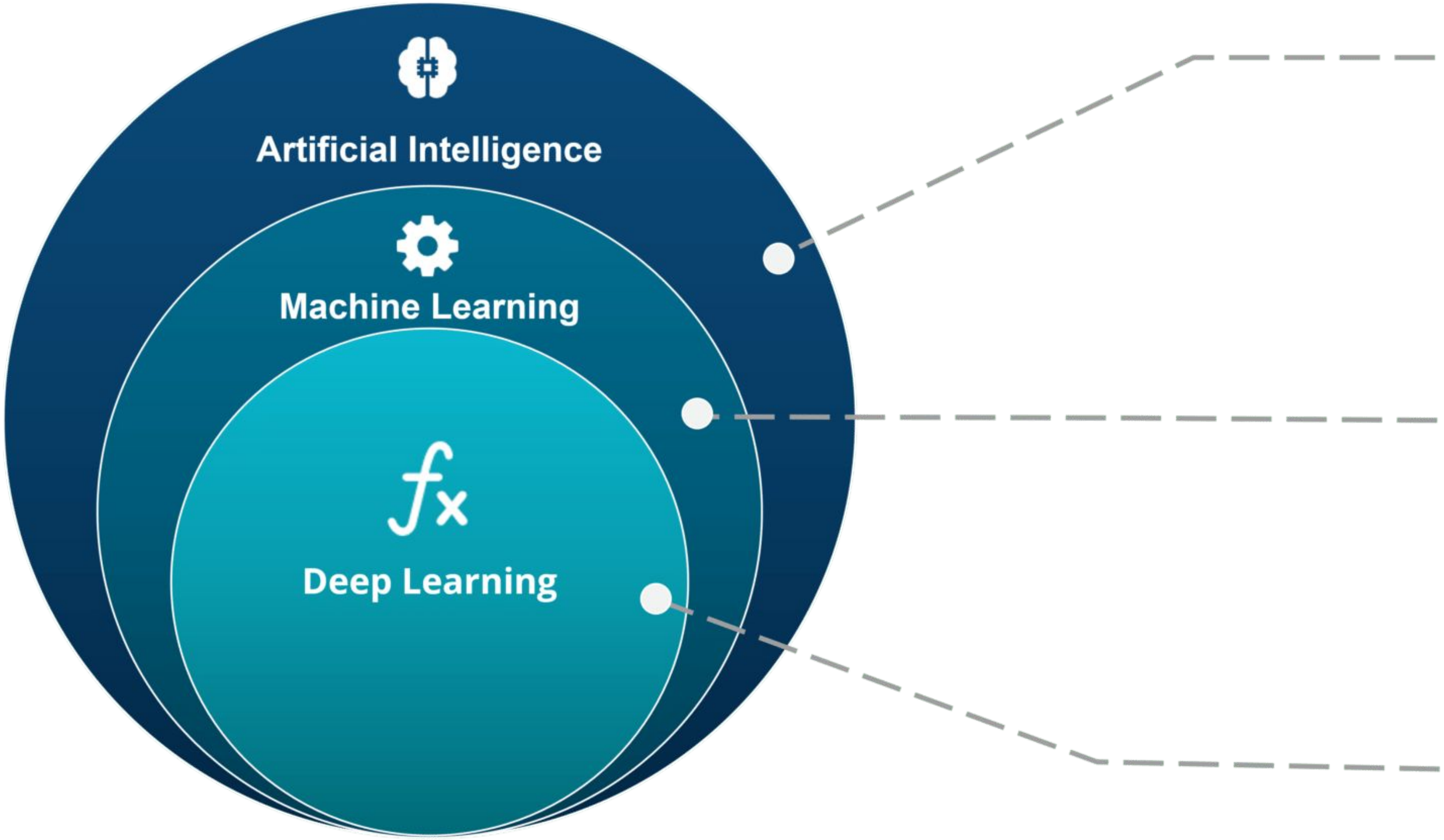
Deep Learning from Scratch

Theory + Practical

AI vs ML vs DL

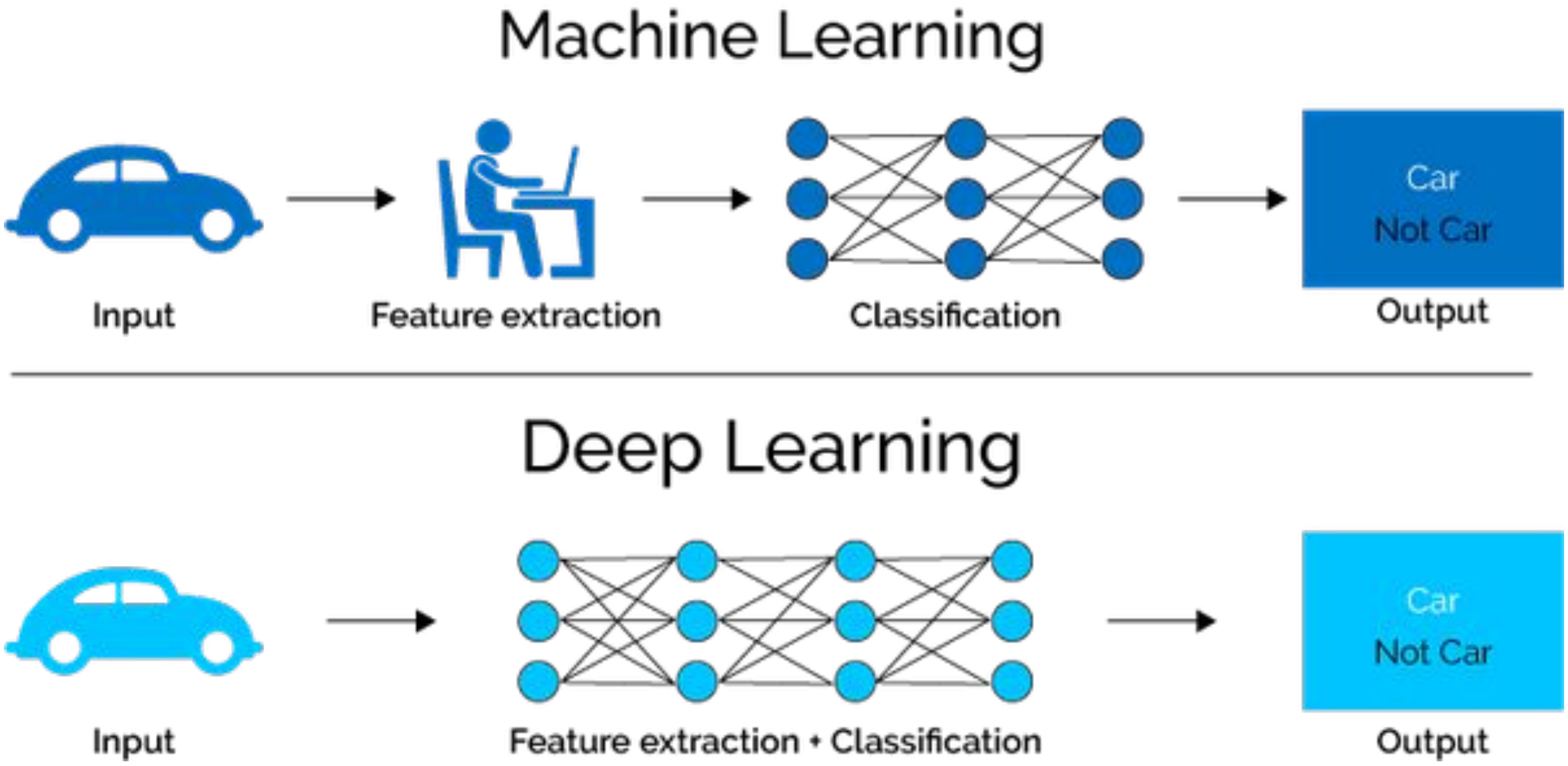


ARTIFICIAL INTELLIGENCE
A technique which enables machines to mimic human behaviour

MACHINE LEARNING
Subset of AI technique which use statistical methods to enable machines to improve with experience

DEEP LEARNING
Subset of ML which make the computation of multi-layer neural network feasible

