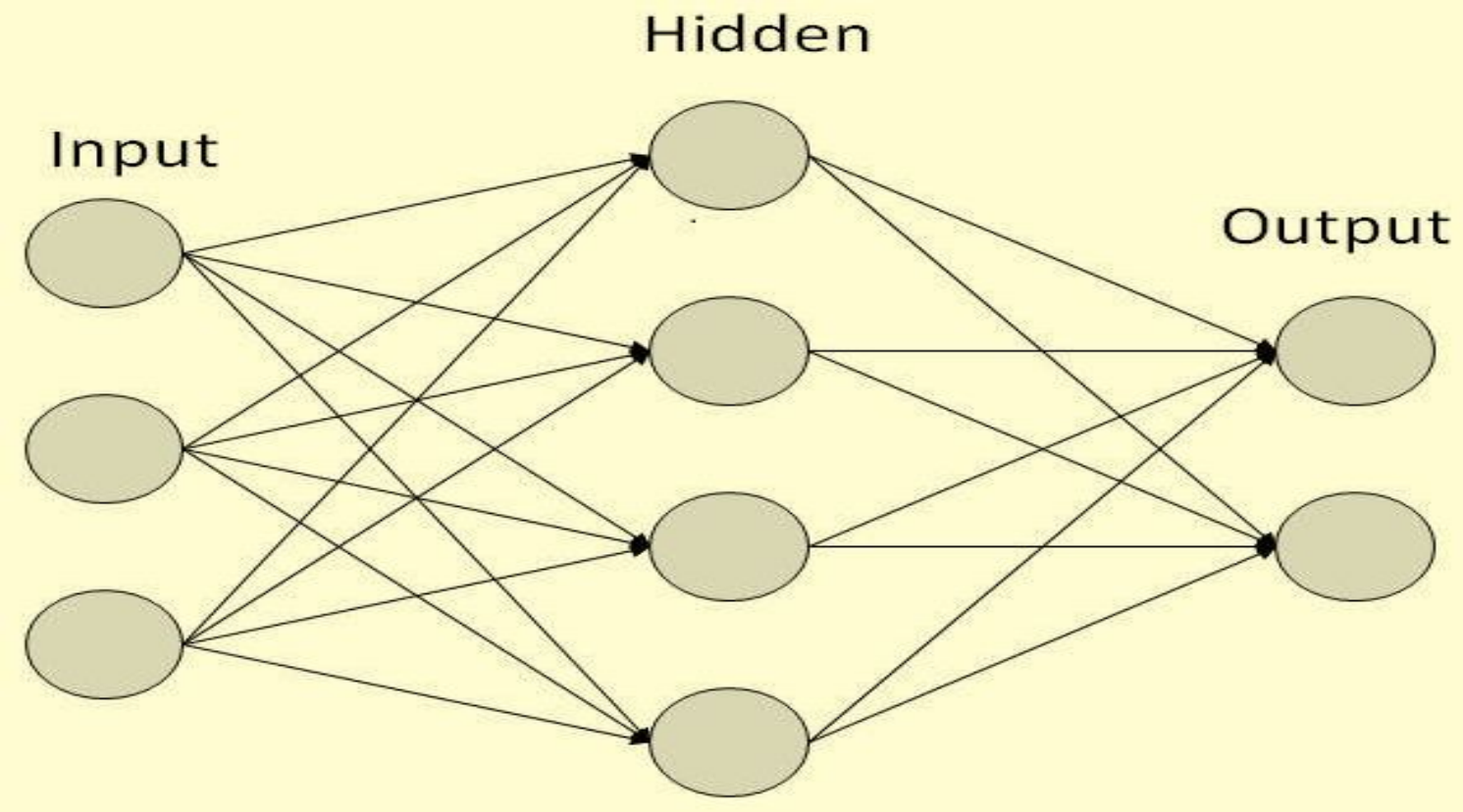
How do Neural Network works?

