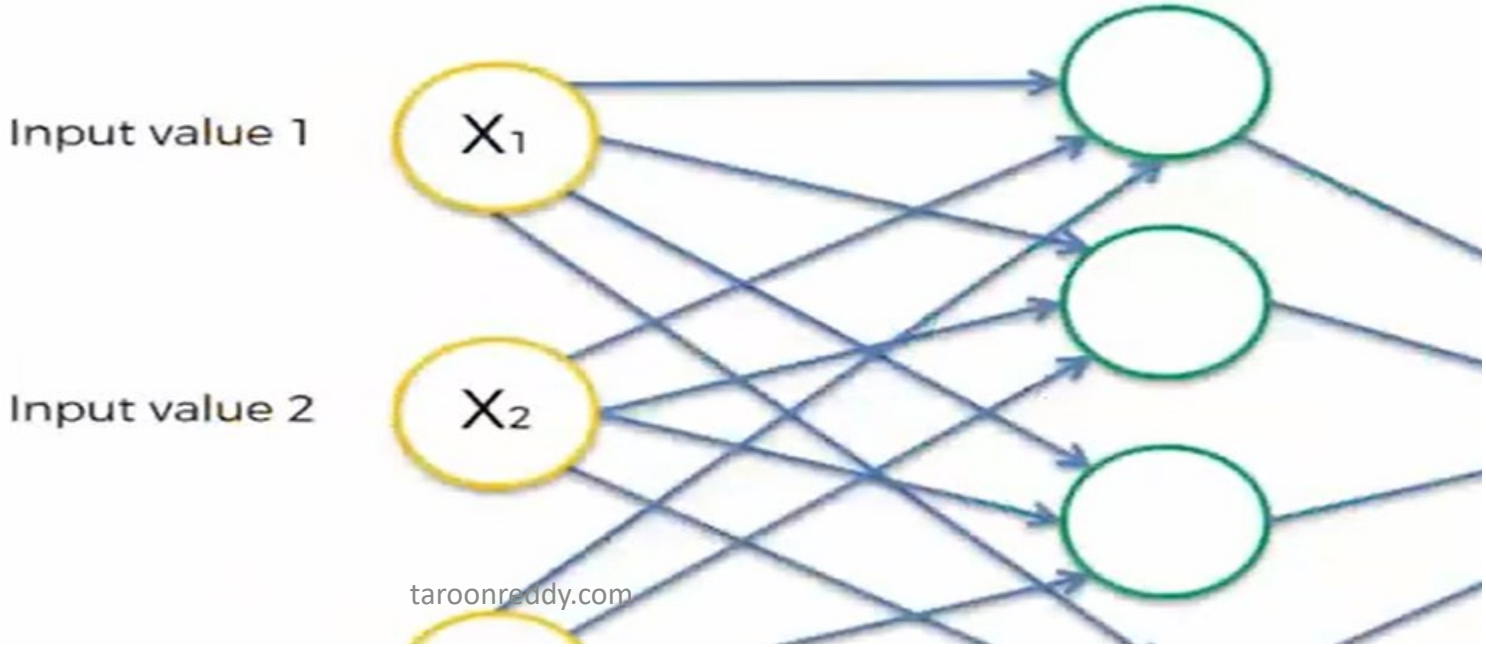
The Activation Functio



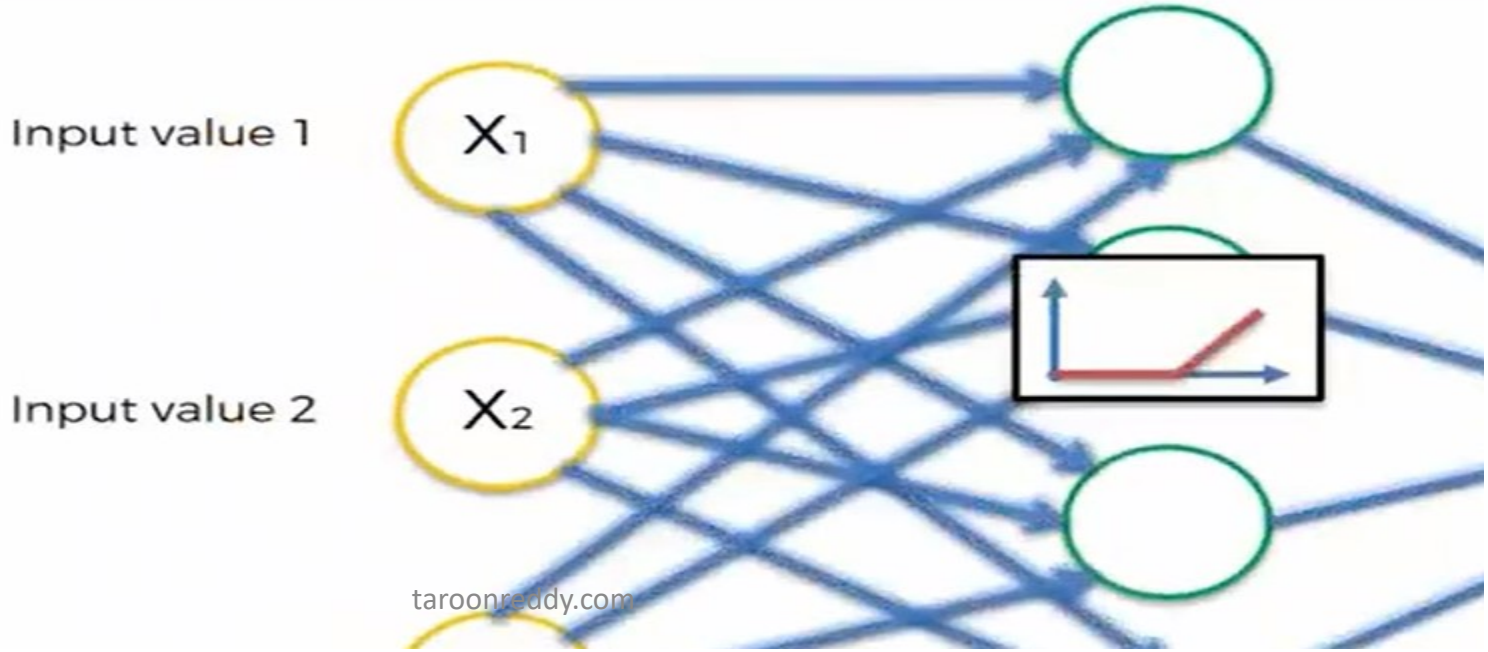
The Activation Functio



The Activation Functio



The Activation Functio

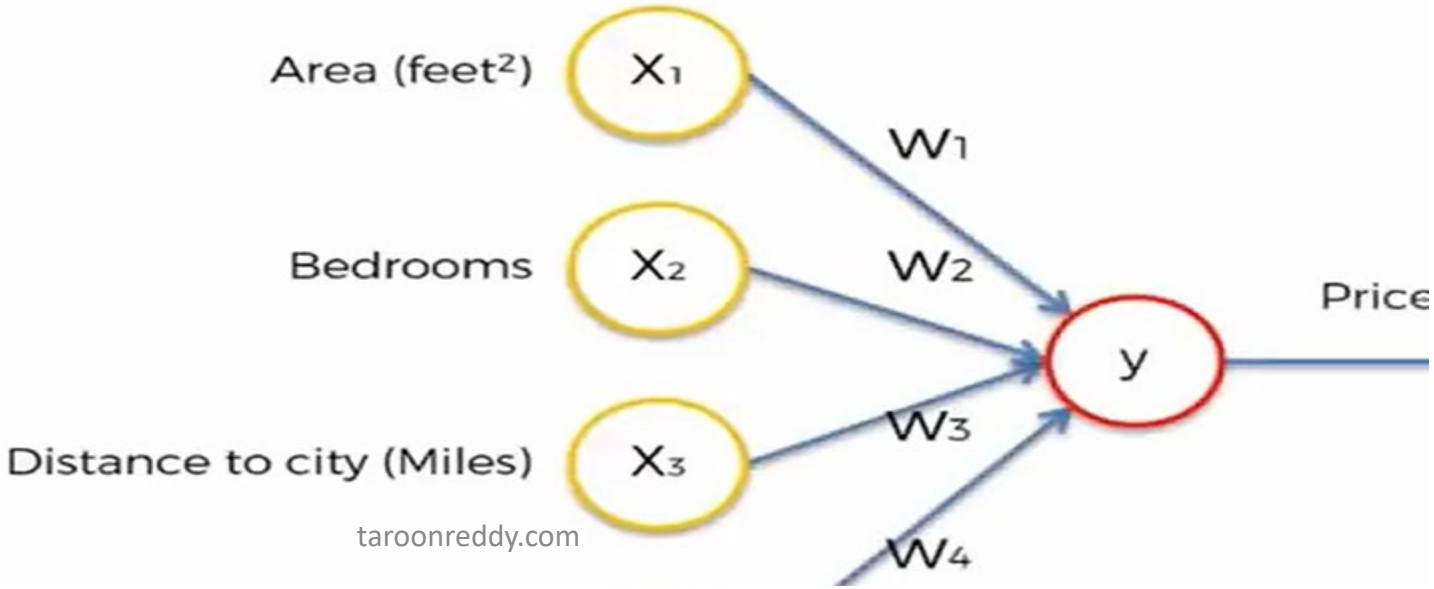


How do NNs

taroonreddy.com

How Do Neural Networ

How Do Neural Networ

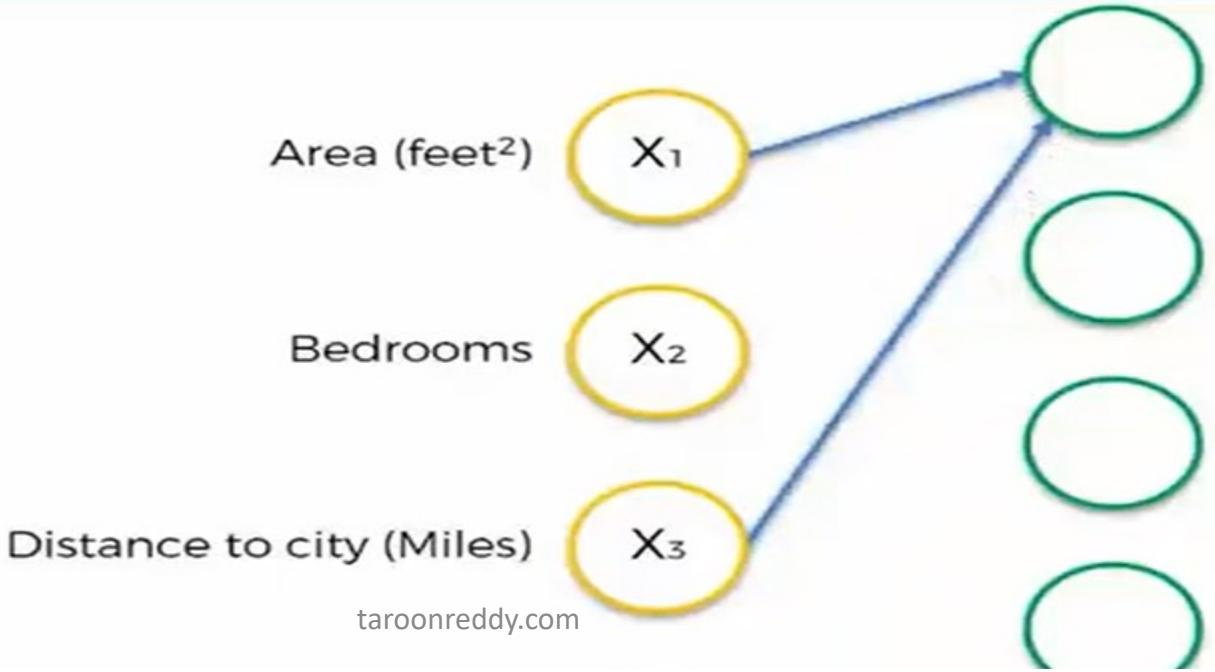