ML vs DL



Machine Learning

Input → Feature extraction → Classification → Output

Car

Not Car

Deep Learning

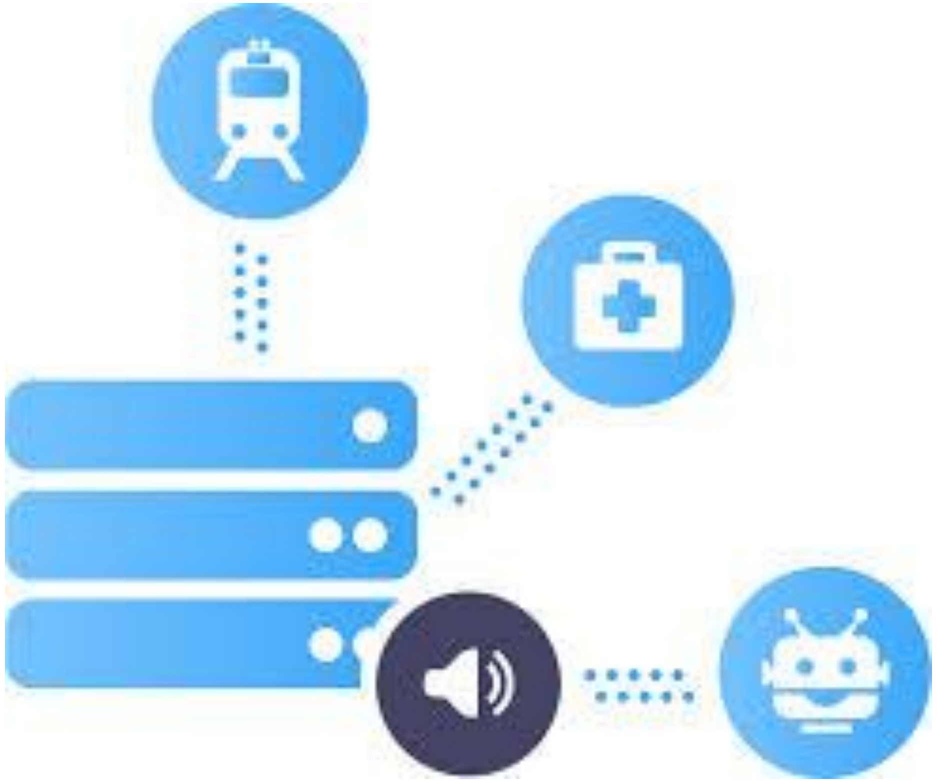
Input → Feature extraction + Classification → Output

Car

Not Car

So, What do you think, what is Deep Learning

Deep learning is a machine learning technique that learns features and task directly from the data, where data may by images, text or sound!



So, What we learn in this course

- Artificial Neural Network
- Convolutional neural network
- Recurrent neural network
- Boltzmann machine Vs Deep BM
- Self organizing maps
- Autoencoder
- GAN (Generative Adversarial Network)
- Deep Q Learning
- *Pre Train Model (CNN* **Architecture** *and many more…*

Keras

Vs

TensorFlow

Thank You!