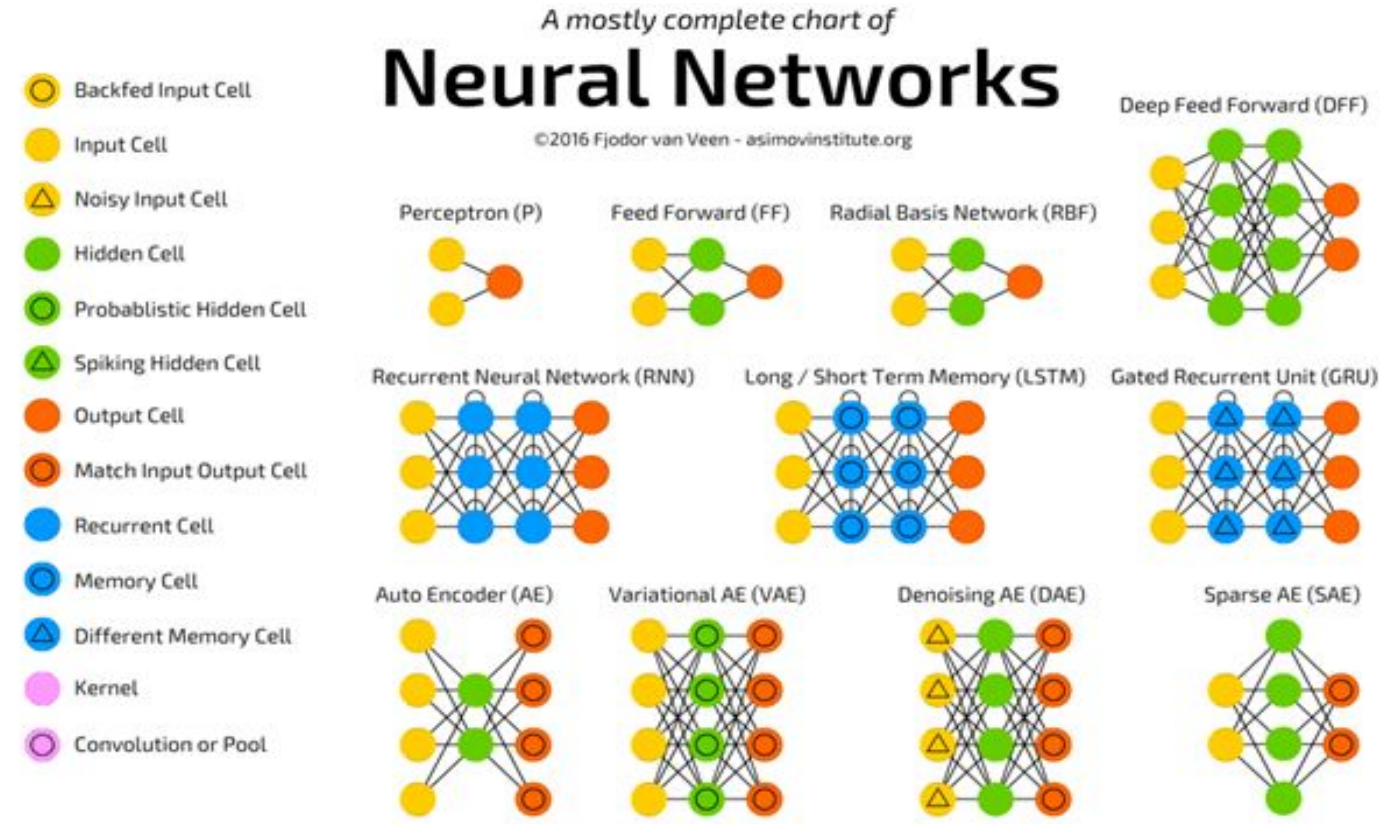


- The leftmost layer in this network is called the input layer, and the neurons within the layer are called *input neurons*.
- The rightmost or *output* layer contains the *output neurons*, or, as in this case, a single output neuron.
- The middle layer is called a *hidden layer*, since the neurons in this layer are neither inputs nor outputs.

Deep Learning Methods

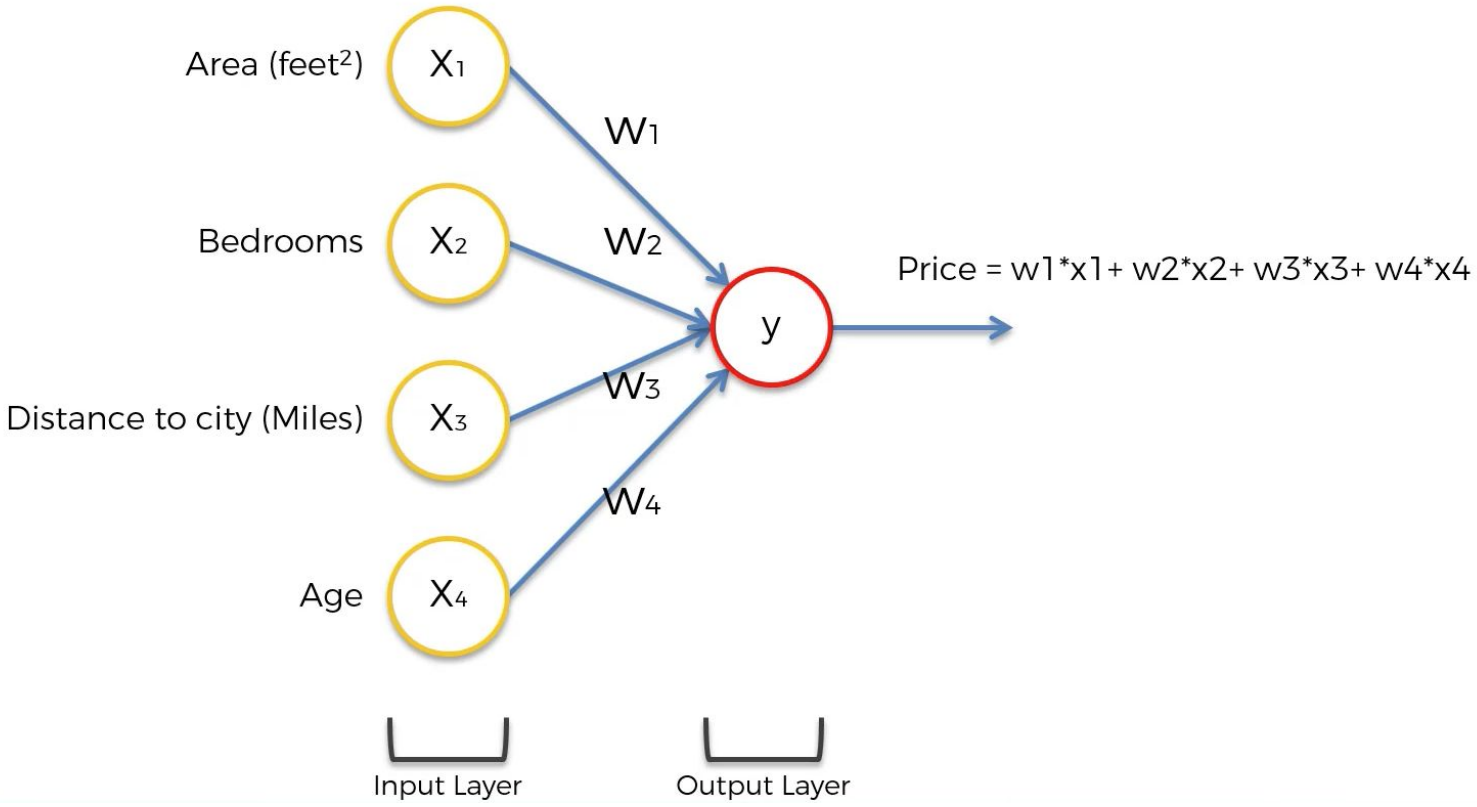
Deep learning then can be defined as neural networks with a large number of parameters and layers in one of four fundamental network architectures:

- Unsupervised Pre-trained Networks
- Convolutional Neural Networks
- Recurrent Neural Networks
- Recursive Neural Networks