How Do Neural Networ

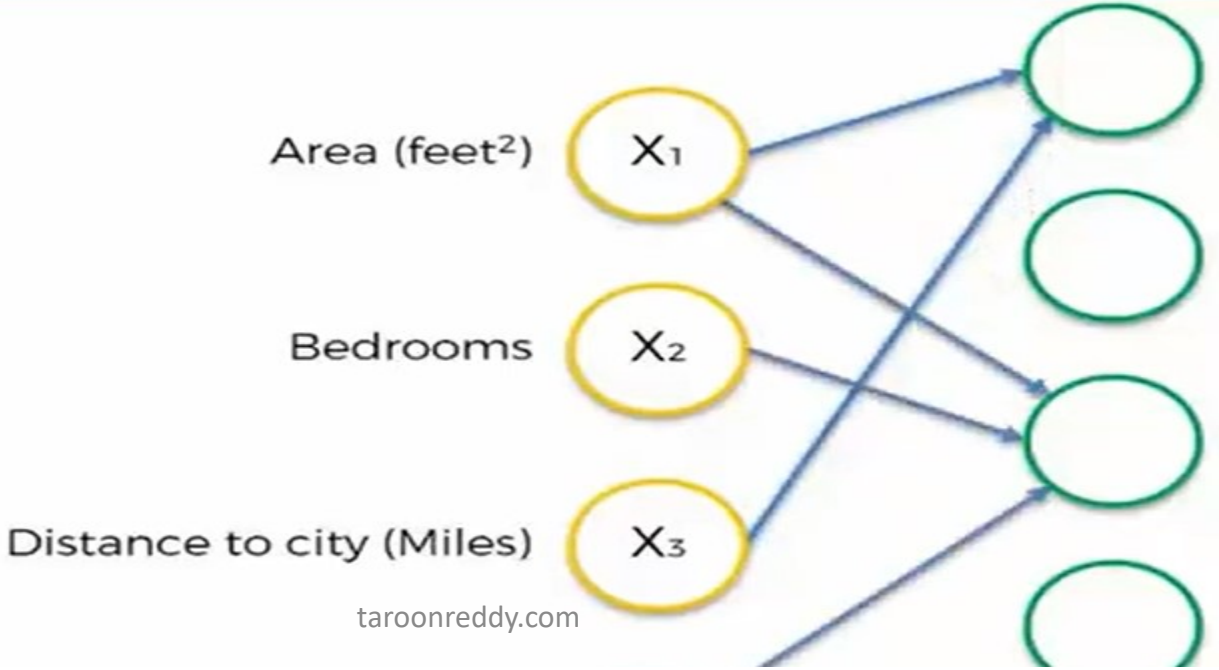
How Do Neural Networ

How Do Neural Networ

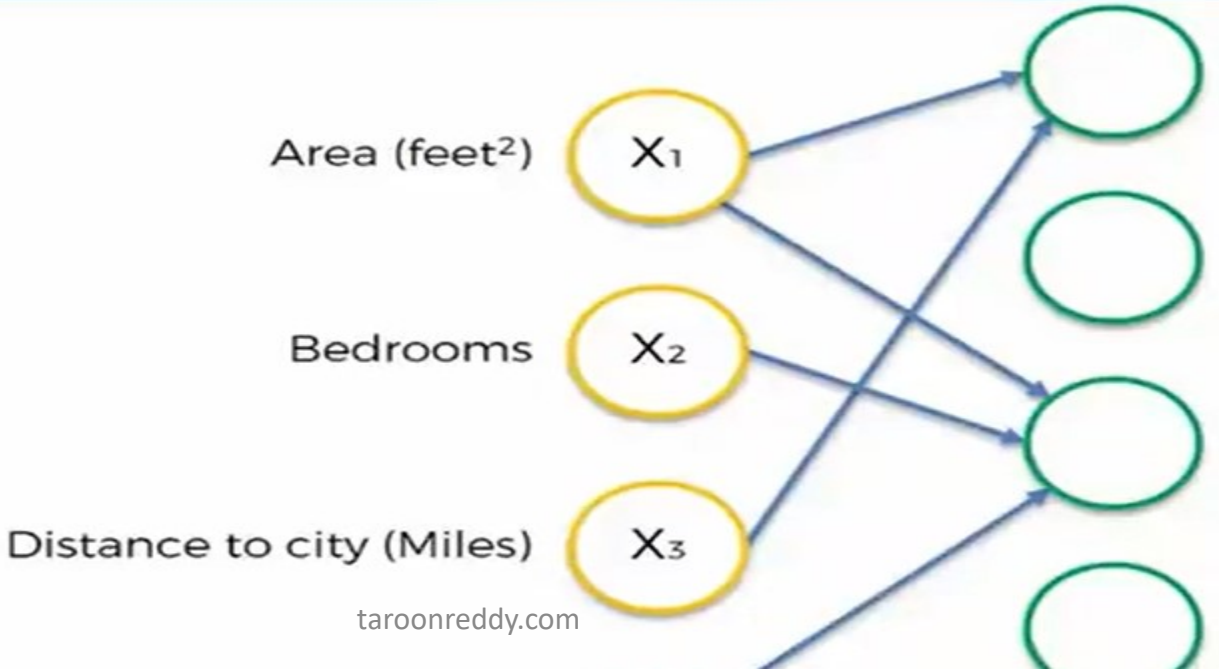
How Do Neural Networ



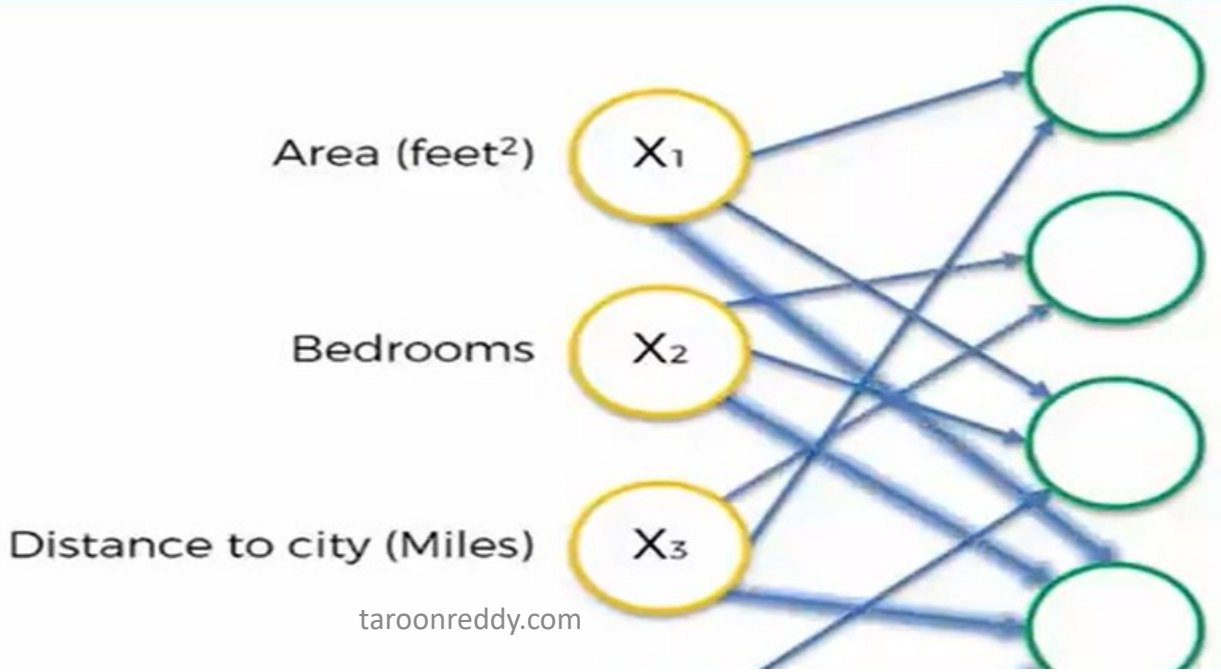
How Do Neural Networ



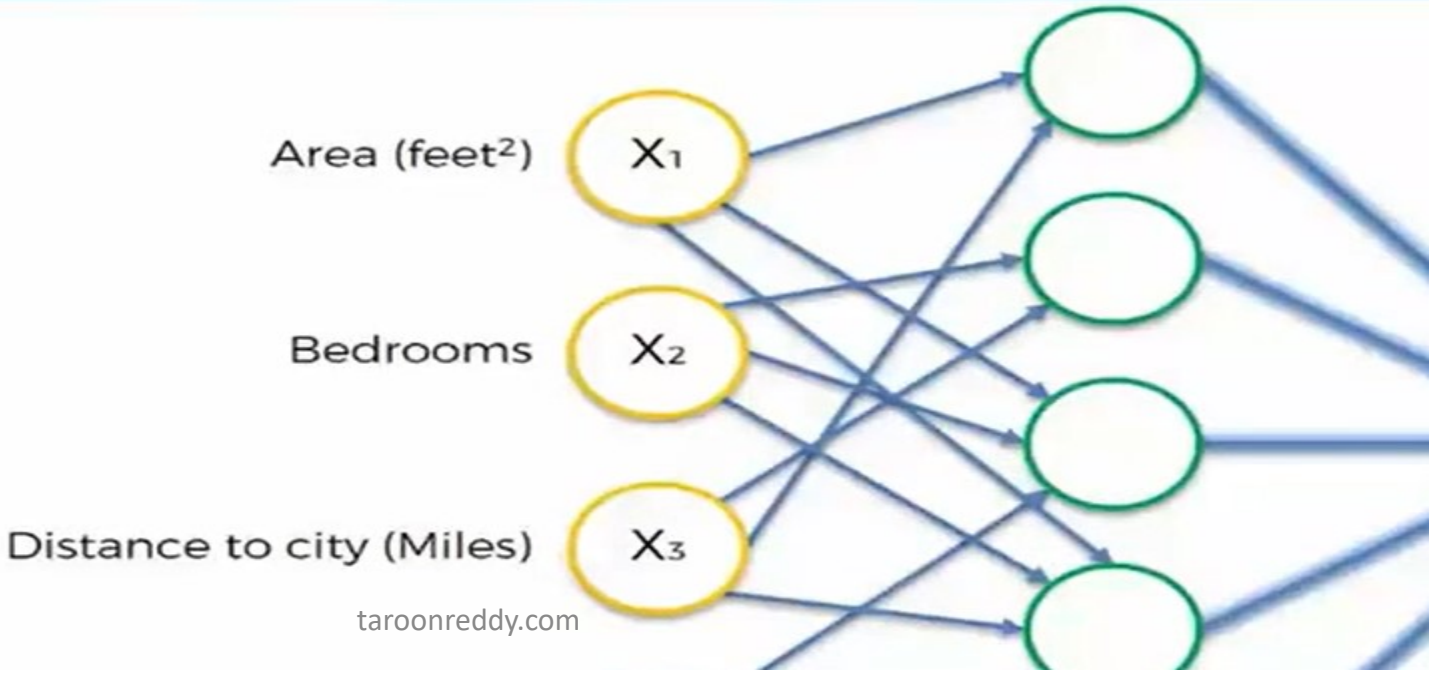
How Do Neural Networ



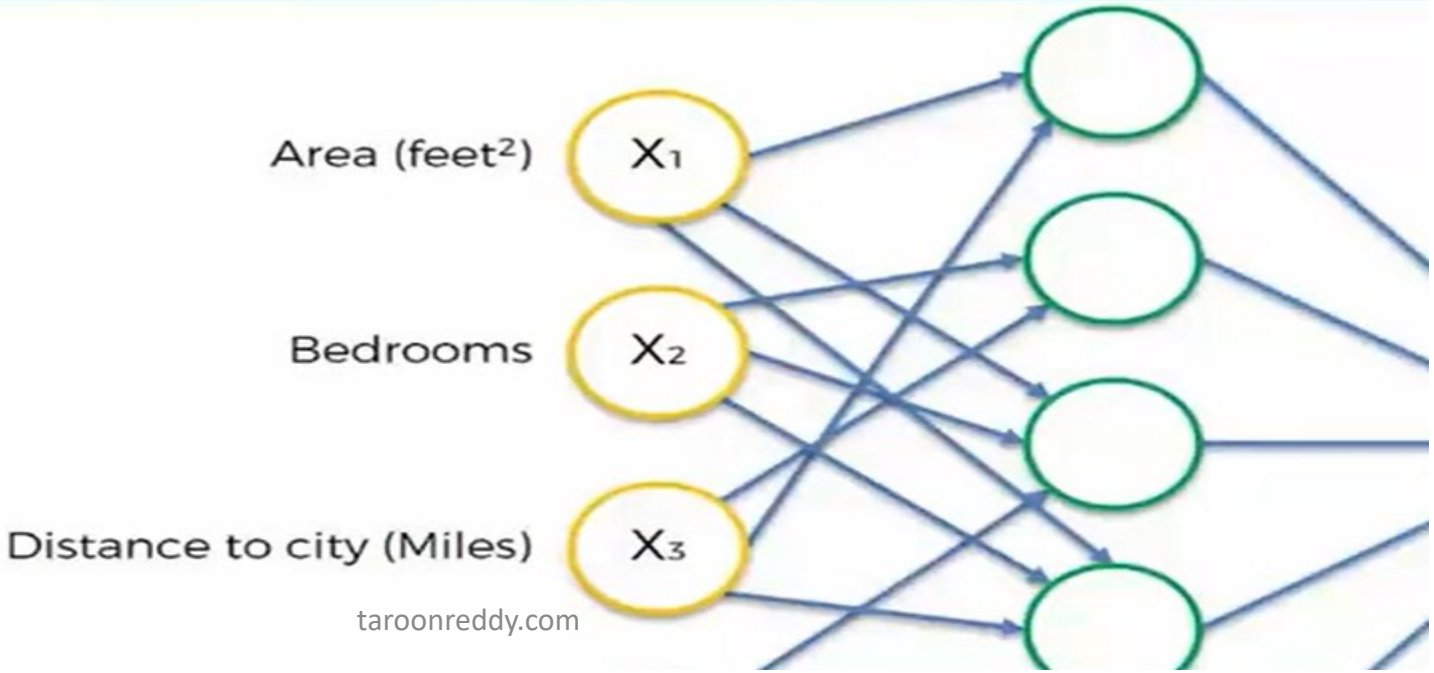