Deep Learning from Scratch

Theory + Practical

Nidhi Chouhan

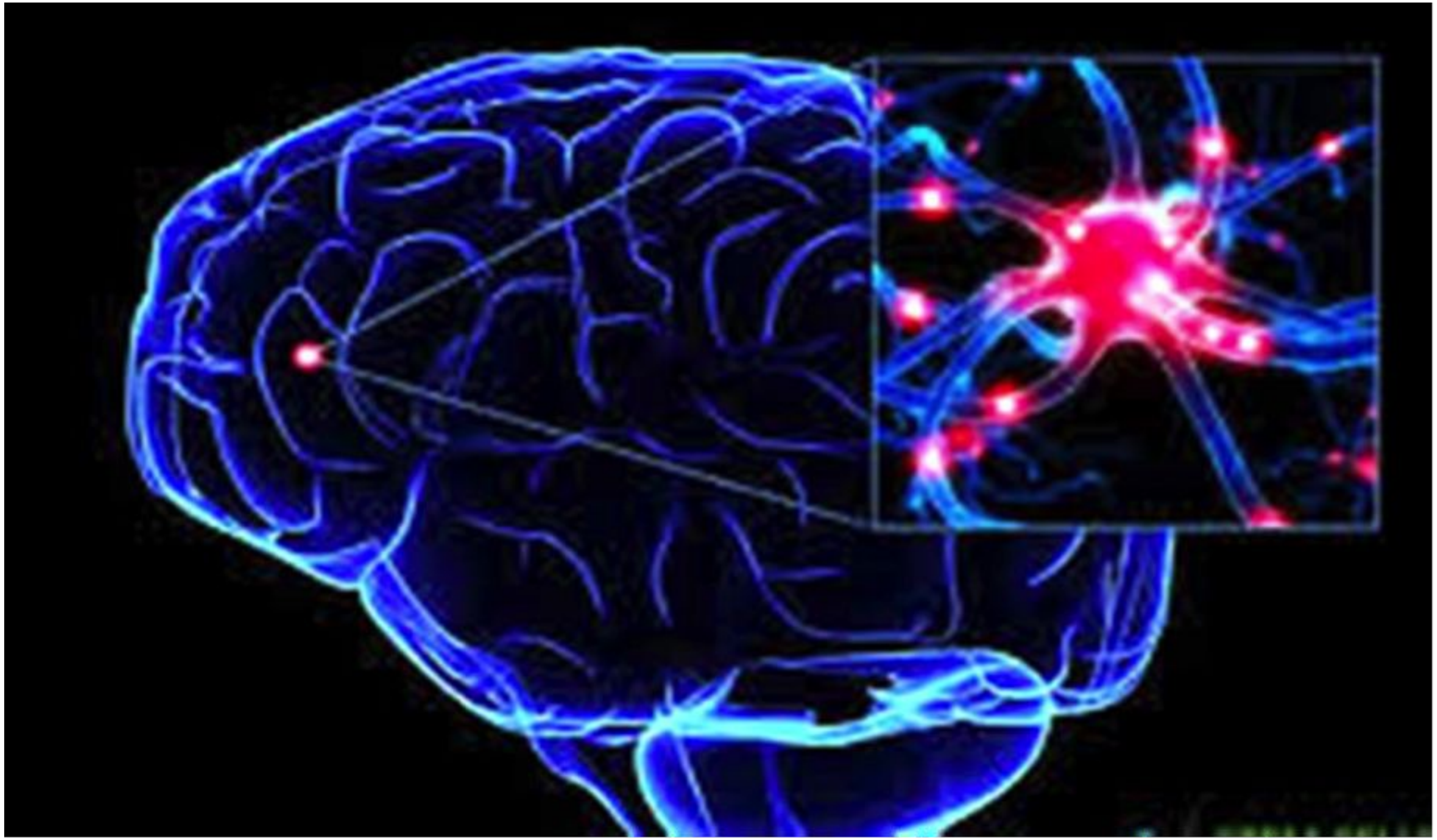
So, What we learn in this course

Supervised Unsupervised Reinforcement Learning

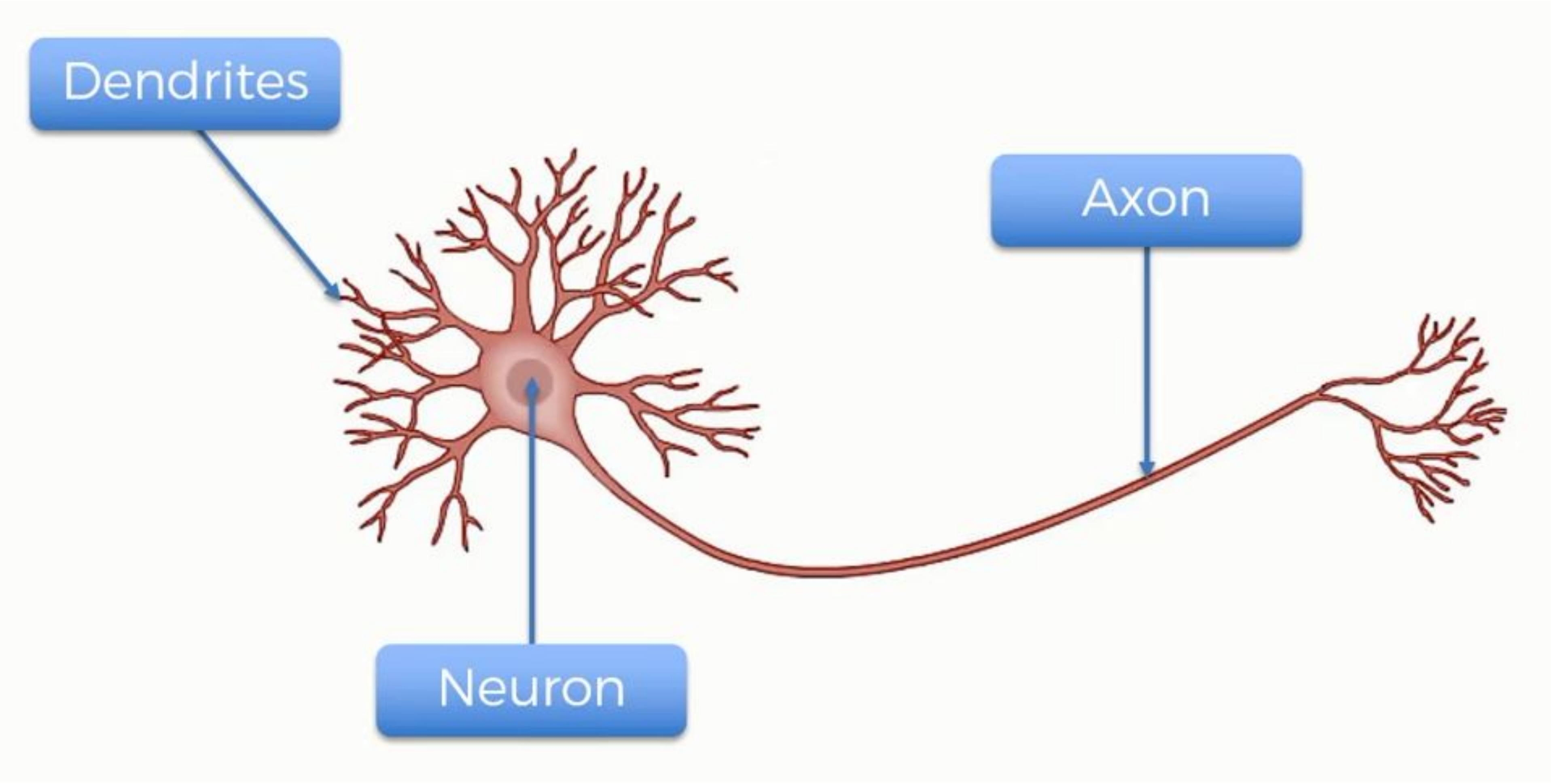
Artificial Neural Network

Regression and Classification

What is Neural



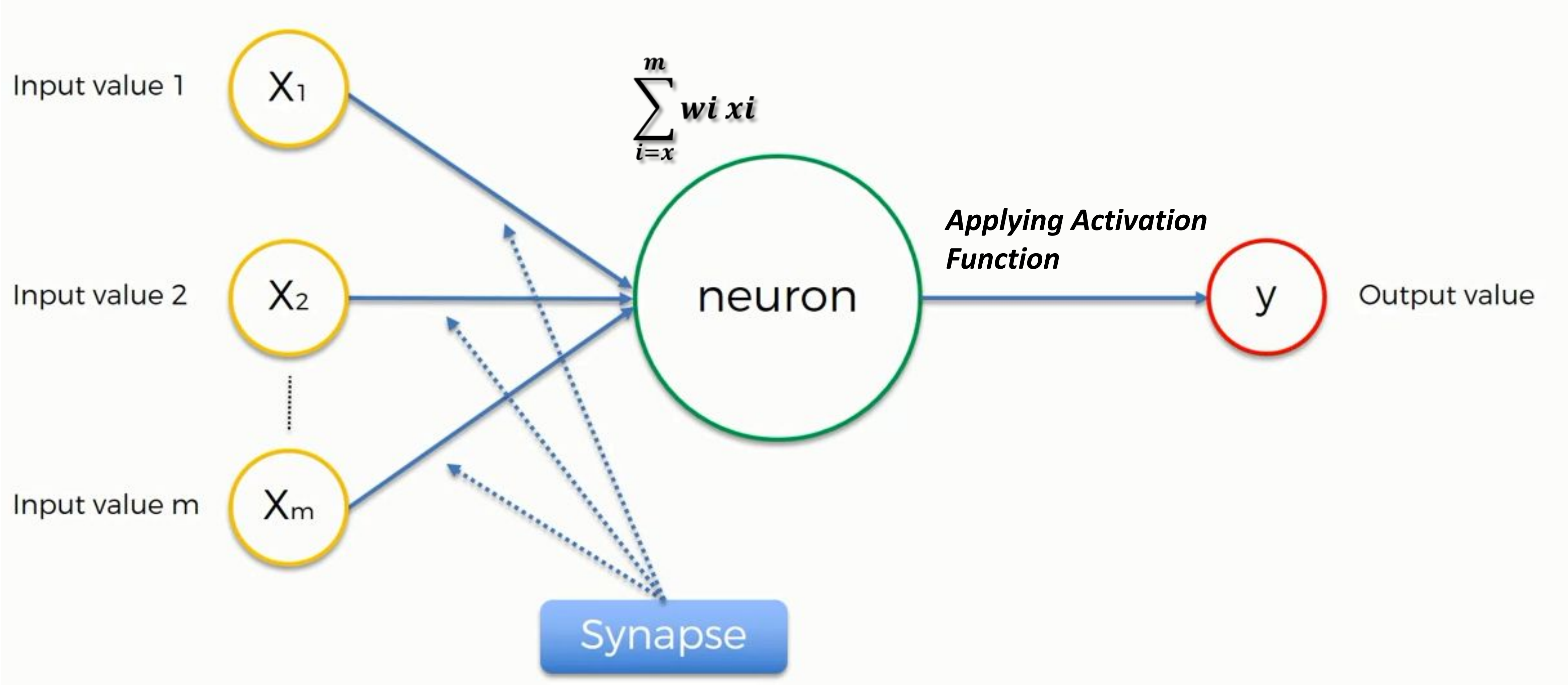
Neural Working



Perceptron in deep learning

Artificial Neural Network

Normalize/Standardize



$$\sum_{i=x}^{m} wi\ xi$$

Applying Activation Function

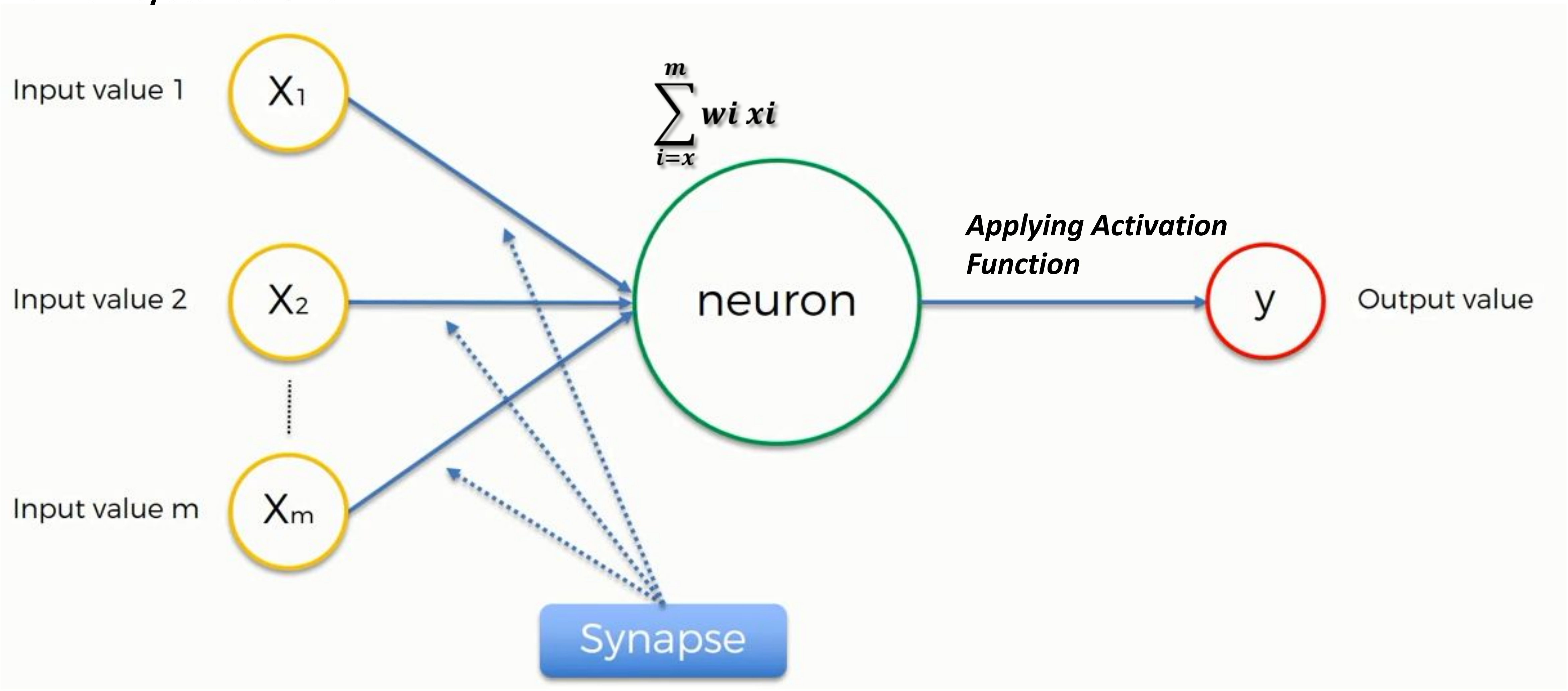
Deep Learning from Scratch

Theory + Practical

Nidhi chouhan

Artificial Neural Network

Normalize/Standardize



$$\sum_{i=x}^{m} wi\, xi$$

Applying Activation Function

What is an Activation Function?