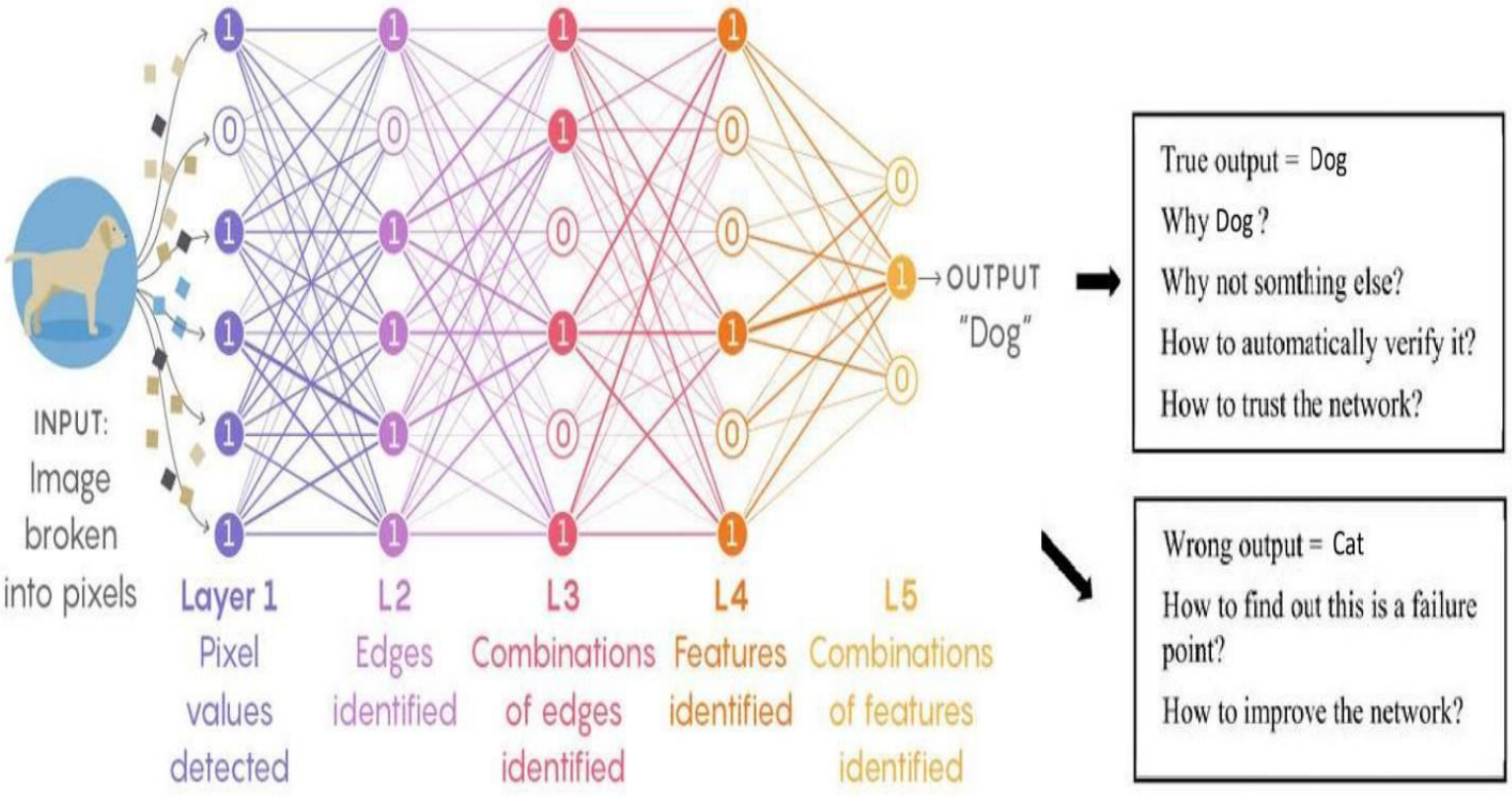


The Most Basic Form of a Neural Network

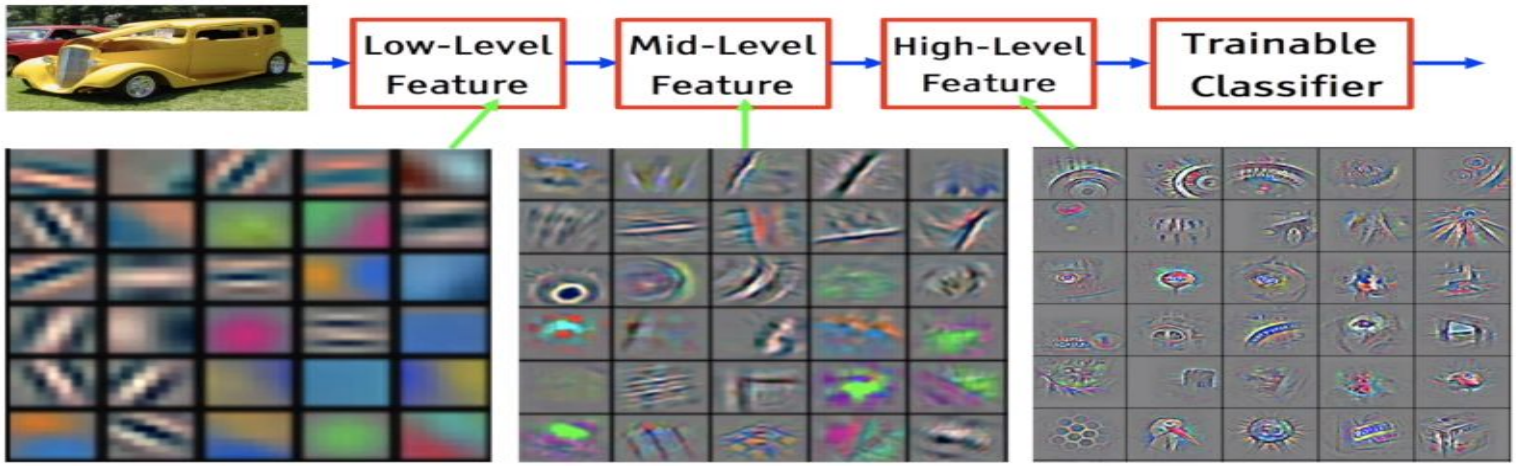
- In this a neural network only has two layers - the input layer and the output layer. The output layer is the component of the neural net that actually makes predictions.
- For example, if you wanted to make predictions using a simple weighted sum (also called linear regression) model..