How Do Neural Networ



How Do Neural Networ

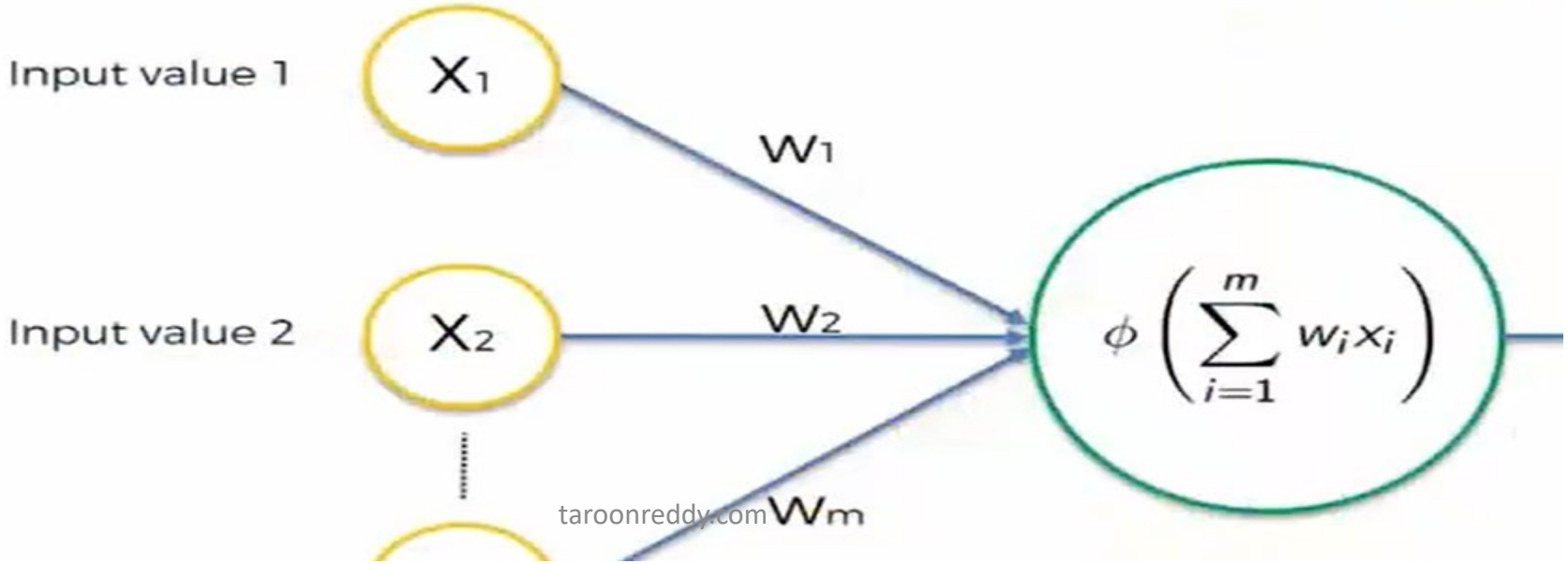


How Do Neural Networ

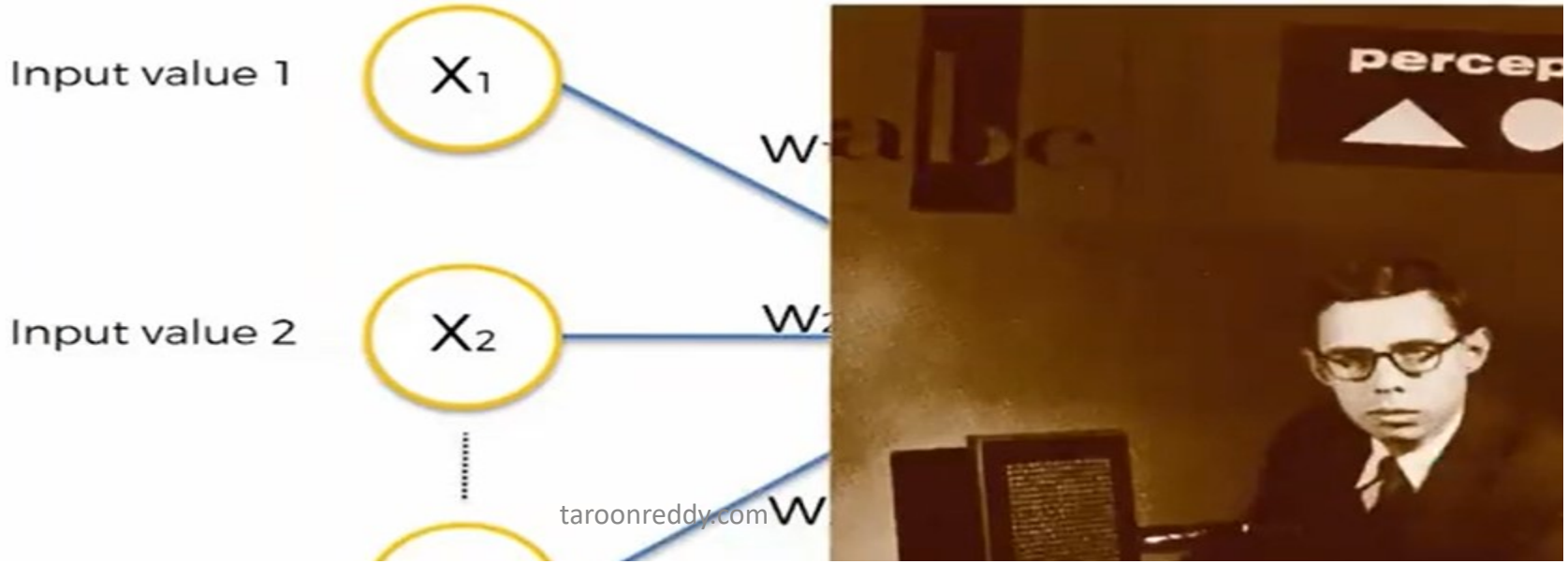


How do Neural Networ

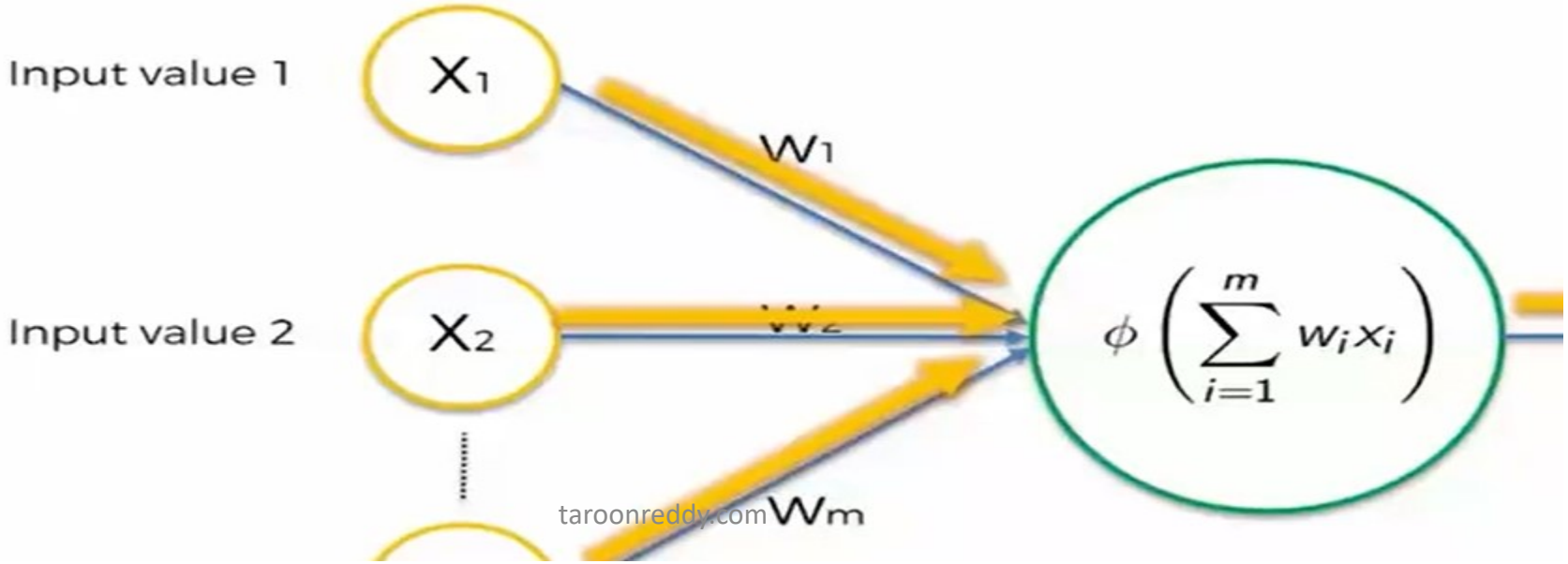
How do Neural Networ



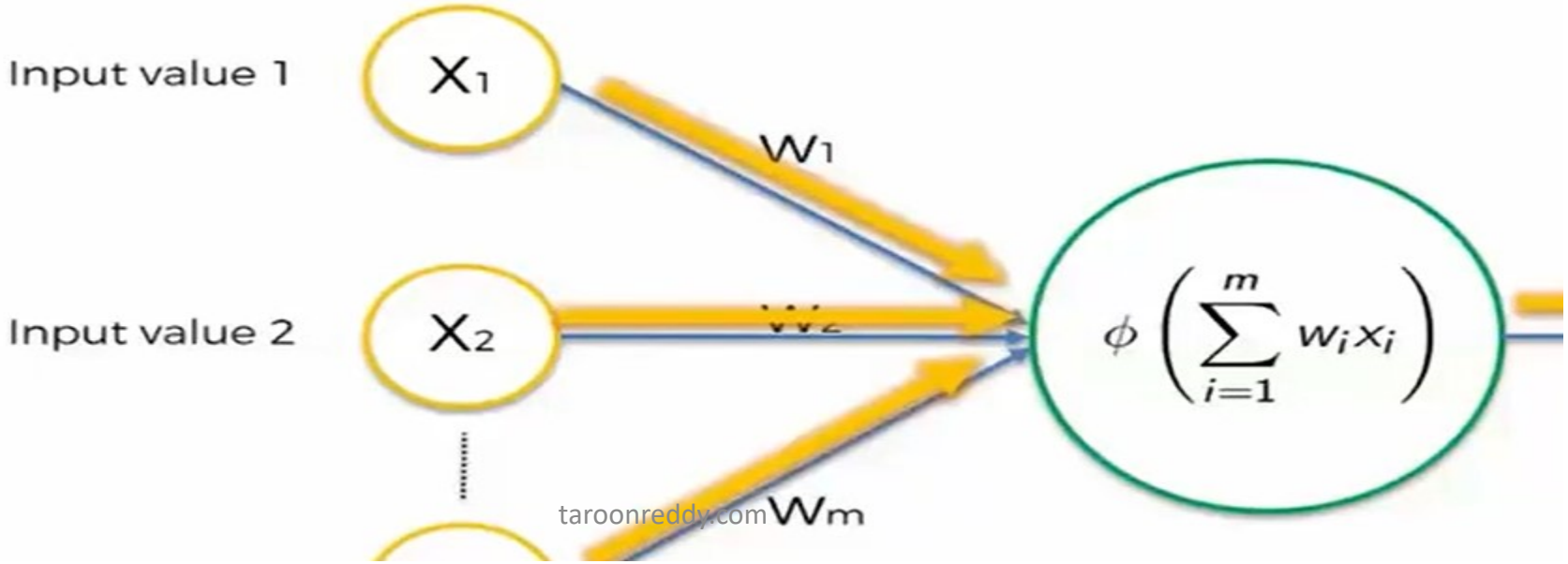
How do Neural Networ



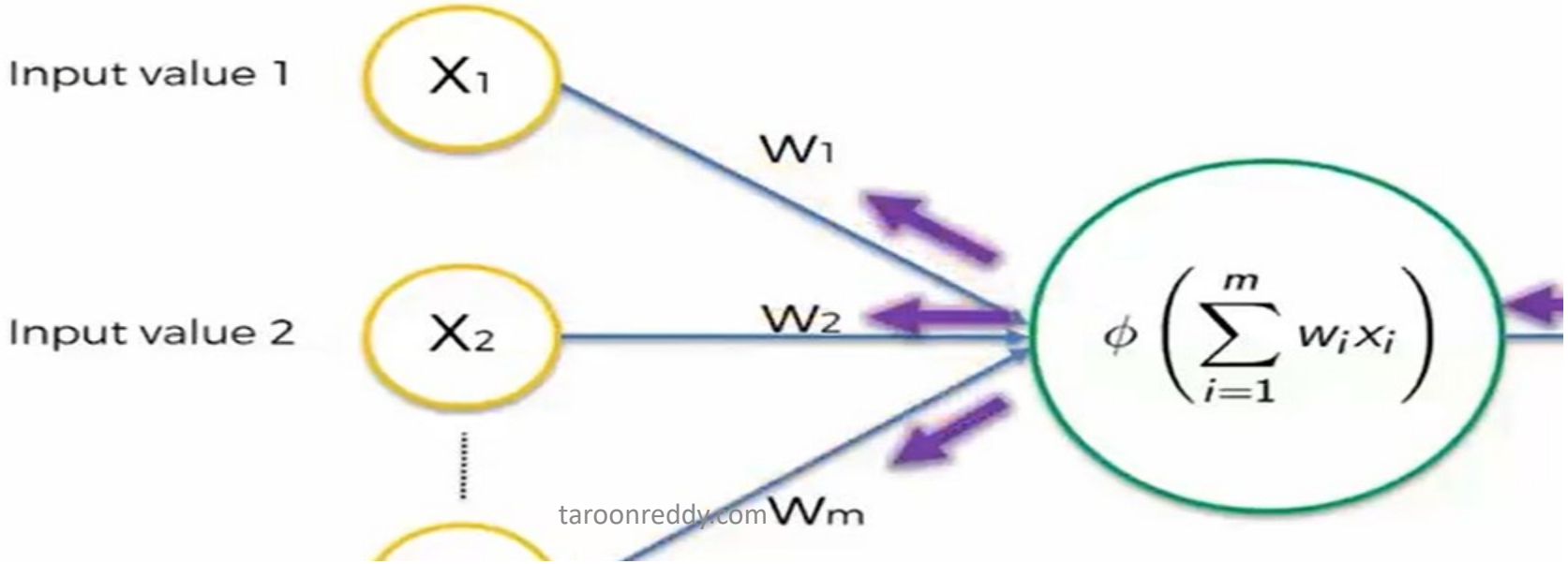