Activation functions are an extremely important feature of the artificial neural networks. They basically decide whether a neuron should be activated or not. Whether the information that the neuron is receiving is relevant for the given information or should it be ignored.

$$Y = \text{Activation}(\Sigma(weight * input) + bias)$$

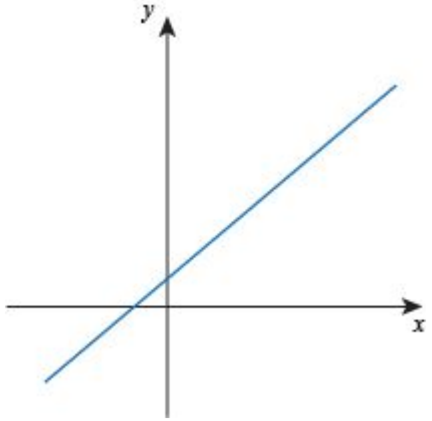
The activation function is the non linear transformation that we do over the input signal. This transformed output is then seen to the next layer of neurons as input.

- Linear **Activation Function**
- Non Linear **Activation Function**

What is an Activation Function?

Linear Function

The function is a line or linear. Therefore, the output of the functions will not be confined between any range



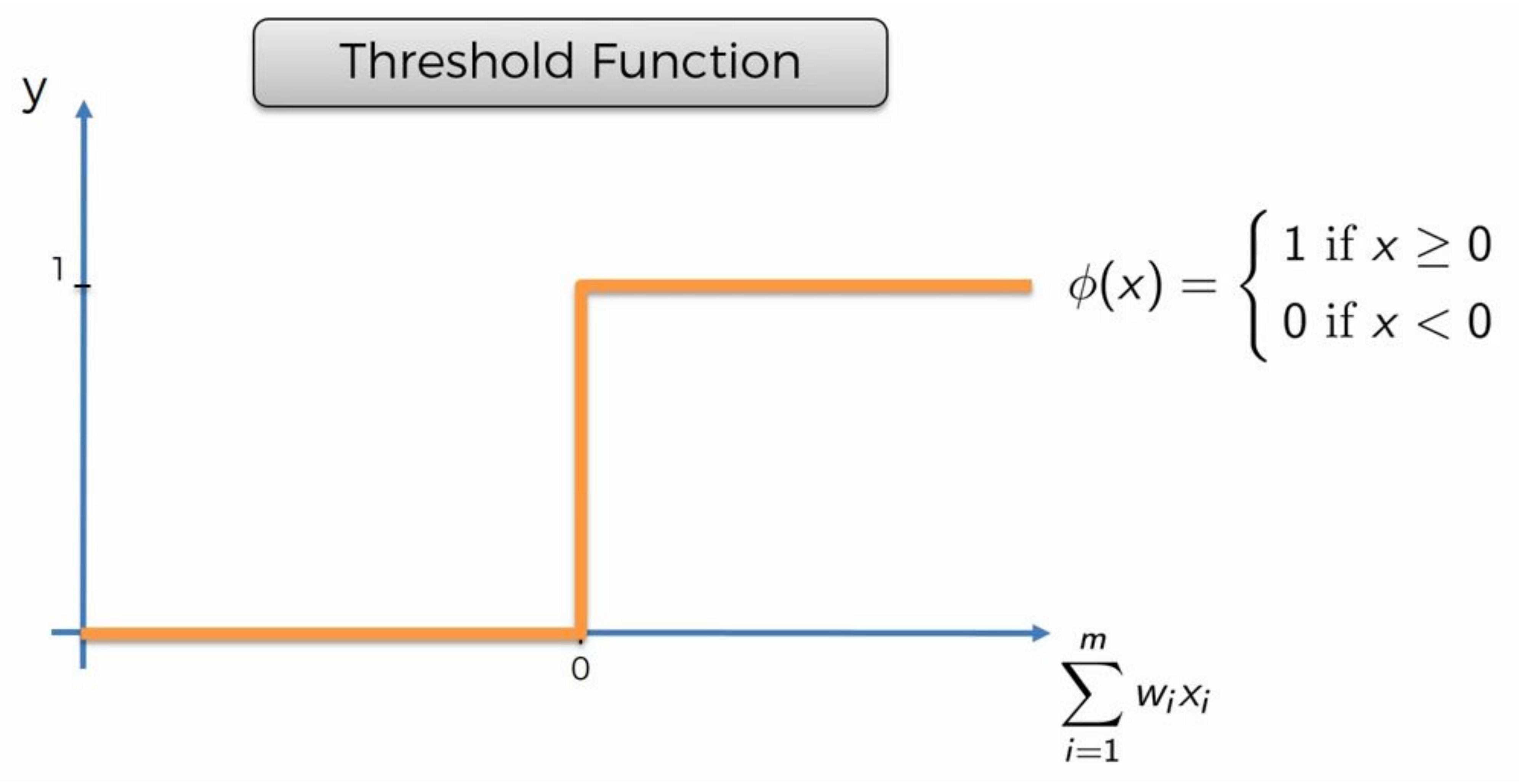
Non Linear Function

They make it easy for the model to generalize or adapt with variety of data and to differentiate between the output

The Nonlinear Activation Functions are mainly divided on the basis of their **range or curves**

1. Threshold
2. Sigmoid
3. Tanh
4. ReLU
5. Leaky ReLU
6. Softmax

Threshold Function?



$$\phi(x) = \begin{cases} 1 \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases}$$

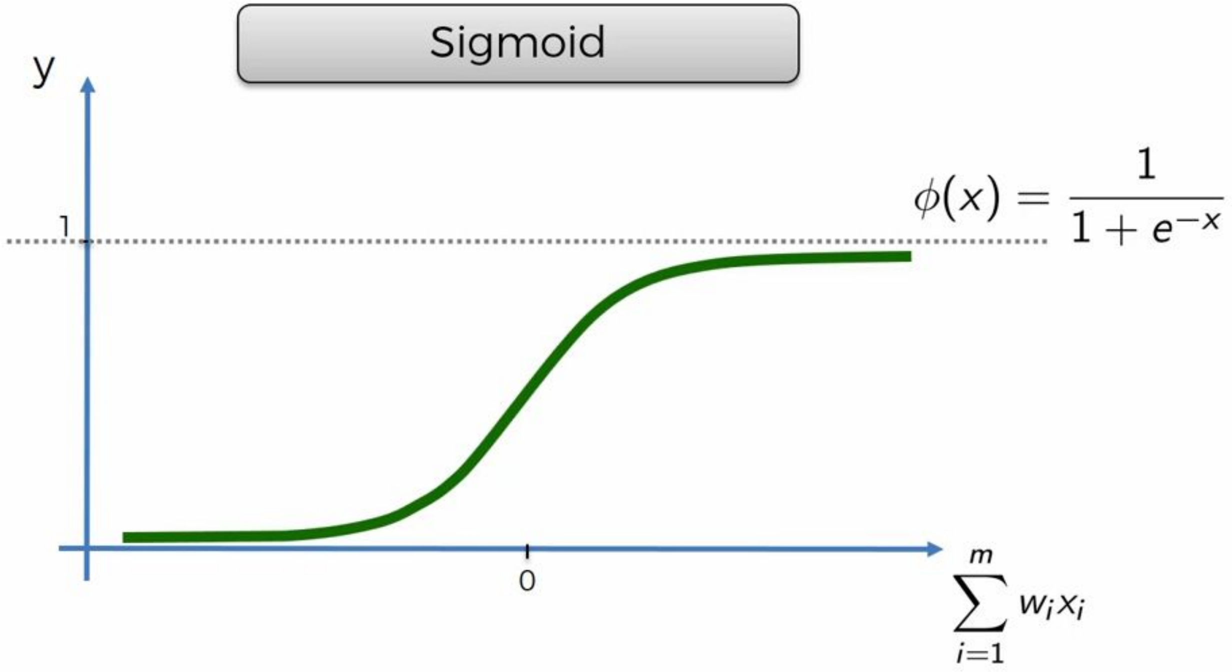
$$\sum_{i=1}^{m} w_i x_i$$

Sigmoid Function?