Identifying a Dog

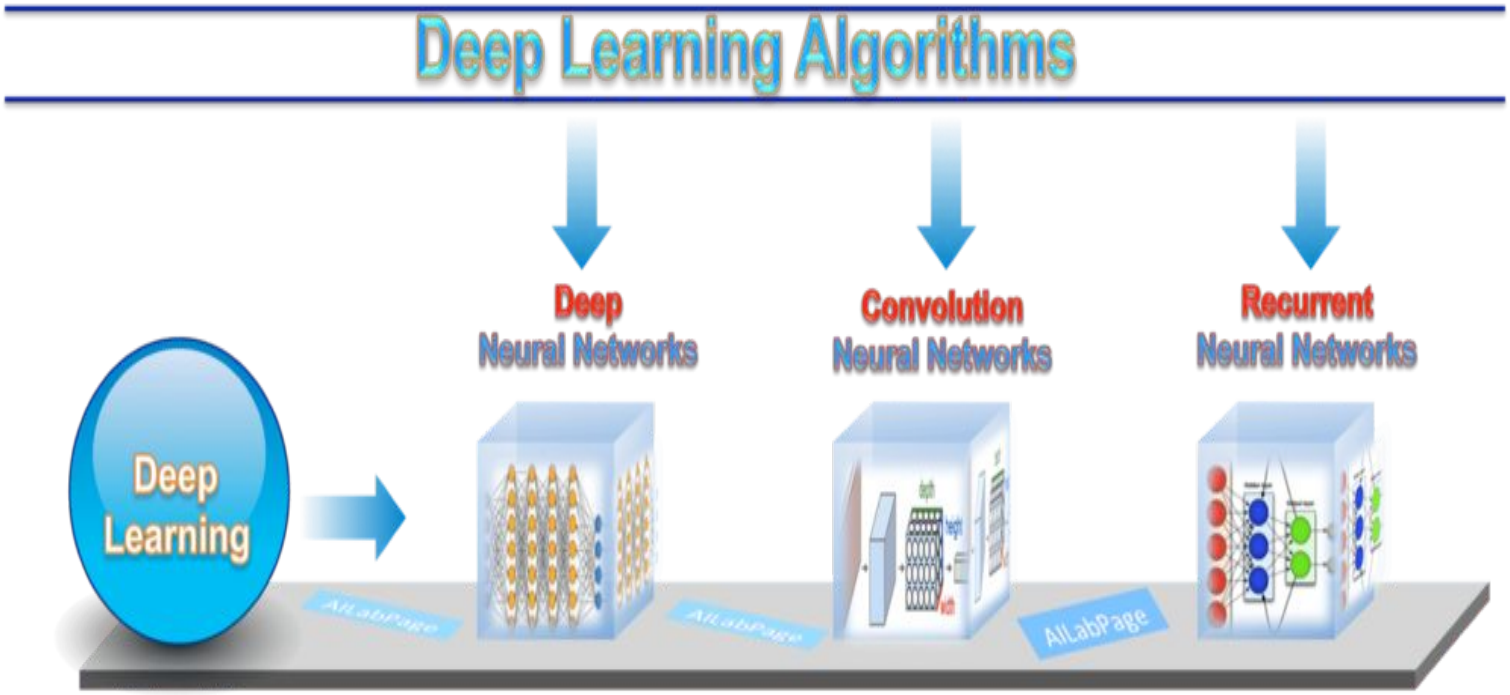


Capturing information



INTERNSHIPSTUDIO

DL types and usages



A method for applying simple mathematical function to Data (Voice, Text, Images or Videos).

It learns high level features by detecting complex interaction

For improved traditional algorithms

Used for

Fraud detection by identifying complex patterns in finance

Identify defects based on deeper anomaly detection for manufacturing

For processing images

Used for

Images, Healthcare, Satellite images (Maps), Retails for in store traffic by video processing, insurance and automotive

For sequenced data

Used for

Customer Services, Social media posts, Photo captioning, Predicting customer behavior based via time series and recommendation systems

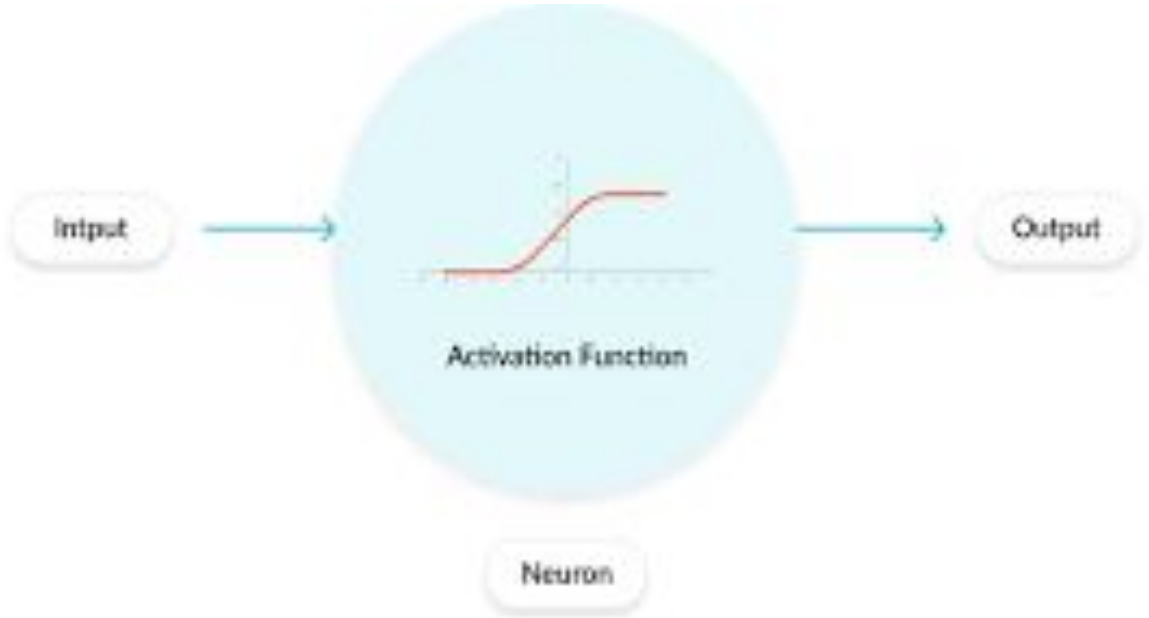
INTERNSHIPSTUDIO

- What is Deep Learning?
- What is the use of Deep Learning?
- What are Neural Networks?
- What is an Activation Function?
- What are the applications of Deep Learning?