How do Neural Networ



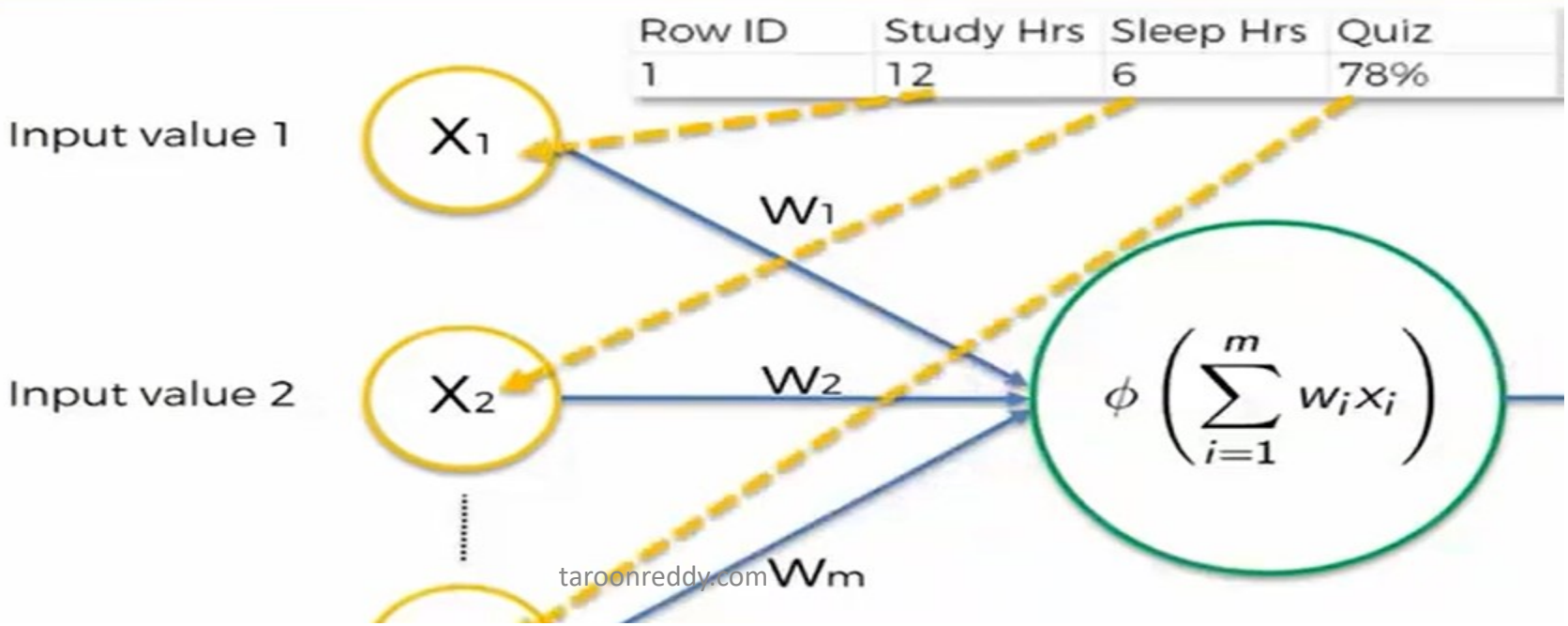
How do Neural Networ



How do Neural Networ

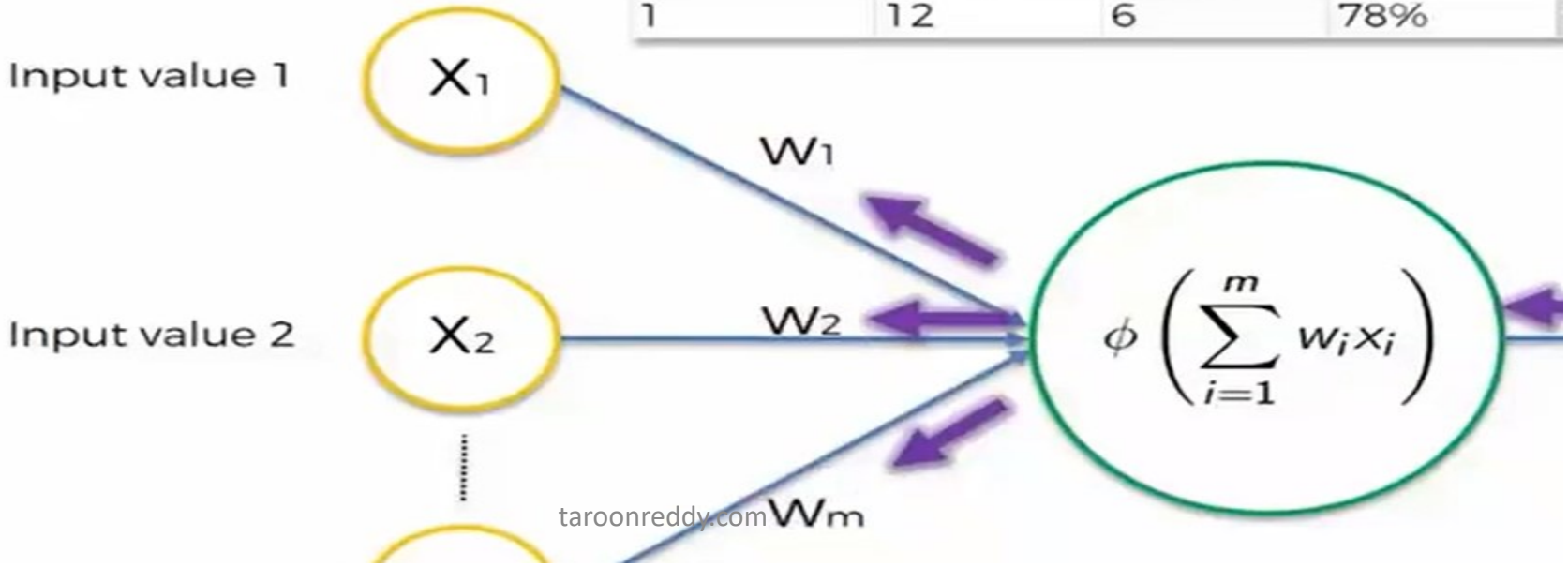


How do Neural Networ



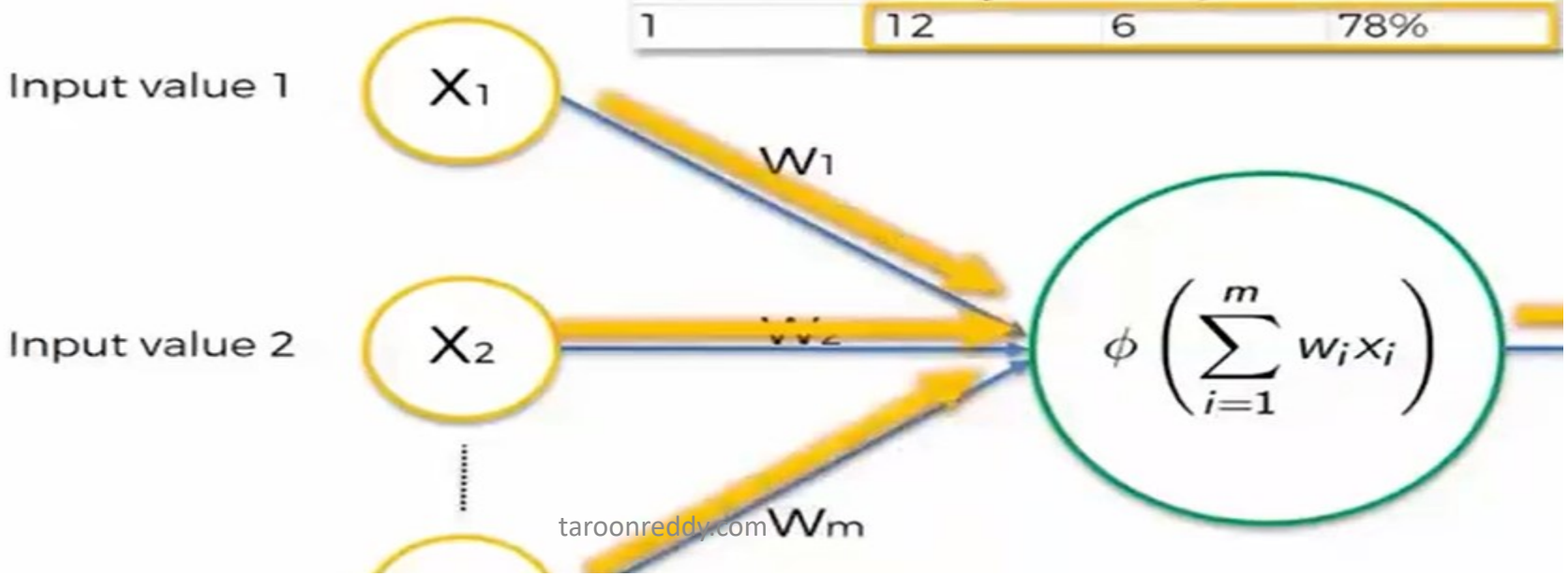
How do Neural Networ

Row ID	Study Hrs	Sleep Hrs	Quiz
1	12	6	78%



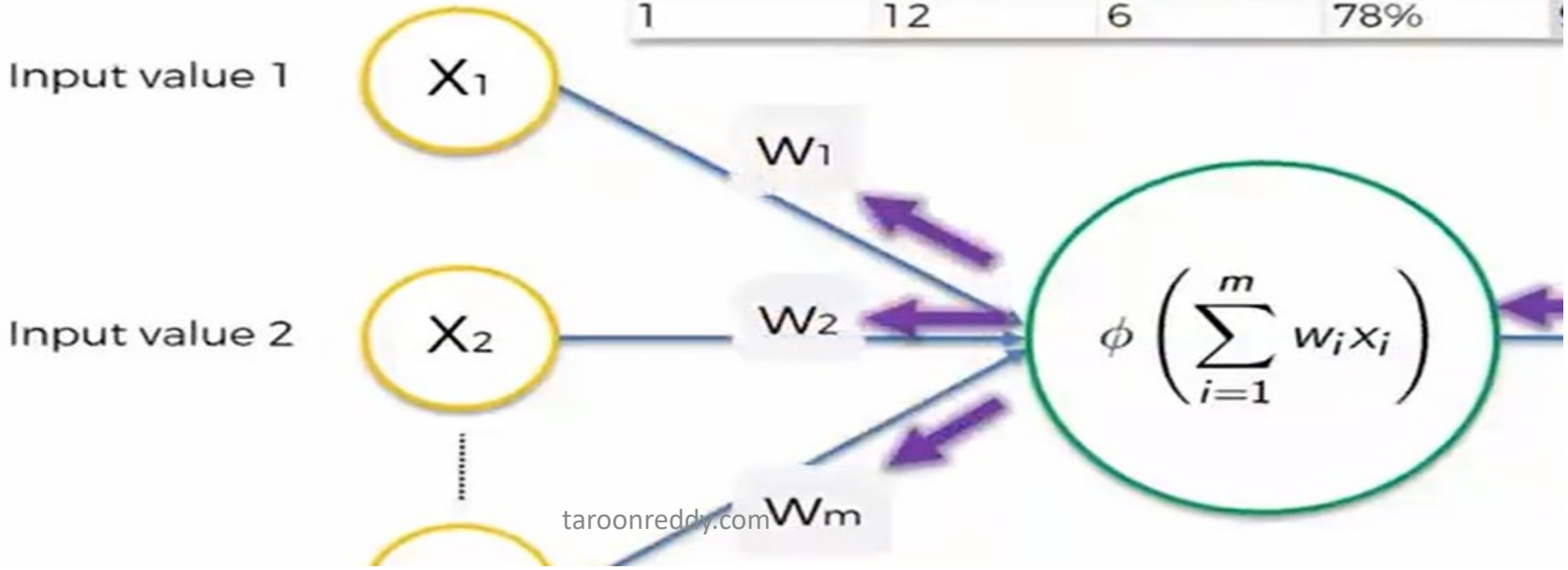