The Sigmoid Function curve looks like a S-shape

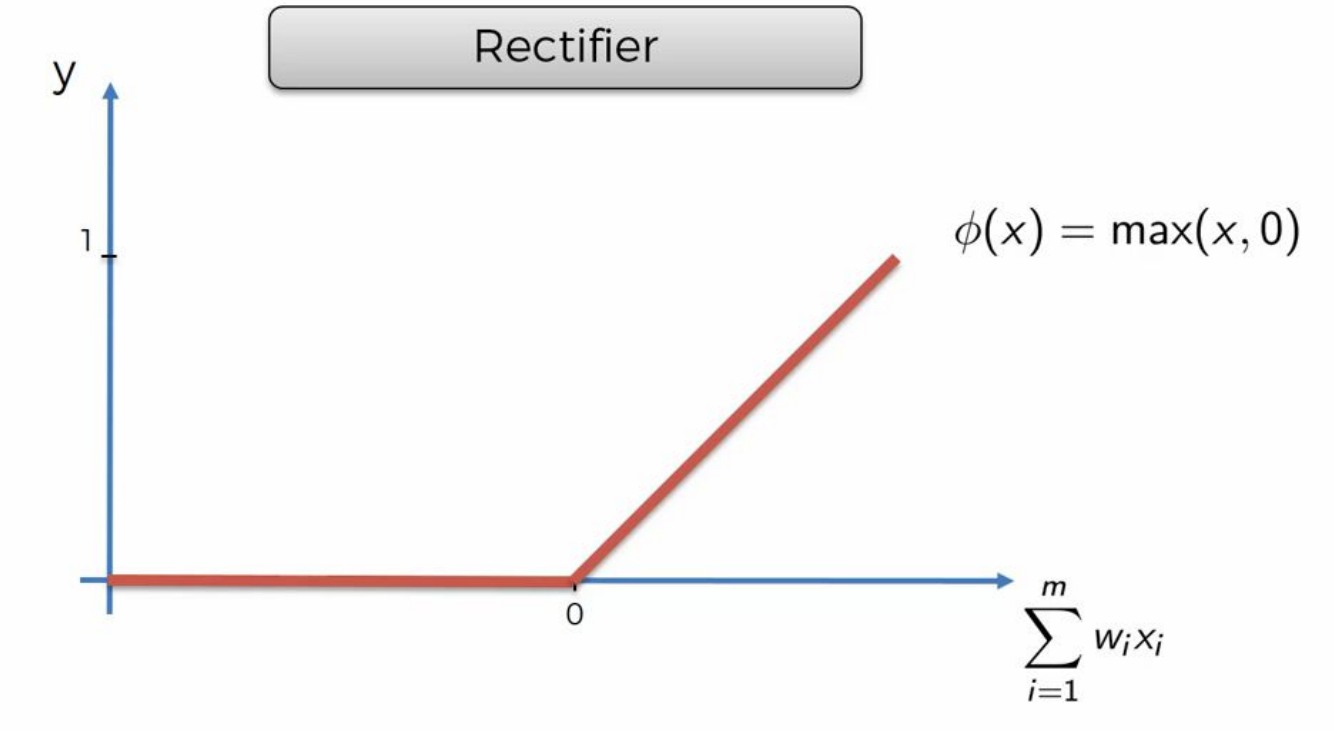
This function reduces extreme values or outliers in data without removing them.

It converts independent variables of near infinite range into simple probabilities between 0 and 1, and most of its output will be very close to 0 or 1.



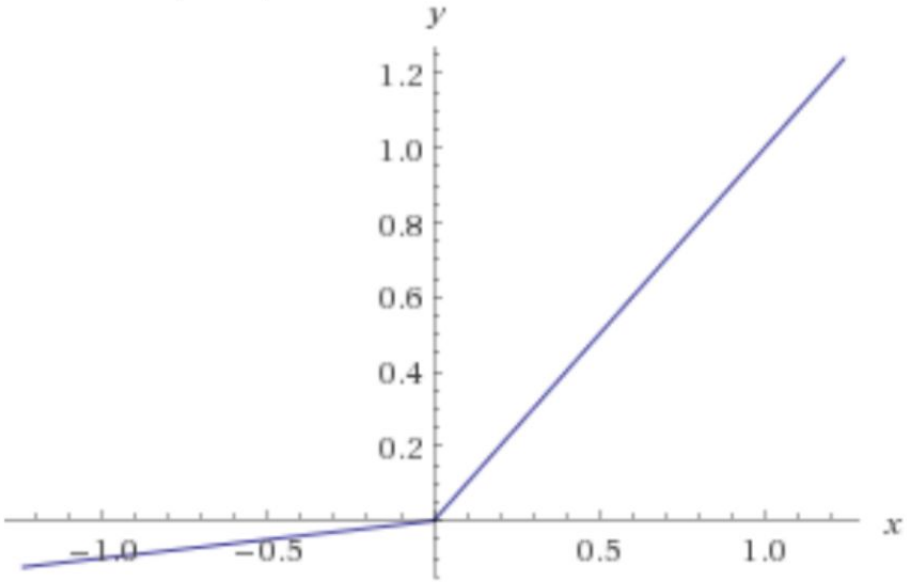
Rectifier (Relu) Function?

ReLU is the most widely used activation function while designing networks today. First things first, the ReLU function is non linear, which means we can easily backpropagate the errors and have multiple layers of neurons being activated by the ReLU function.



Leaky Relu Function?

Leaky ReLU function is nothing but an improved version of the ReLU function. As we saw that for the ReLU function, the gradient is 0 for x<0, which made the neurons die for activations in that region. Leaky ReLU is defined to address this problem. Instead of defining the Relu function as 0 for x less than 0, we define it as a small linear component of x.



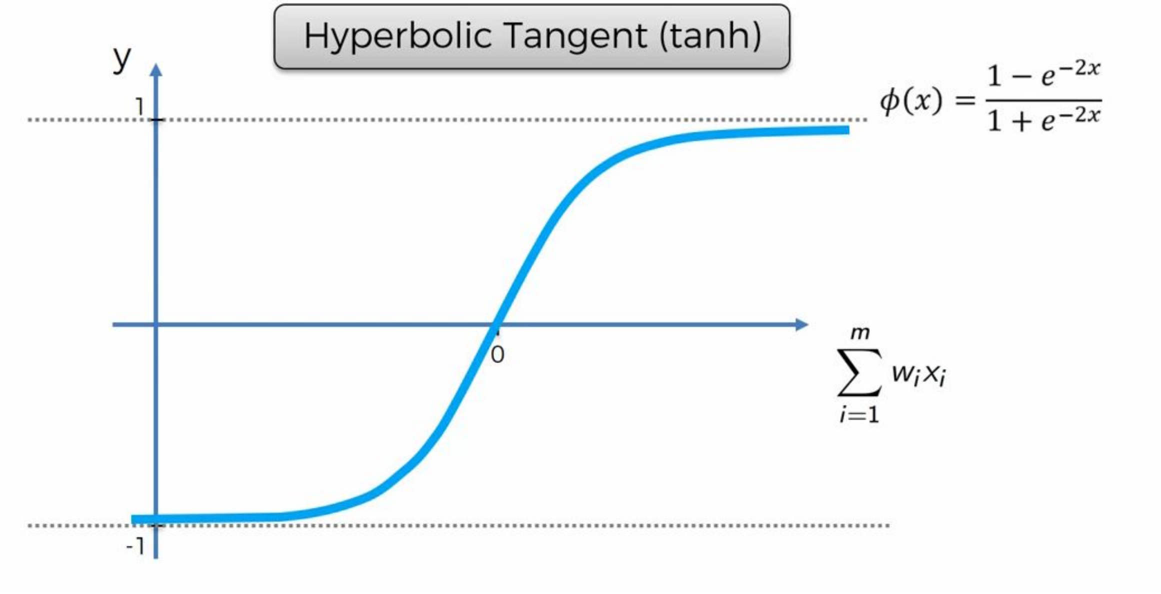
What we have done here is that we have simply replaced the horizontal line with a non-zero, non-horizontal line. Here a is a small value like 0.01 or so.