Deep Learning

INTERNSHIPSTUDIO

Why Do We Need Activation Functions?



- In neural networks to make neuron intelligent on activation process is needed i.e. when to get activated & when not to.
- A perceptron is either 0 or 1 we need a smoother function that progressively changes from 0 to 1 with no discontinuity.

What is an Activation Function?

INTERNSHIPSTUDIO

An activation function is a non-linear function applied by a neuron to introduce non-linear properties in the network.

- A relationship is linear if a change in the first variable corresponds to a constant change in the second variable.
- A non-linear relationship means that a change in the first variable doesn't necessarily correspond with a constant change in the second. However, they may impact each other but it appears to be unpredictable.

Linear function

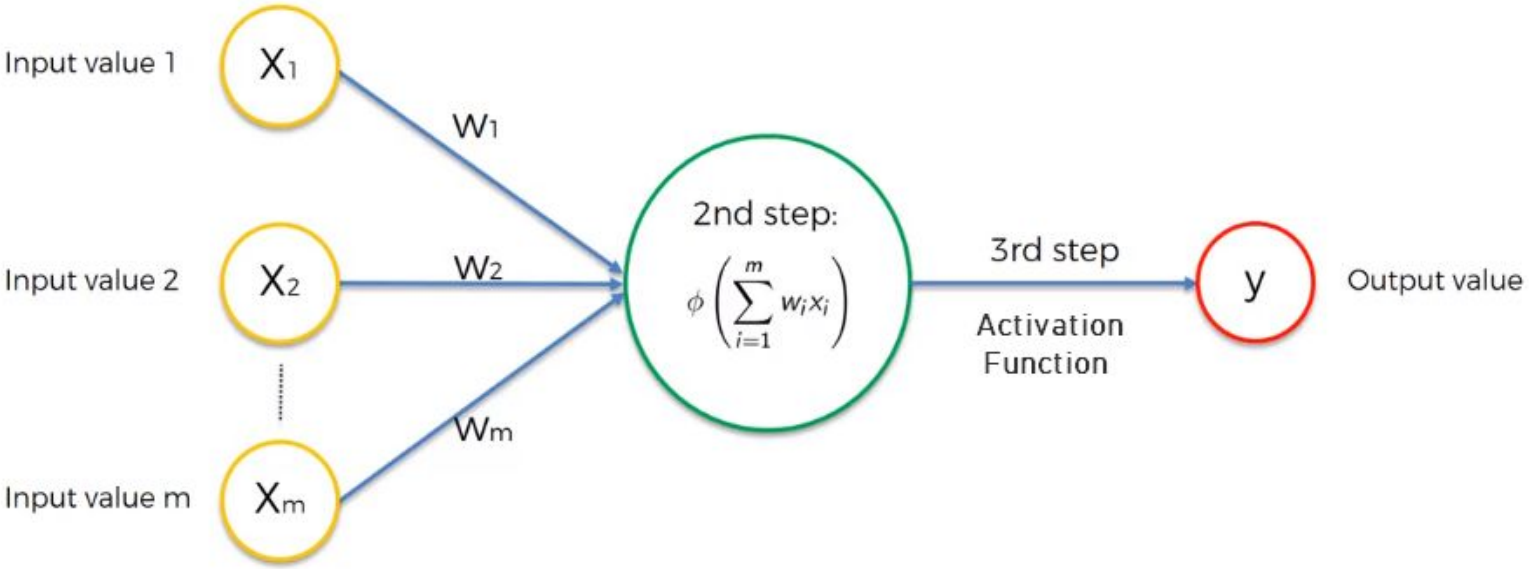
Non-linear function

Activation Function- Working

- What an artificial neuron do is to calculate a weighted sum of its input, adds a bias and then decides whether it should be "fired" or not.

$$Y = \sum(weight * input) + bias$$

- Considering the neuron of the figure above, the value of Y can be anything ranging from -inf to +inf. The neuron really doesn't know the bounds of the value. How do we decide whether the neuron should fire or not? We use activatioin functions for this purpose. To check the Y value produced by a neuron and decide whether outside connections should consider this neuron as fired or not.

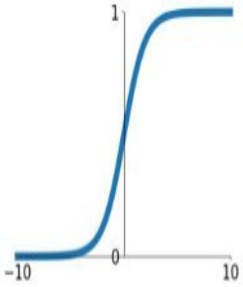


Types of Activation functions

INTERNSHIPSTUDIO

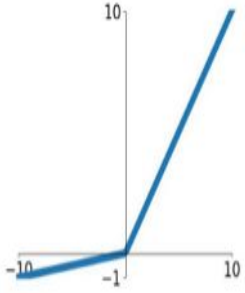
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



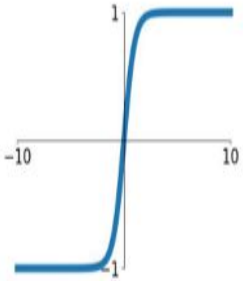
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$