How do Neural Networ

Row ID	Study Hrs	Sleep Hrs	Quiz
1	12	6	78%



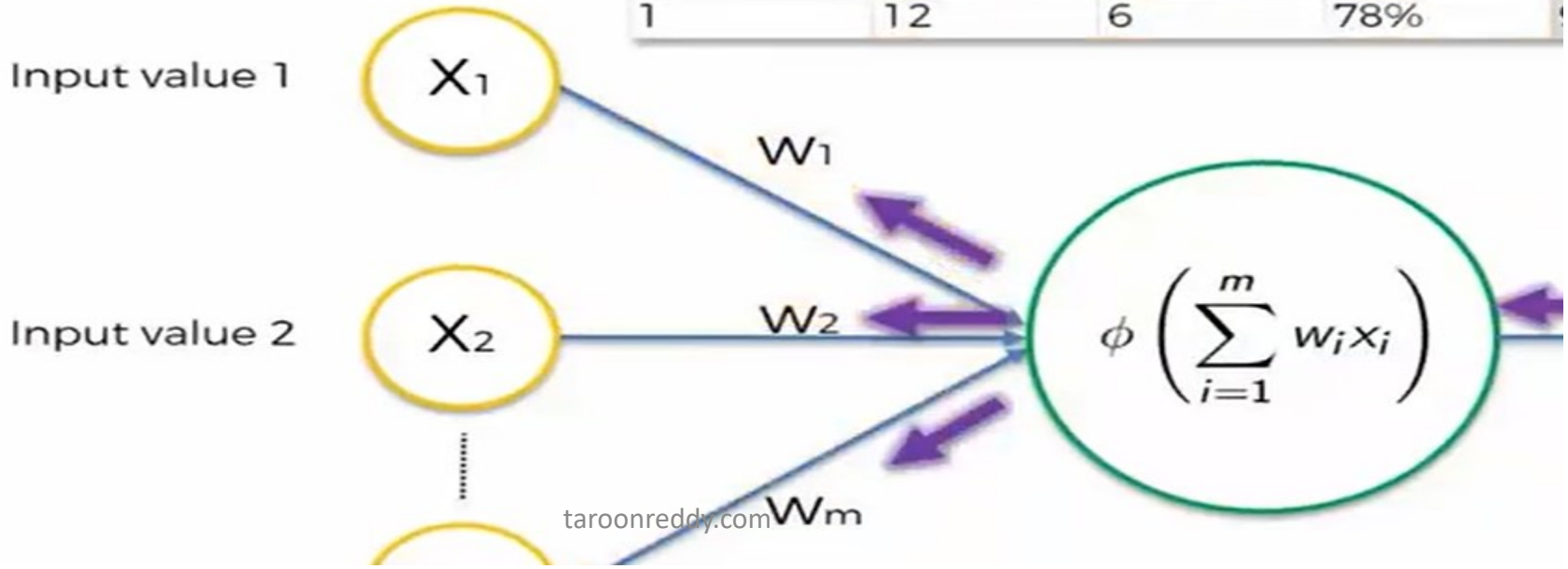
How do Neural Networ

Row ID	Study Hrs	Sleep Hrs	Quiz
1	12	6	78%



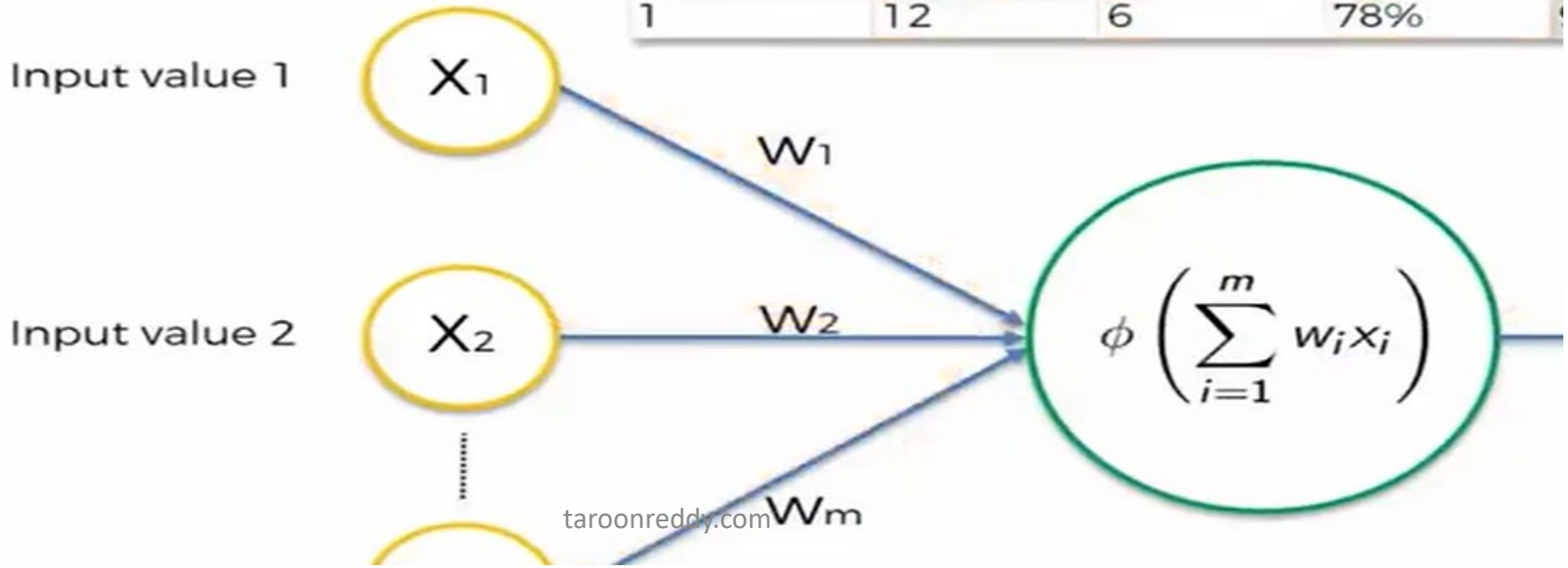
How do Neural Networ

Row ID	Study Hrs	Sleep Hrs	Quiz
1	12	6	78%



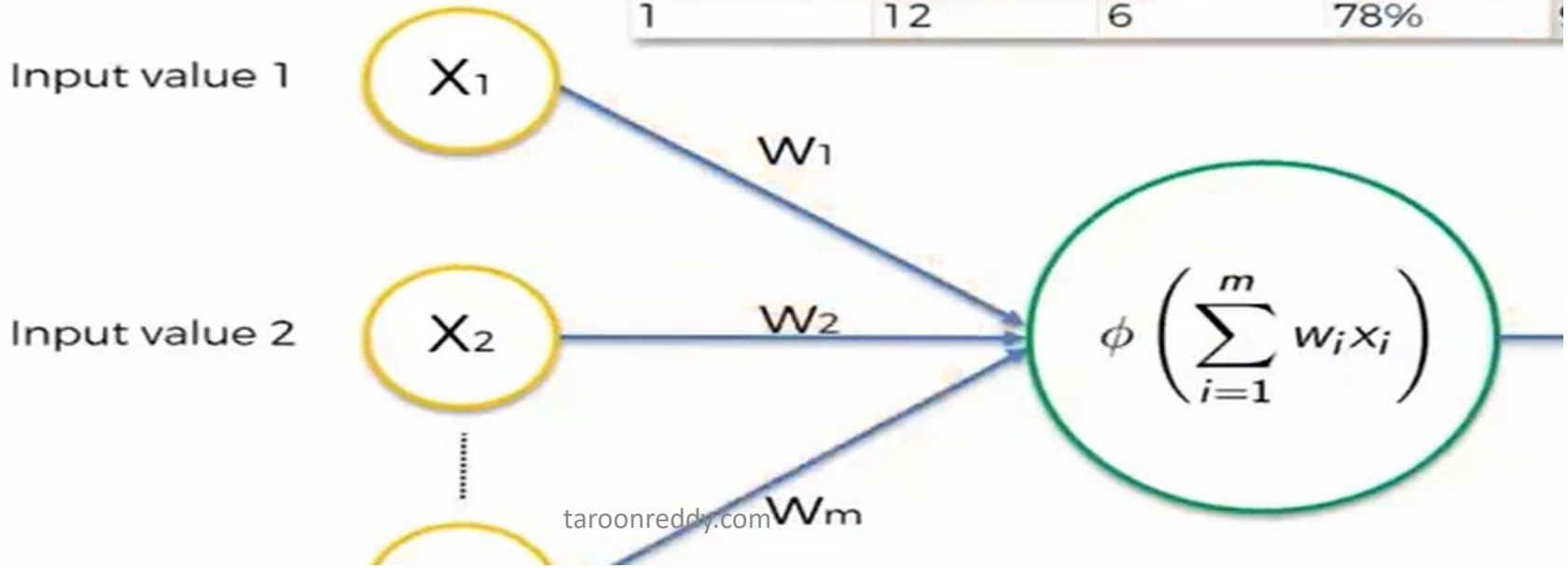
How do Neural Networ