Hyperbolic tangent function?

Pronounced “*tanch*,” tanh is a hyperbolic trigonometric function

The tangent represents a ratio between the opposite and adjacent sides of a right triangle, tanh represents the ratio of the hyperbolic sine to the hyperbolic cosine: tanh(*x*) = sinh(*x*) / cosh(*x*)

Unlike the Sigmoid function, the normalized range of tanh is –1 to 1 The advantage of tanh is that it can deal more easily with negative numbers

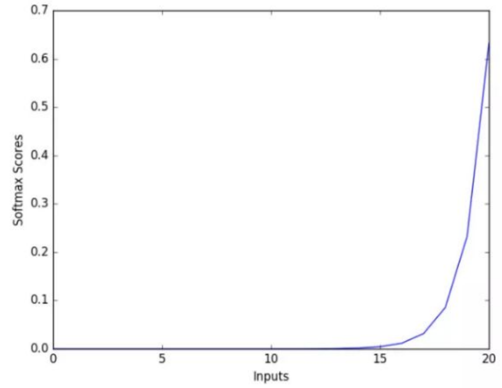


Softmax Function (for Multiple Classification)?

Softmax function calculates the probabilities distribution of the event over ‘n’ different events. In general way of saying, this function will calculate the probabilities of each target class over all possible target classes. Later the calculated probabilities will be helpful for determining the target class for the given inputs.

The main advantage of using Softmax is the output probabilities range. The range will 0 to 1, and the sum of all the probabilities will be equal to one. If the softmax function used for multi-classification model it returns the probabilities of each class and the target class will have the high probability.

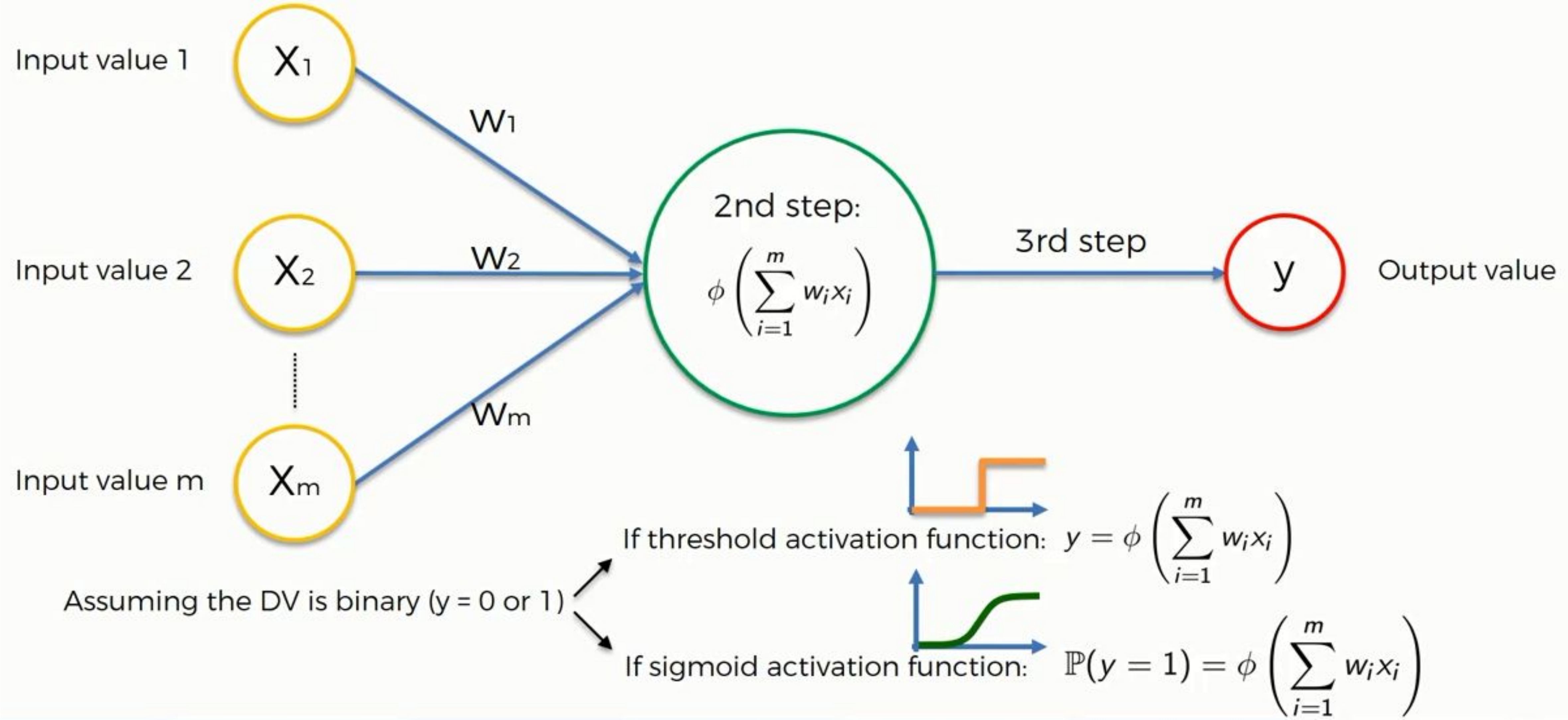
The formula computes the exponential (e-power) of the given input value and the sum of exponential values of all the values in the inputs. Then the ratio of the exponential of the input value and the sum of exponential values is the output of the softmax function.



Softmax Graph

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

Activation Function Example



Assuming the DV is binary (y = 0 or 1)

If threshold activation function: $y = \phi\left(\sum_{i=1}^{m} w_i x_i\right)$

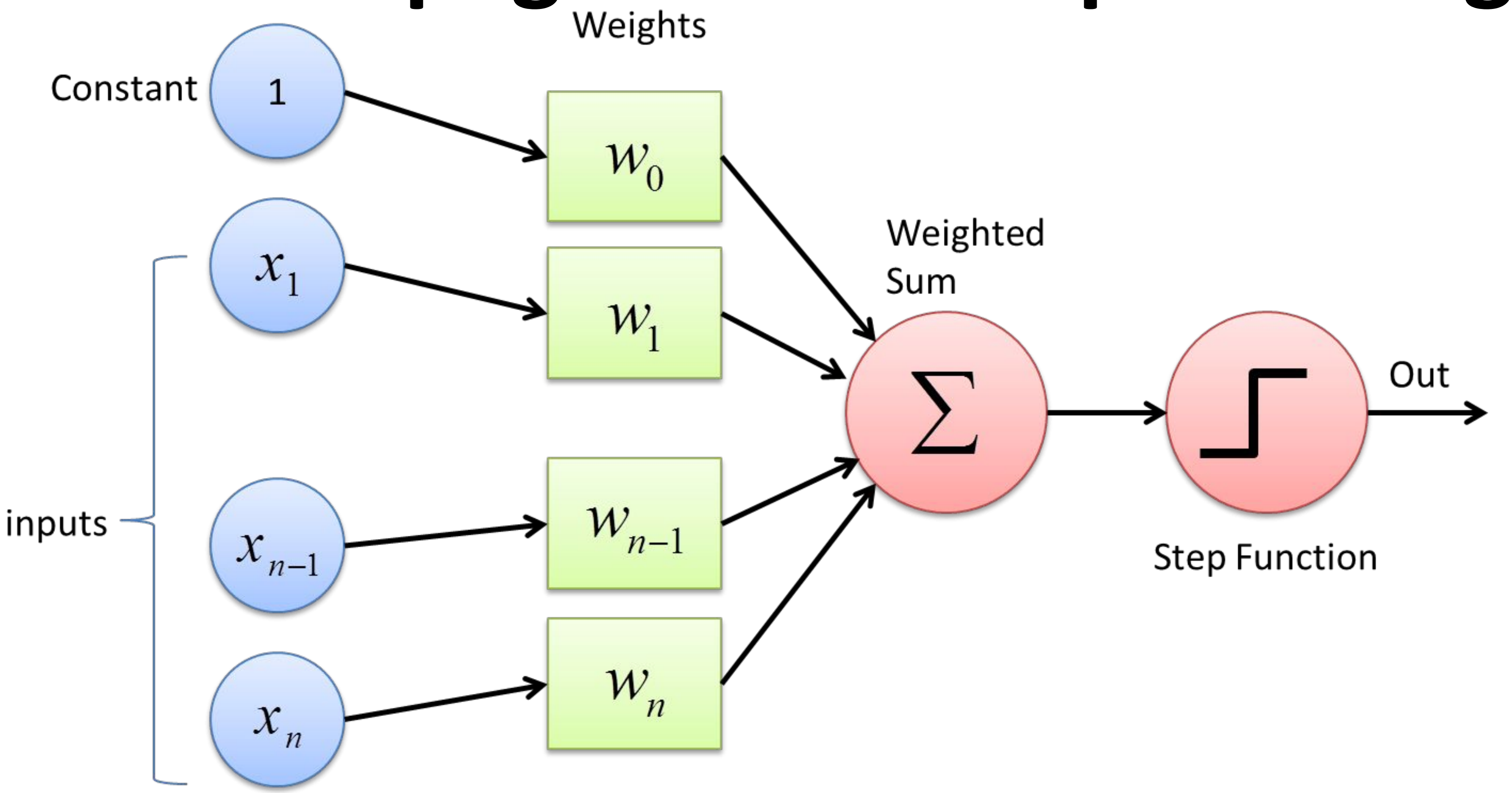
If sigmoid activation function: $\mathbb{P}(y = 1) = \phi\left(\sum_{i=1}^{m} w_i x_i\right)$