Maxout

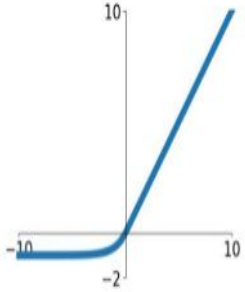
$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ReLU

$$\max(0, x)$$

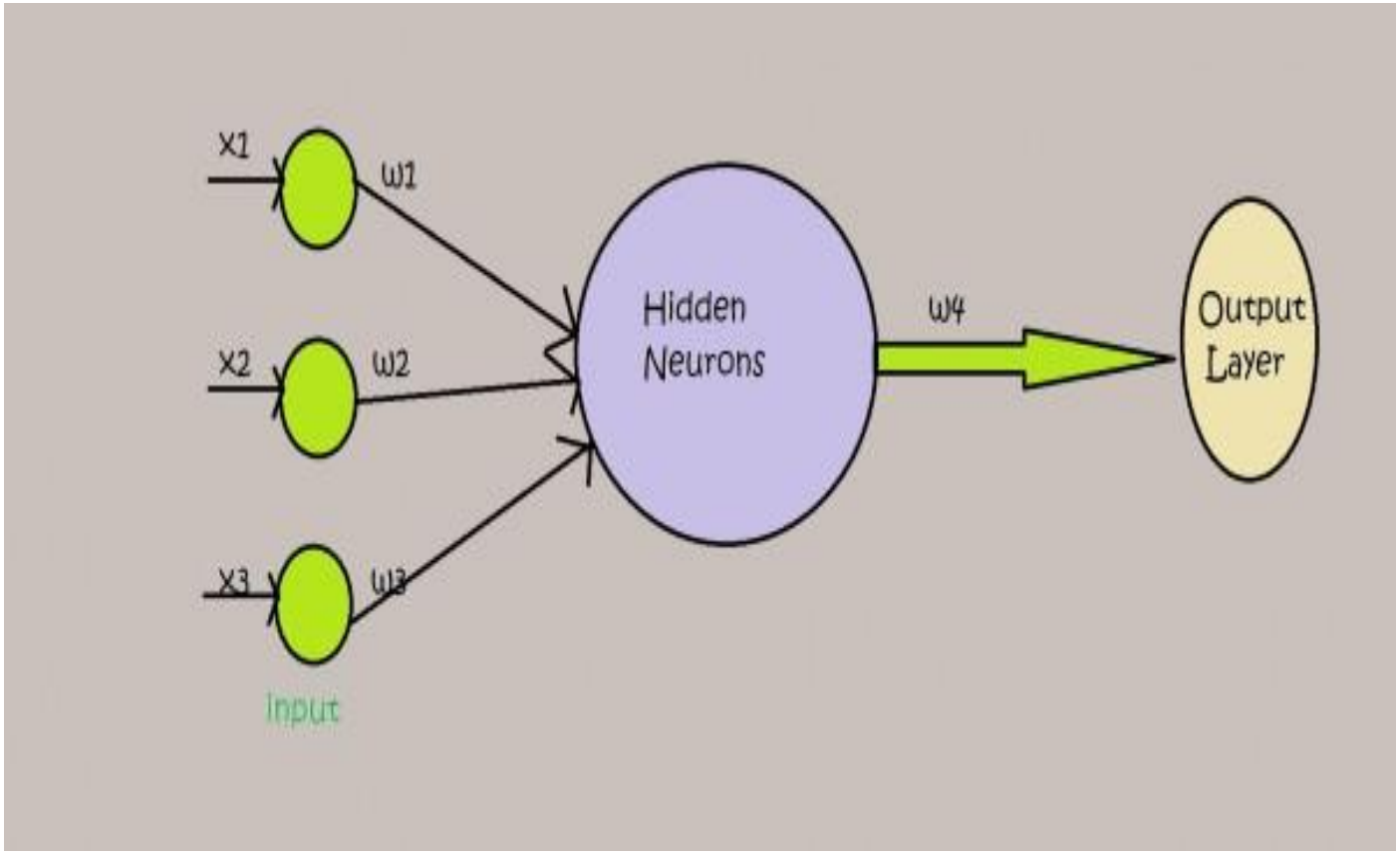
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