Row ID	Study Hrs	Sleep Hrs	Quiz
1	12	6	78%



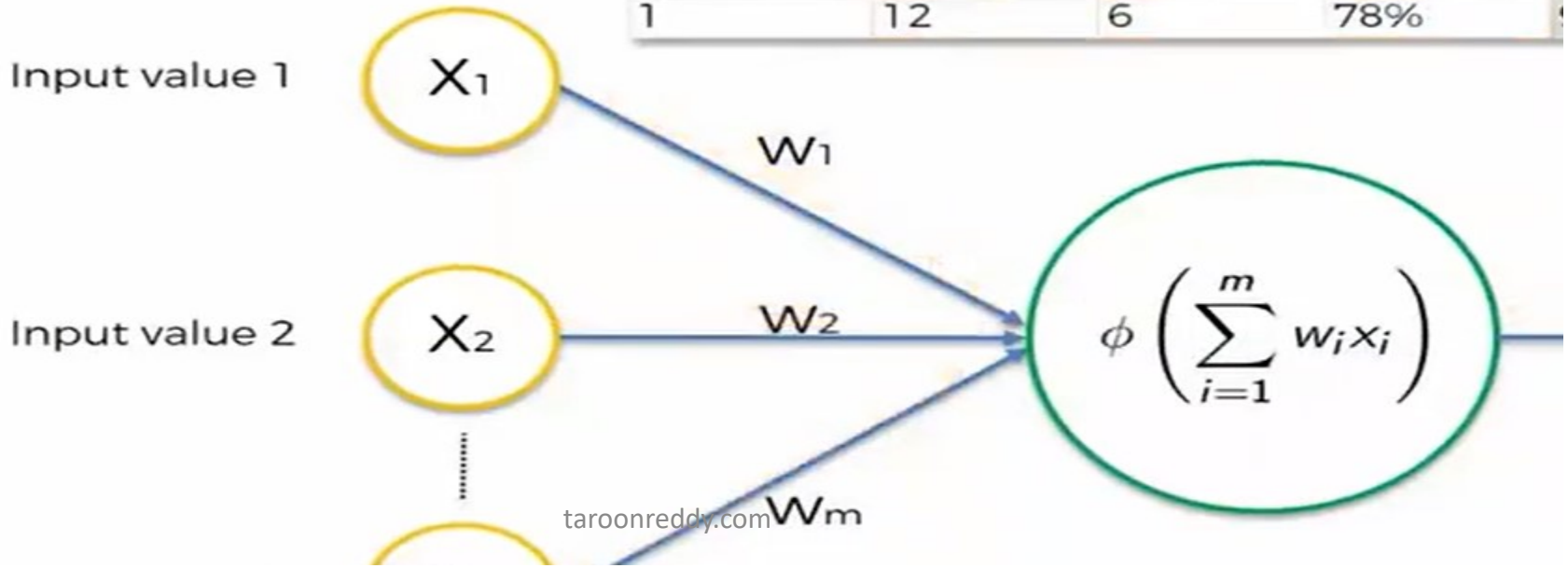
How do Neural Networ

Row ID	Study Hrs	Sleep Hrs	Quiz
1	12	6	78%

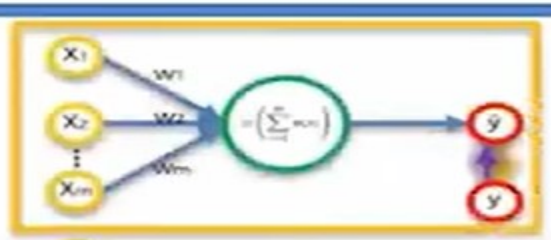


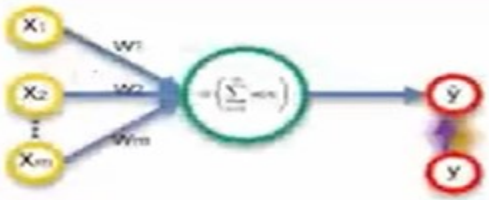
How do Neural Networ

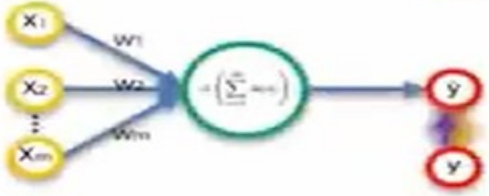
Row ID	Study Hrs	Sleep Hrs	Quiz
1	12	6	78%