Thank You!

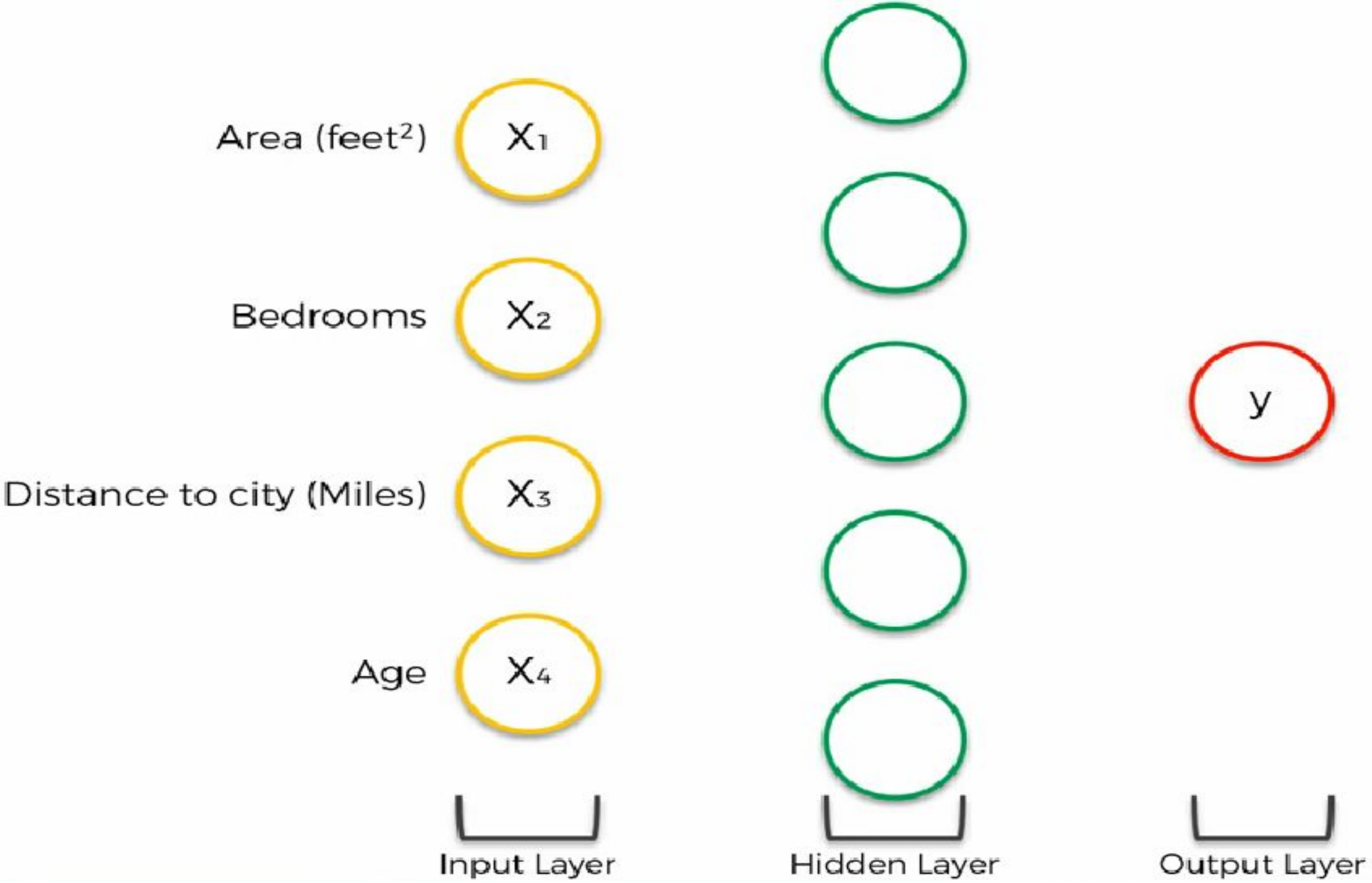
Deep Learning from Scratch

Theory + Practical

How Neural Network Work
and
Back Propagation in deep learning



How Neural Network Work with many neurons

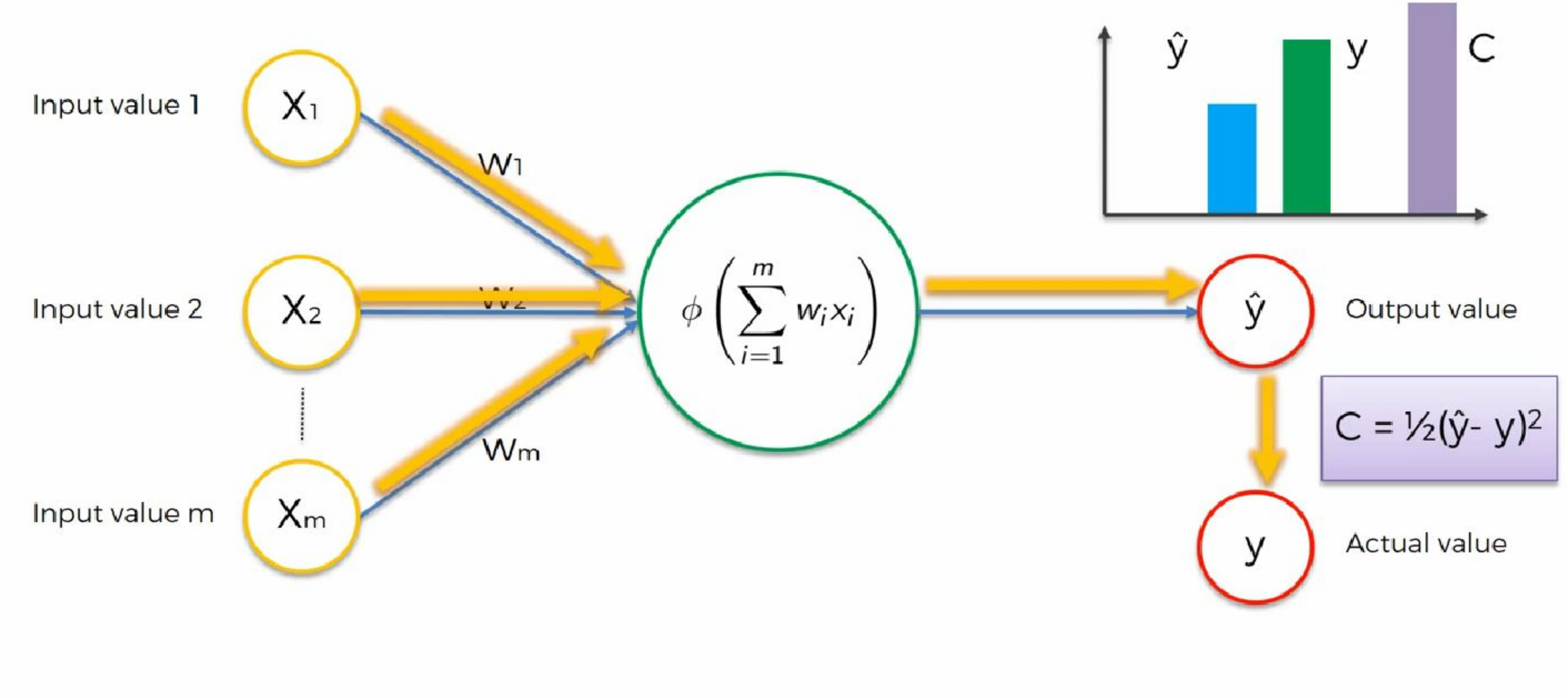


Back Propagation in deep learning

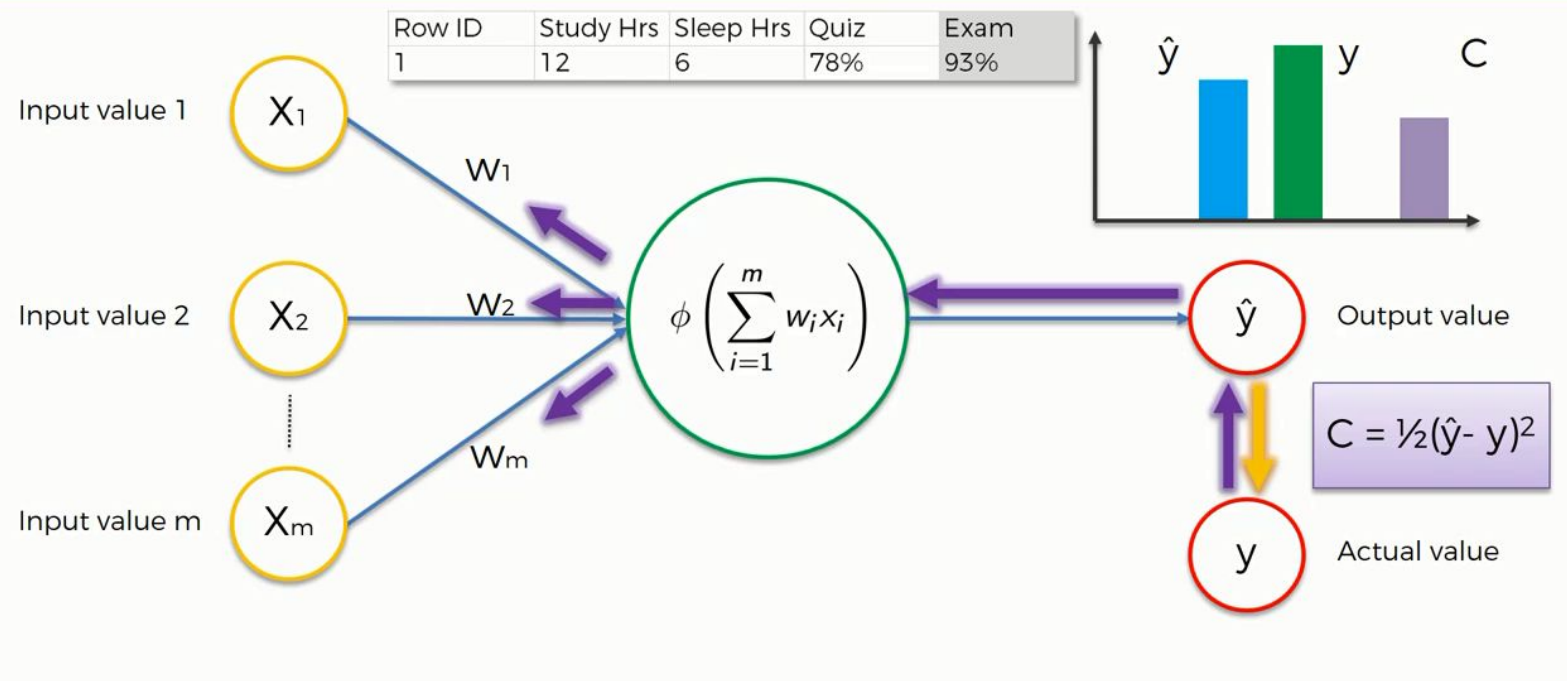
Back-propagation is the essence of neural net training. It is the method of fine-tuning the weights of a neural net based on the error rate obtained in the previous epoch (i.e., iteration). Proper tuning of the weights allows you to reduce error rates and to make the model reliable by increasing its generalization.