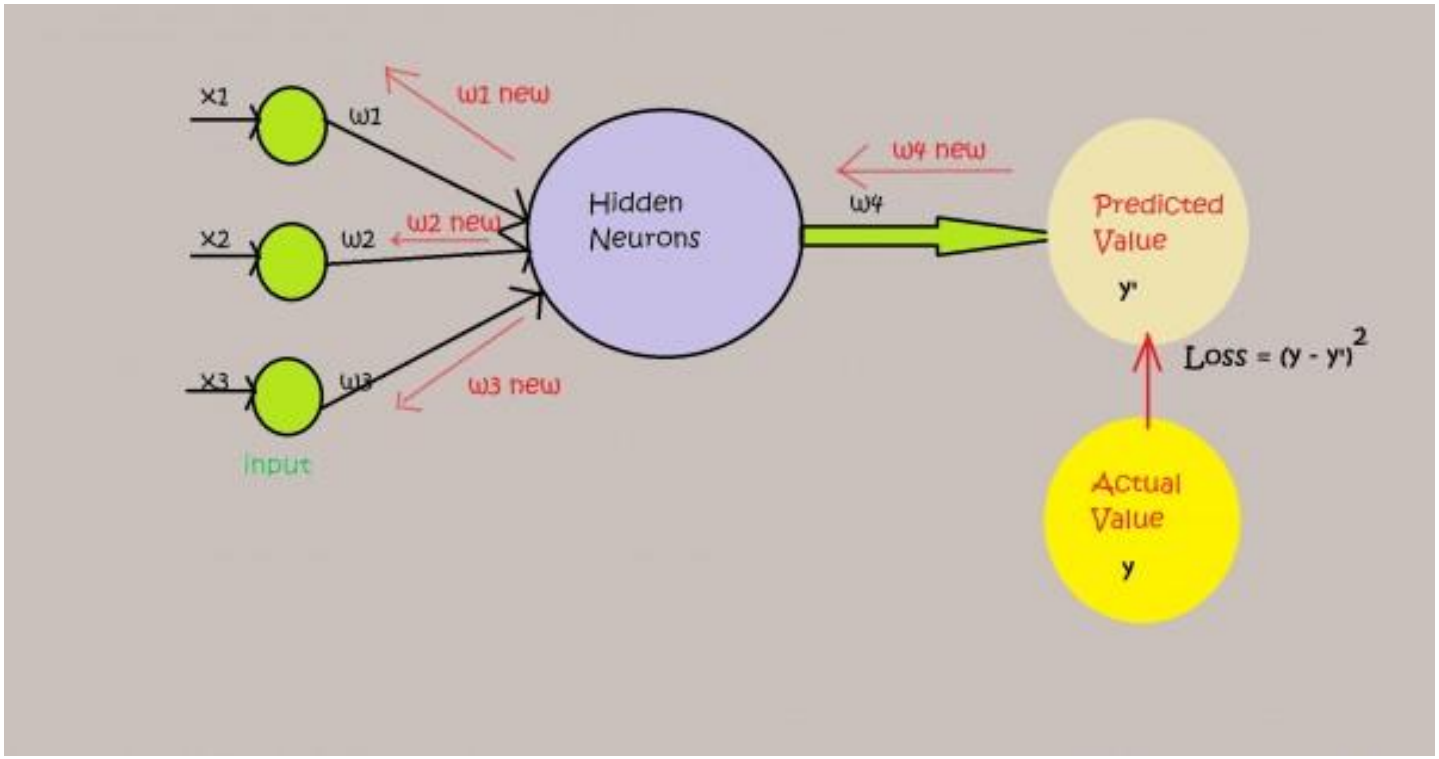
Forward Propagation

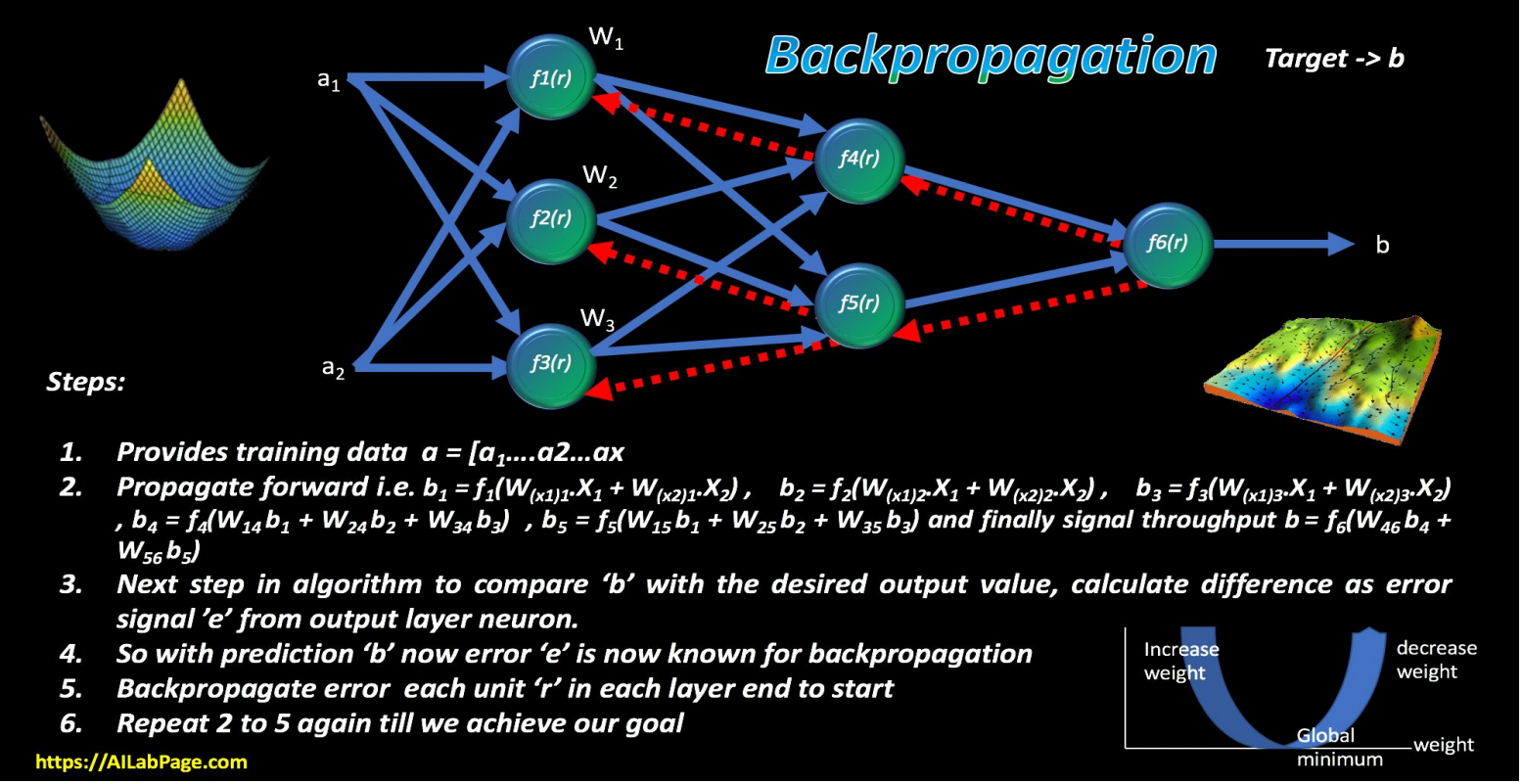
In feed forward network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network.



Backward Propagation

- Backpropagation is a widely used algorithm in training feedforward neural networks for supervised learning.
- Firstly it propagates in forward direction and gives the predicted output.
- Then, we check the difference between the predicted output and actual output.
- In order to minimize this loss we will back propagate and update our weights.





- ***Set inputs and desired outputs*** – Choose inputs and set the desired outputs
- ***Set random weights*** – This is needed for manipulating the output values.
- ***Calculating the error*** – Calculating error helps to check how far is the required output from the actual. How good/bad is the model output from the actual output.
- ***Minimising the error*** – Now at this step, we need to check the error rate for its minimization
- ***Updating the parameters*** – In case the error has a huge gap then, change/update the parameters i.e. weights and biases to reduce it. Repeating this check and update process until error gets minimised is the motive here.
- ***Model readiness for a prediction*** – After the last step, we get our error optimised and, we can now test our output with some testing inputs.