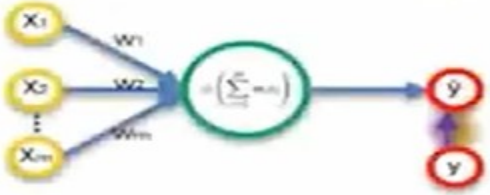


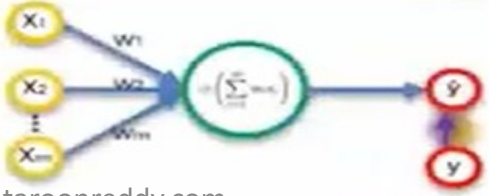
How do Neural Networ





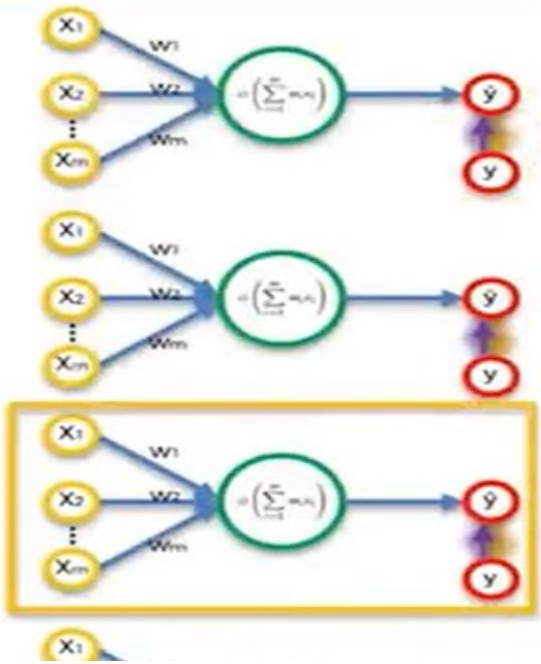


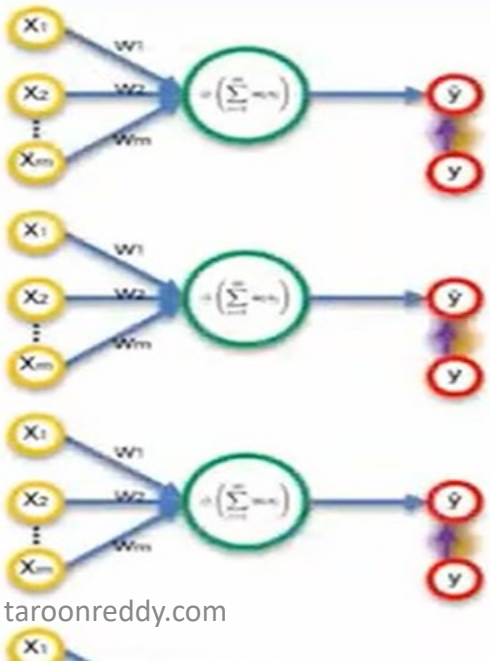




Row ID	Study Hrs	Sle
1	12	6
2	22	6.5
3	115	4
4	31	9
5	0	10
6	5	8
7	92	6
8	57	8

How do Neural Networ

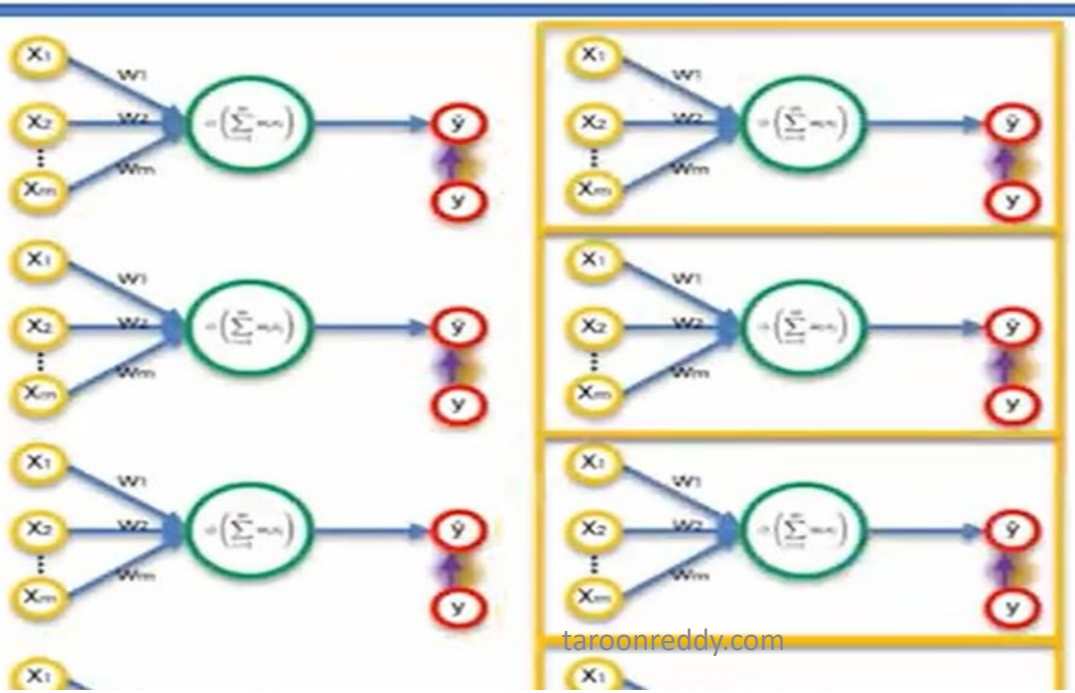




Row ID	Study Hrs	Sle
1	12	6
2	22	6.5
3	115	4
4	31	9
5	0	10
6	5	8
7	92	6
8	57	8

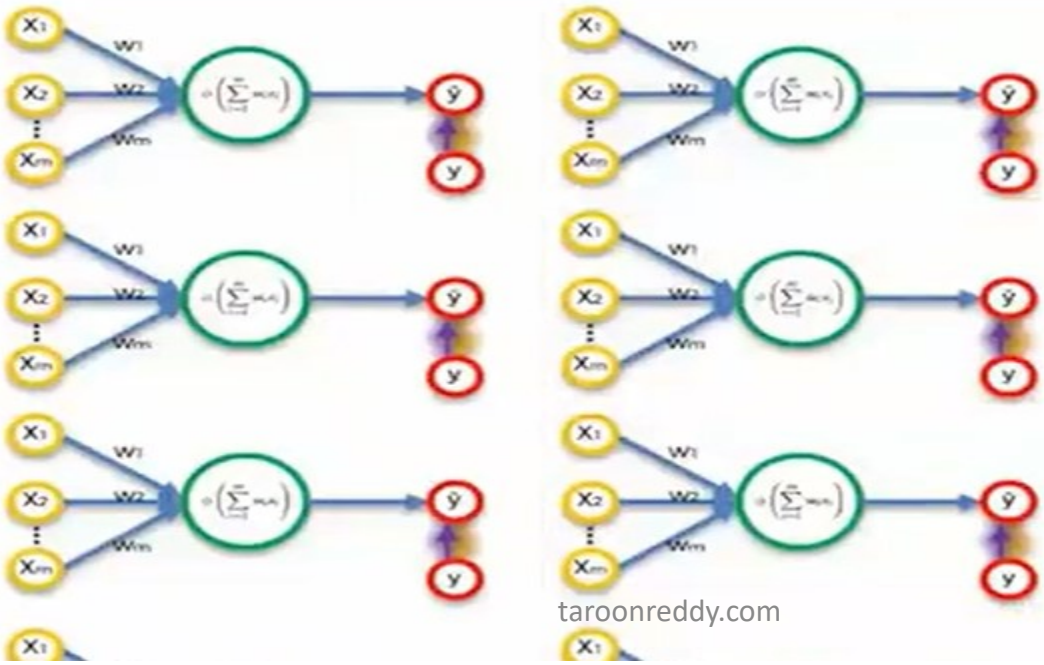
taroonreddy.com

How do Neural Networ



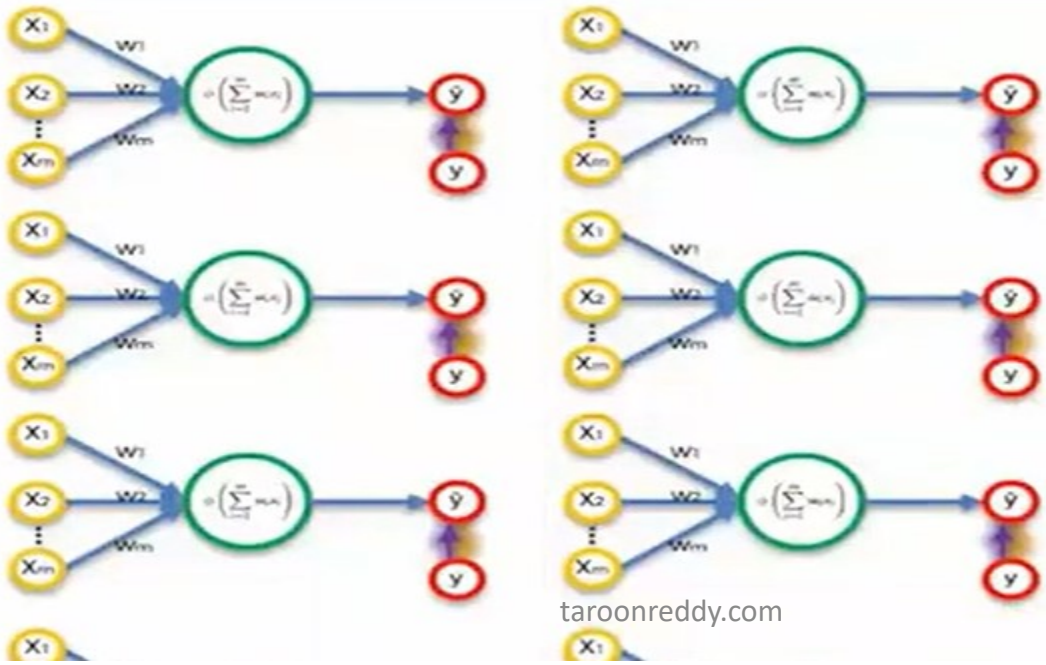
Row ID	Study Hrs	Sle
1	12	6
2	22	6.5
3	115	4
4	31	9
5	0	10
6	5	8
7	92	6
8	57	8