Backpropagation is a short form for "backward propagation of errors." It is a standard method of training artificial neural networks. This method helps to calculate the gradient of a loss function with respects to all the weights in the network.

Back Propagation in deep learning



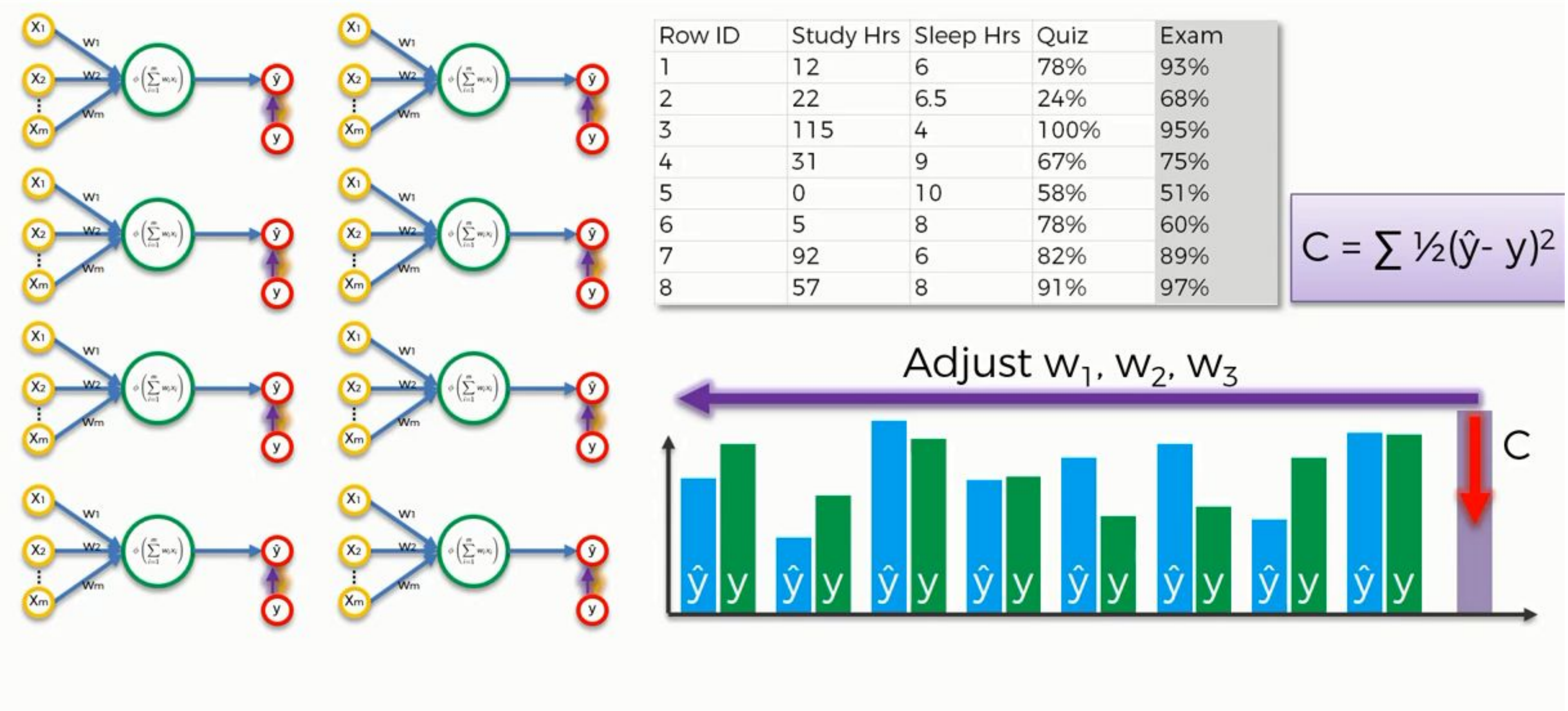
Back Propagation in deep learning



Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%

$$C = ½(\hat{y} - y)^2$$

Back Propagation in deep learning (epoch)



Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

$$C = \sum \tfrac{1}{2}(\hat{y} - y)^2$$

Adjust w_1, w_2, w_3

For further assistance Visit Stack Exchange

https://stats.stackexchange.com/questions/154879/a-list-of-cost-functions-used-in-neural-networks-alongside-applications