Forward Propagation Steps

Input -

- 1. The features of your data is passed through input layer at first.
- 2. Firstly some weights are assigned to each input features. In this diagram, x1, x2, x3 are input features.

Hidden Neurons

- 3. Then, the information is passed through all the neurons in the hidden layer.
- 4. Then, they the input features are multiplied with their respective weights and added together along with a bias to get a value

  y = x1*w1 + x2*w2 + x3*w3 + bias

- This function is multiplied with output weight w4 and passed to

Activation function

- 5. This value is passed to activation function so that only required activation functions can get activated as per the passed threshold values.

  z = y*w4 = Activation(z)

Output

- 6. Finally, after the pre processing is done in hidden neurons, we get the output depending on number of classes.

LIMITATIONS - Using FF propagation we cant update our weights. Since, initial weights are generally randomly initialized so they will most likely give huge losses.

- We need to minimize this loss such that our predicted values are closer to actual values. Therefore, another technique came into play later on known as back propagation in which we go back and update the weights for minimizing the cost functions.

Backward Propagation Steps

INPUT LAYER

1. Firstly some weights are assigned to each input features. In this diagram, x1, x2, x3 are input features.
2. Then, the information is passed through all the neurons in the hidden layer.

PRE PROCESSING IN HIDDEN LAYER

3. Then, they the input features are multiplied with their respective weights and added together along with a bias to get a value

    y = x1*w1 + x2*w2 + x3*w3 + bias

4. This function is multiplied with output weight w4 and passed to activation function in the hidden layer. It activates only required activation functions can get activated as per the passed threshold values.

    z = y*w4 = Activation(z)

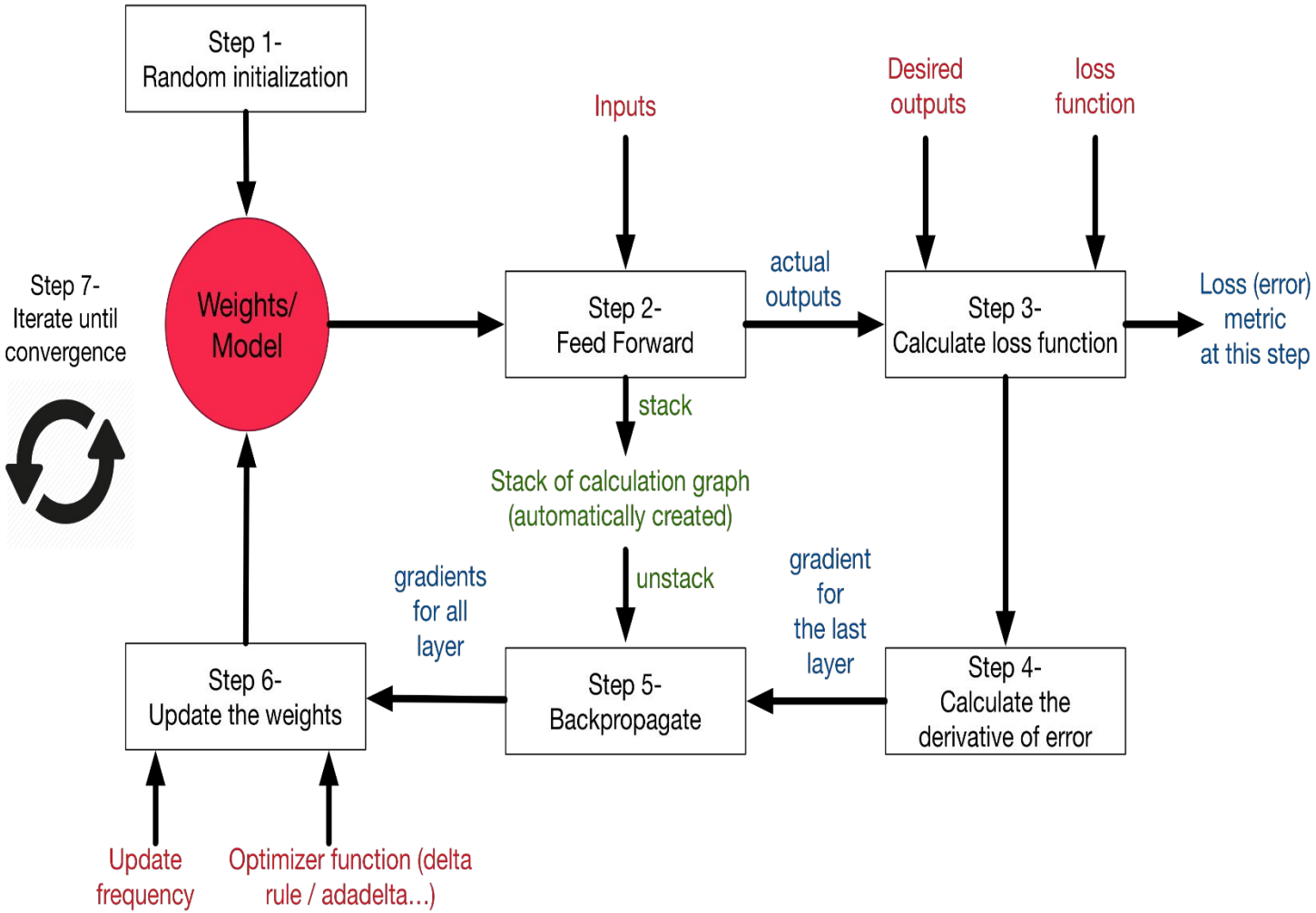
CALCULATING LOSS FUNCTION

5. Then loss value is calculated using square of difference of actual and predicted output.

    Loss = (y - y') square

6. We do square so that in case we get negative value it becomes positive.

Learning steps- Neural Networks



Learning steps- Neural Networks

INTERNSHIPSTUDIO

INTERNSHIPSTUDIO

- Why we need an activation function?
- What is Forward Propagation?
- What is Backward Propagation?
- What are different types of activation function?