How do Neural Networ



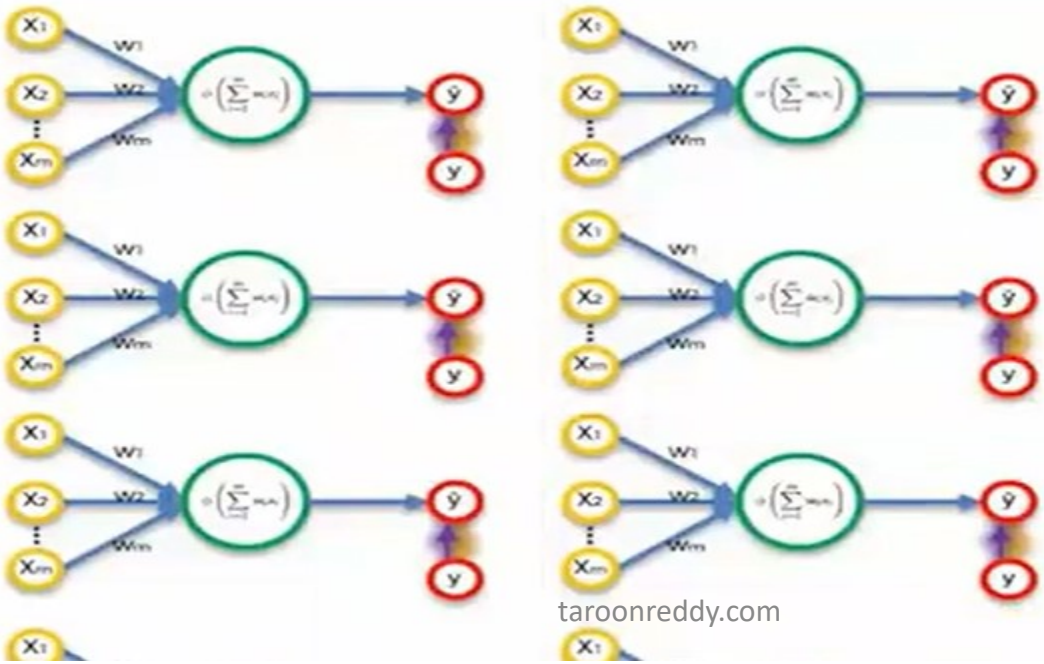
Row ID	Study Hrs	Sle
1	12	6
2	22	6.5
3	115	4
4	31	9
5	0	10
6	5	8
7	92	6
8	57	8

How do Neural Networ



Row ID	Study Hrs	Sle
1	12	6
2	22	6.5
3	115	4
4	31	9
5	0	10
6	5	8
7	92	6
8	57	8

How do Neural Networ



Row ID	Study Hrs	Sle
1	12	6
2	22	6.5
3	115	4
4	31	9
5	0	10
6	5	8
7	92	6
8	57	8

Training the ANN with Stochasti

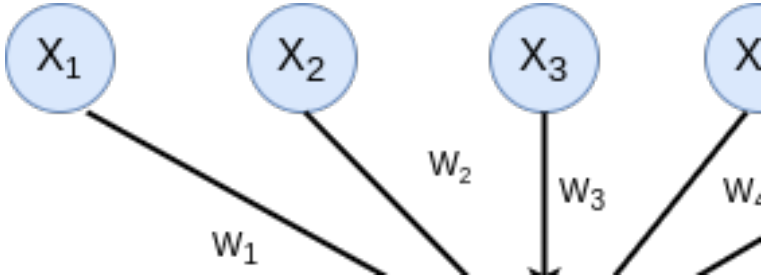
STEP 1: Randomly initialise the weights to small numbers close to 0

STEP 2: Input the first observation of your dataset in the input layer,

STEP 3: Forward-Propagation: from left to right, the neurons are act
neuron's activation is limited by the weights. Propagate the activati

STEP 4: Compare the predicted result to the actual result. Measure

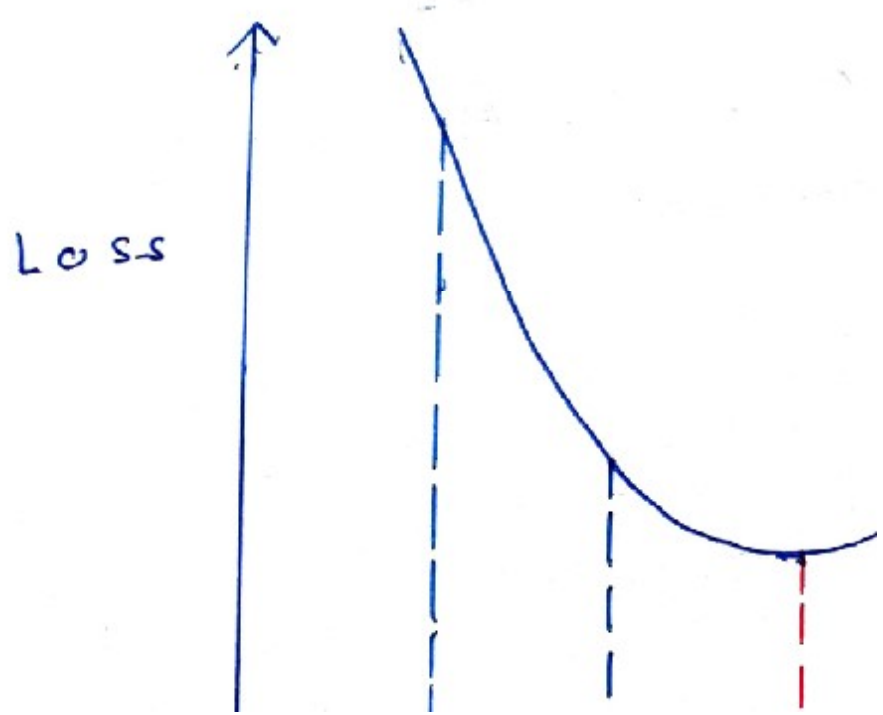
STEP 5: Back-Propagation: from right to left, the error is back-propa
how much they are responsible for the error. The learning rate deci



Here w1,w2, w3 are the weights of there corresponding features like x1,x2, x3 and b is a constant called the bias. Its importance is that it gives flexibility. So, using such an equation the machine tries to predict a value y which may be a value we need like the price of the house. Now, the machine tries to perfect its prediction by tweaking these weights. It does so, by comparing the predicted value y with the actual value of the example in our training set and using a function of their differences. This function is called a loss function. The machine tries to decrease this loss function or the error, i.e tries to get the prediction value close to the actual value.

Gradient Descent

This method is the key to minimizing the loss function and achieving our target, which is to predict close to the original value.



taroonreddy.com

In this diagram, above we see our loss function graph. If we observe we will see it is basically a parabolic shape or a convex shape, it has a specific global minimum which we need to find in order to find the minimum loss function value. So, we always try to use a loss function which is convex in shape in order to get a proper minimum. Now, we see the predicted results depend on the weights from the equation. If we replace equation 1 in equation 2 we obtain this graph, with weights in X-axis and Loss on Y-axis.

Initially, the model assigns random weights to the features. So, say it initializes the weight=a. So, we can see it generates a loss which is far from the minimum point L-min. Now, we can see that if we move the weights more towards the positive x-axis we can optimize the loss function and achieve minimum value. But, how will the machine know? We need to optimize weight to minimize error, so, obviously, we need to check how the error varies with the weights. To do this we need to find the derivative of the Error with respect to the weight. **This derivative is called Gradient.**

Gradient = dE/dw
Where E is the error and w is the weight.

Let's see how this works. Say, **if the loss increases with an increase in weight so Gradient will be positive,** So we are basically at the point C, where we can see this statement is true. **If loss decreases with an increase in weight so gradient will be negative**. We can see point A, corresponds to such a situation. Now, from point A we need to move towards positive x-axis and the gradient is negative. From point C, we need to move towards negative x-axis but the gradient is positive. **So, always the negative of the Gradient shows the directions along which the weights should be moved in order to optimize the loss function.** So, this way the gradient guides the model whether to increase or decrease weights in order to optimize the loss function.

The model found which way to move, now the model needs to find by how much it should move the weights. This is decided by a parameter called **Learning Rate denoted by Alpha**. the diagram we see, the weights are moved from point A to point B which are at a distance of *dx.*

dx = alpha * |dE/dw|
So, the distance to move is the product of learning rate parameter alpha and the magnitude of change in error with a change in weight at that point.
Now, we need to decide the Learning Rate very carefully. If it is very large the values of weights will be changed with a great amount and it would overstep the optimal value. If it is very low it takes tiny steps and takes a lot of steps to optimize. The updated weights are changed according to the following formula.

w=w — alpha * |dE/dw|
where w is the previous weight.
With each epoch, the model moves the weights according to the gradient to find the best weights.

Now, this is a loss optimization for a particular example in our training dataset. Our dataset contains thousands of such examples, so it will take a huge time to find optimal weights for all. Experiments have shown that if we optimize on only one sample of our training set, the weight optimization is good enough for the whole dataset. So, depending upon the methods we have different types of gradient descent mechanisms.

Gradient Descent Methods

Stochastic Gradient Descent: When we train the model to optimize the loss function using only one particular example from our dataset, it is called Stochastic Gradient Descent.

Batch Gradient Descent: When we train the model to optimize the loss function using the mean of all the individual losses in our whole dataset, it is called Batch Gradient Descent.

Mini-Batch Gradient Descent: Now, as we discussed batch gradient descent takes a lot of time and is therefore somewhat inefficient. If we look at SGD, it is trained using only 1 example. So, how good do you think a baby will learn if it is shown only one bike and told to learn about all other bikes? It's simple its decision will be somewhat biased to the peculiarities of the shown example. So, it is the same for the SGD, there is a possibility that the model may get too biased with the peculiarity of that particular example. So, we use the mean of a batch of 10–1000 examples to check the optimize the loss in order to deal with the problems.

Keras & Tensorflow

Artificial Neural Network