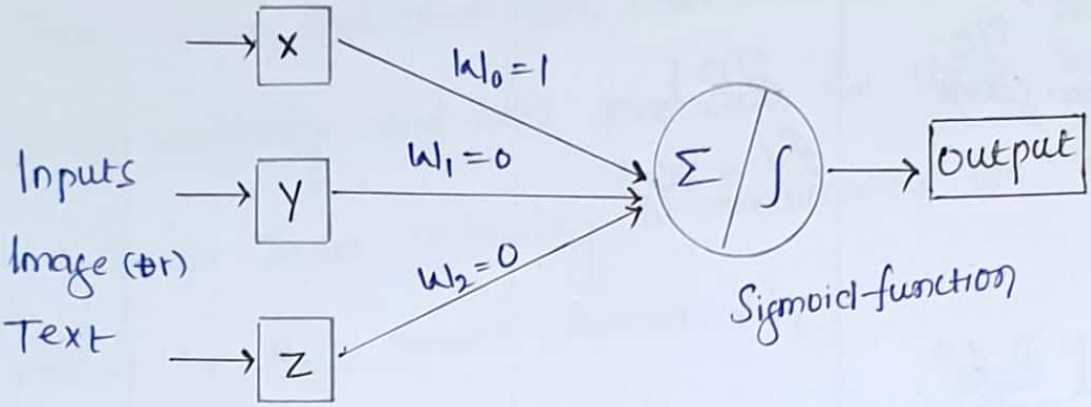
Perceptron :-

Percepton is a neutal network unit (an artificial neuron) That does certain computations to detect features or business intelligence in the input data.



Activation function :-

Activation functions are mathematical equations, that determine the output of a neural network. The function is attached to each neuron in the network and determines wether it should be activated or not.

Working process :- i) Take input and multiply by the neurons weight. then add the bias

ii) feed the result x to the activation function : $f(x)$

iii) Take the output and transmit to the next layer of neurons.

Neural Network form → $a(W \cdot I + b)$ → Bias
- W → weight
- I → input.
- a → Activation function

Generally activation function is operating at layer. it deals with perceptron, if more than one layer then we go with multi layer Perceptron.

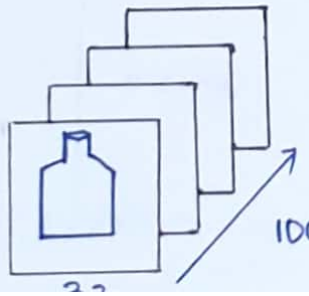
if we observe neural network bond Initially it has too many Inputs when that deals with other newton ie activating neuron with Sum weightage and bias previously we shown in figure. that particular input activating the functionality. the resultant output should be that neuron functionality.

In process of input to output there dealing with some other neurons also. in the form of layer they are structured this layers are generally called as hidden layers.

The number of hidden layers should be between size of input layers and size of output layers.

The number of hidden neurons should be $\frac{2}{3}$rd size of input layers. [Generally consider not exact].

Ex: Extraction of features from an Image is done by multi-layer perceptron.



⇒ Represent as - pixels - 32 x 32 x 3 → HL1 (channel)

Similarly - 16 x 16 x 64 - HL2

$[w_3, b_3]$ 8 x 8 x 128 - HL3

4 x 4 x 256 - HL4

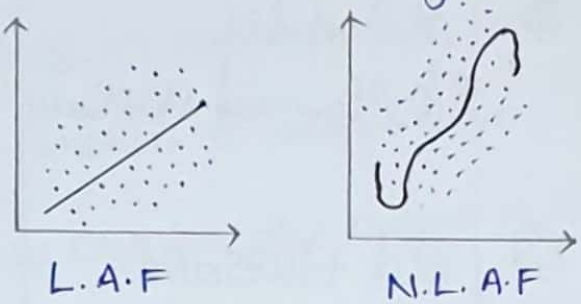
$[w_m, b_m]$ 1 x 1 x 1024 ⇒ 1024 HL m. (cat / dog)

Types of Activation functions :-

Generally Activation functions classified as two categories -

① Linear Activation function

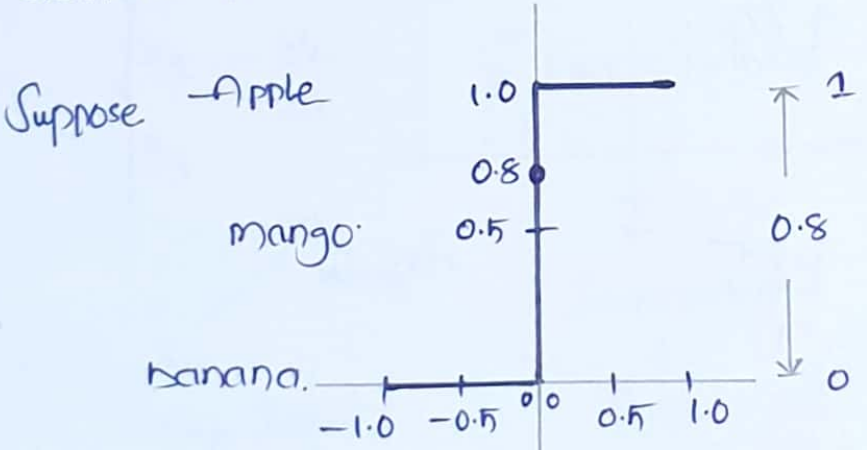
② Non-linear Activation "



* Linear Activation function Generally Indicates line eqn ie $y = mx + c$ where if feature of 'x' Increases 'y' value also increase.

Ex: for suppose if we observe budget of Indian Govt, if it is public holic then budget decrease Simeltiniously stock also decrease. if the case where budget Increase and not public holic then stacks also Increase ie there is positive relation in between them.

Binary Step function :- it is a threshold based activation - function, which means after a certain threshold neuron is activated. and below that threshold neuron is deactivated.



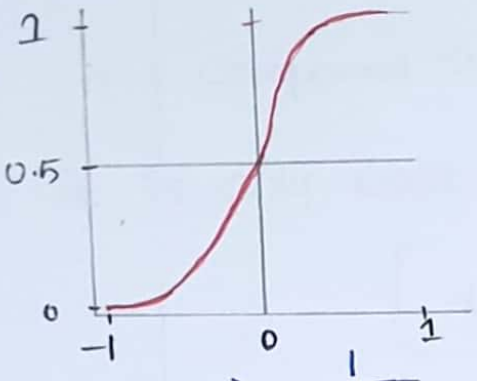
Range (0-1)
pointing 0.8 ie
category of 1
point belongs to
an apple

Sigmoid function :- $\sigma(x)$

Sigmoid function is also called as Squashing function, as its domain is the set of all real numbers and its range is (0,1), if the input to the function is either a very large negative number or very large positive number the output is always between 0 and 1. and it is a non-linear activation function.

formula – $f(x) = \frac{1}{(1+e^{-x})}$; $x = w1.1 + b$

Sigmoid Curve :-



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

$x = \sum w_i x_i + bias$

Ex $w1 = 2$, $I = 4$, bias = 1

$x = w1\,1 + b$

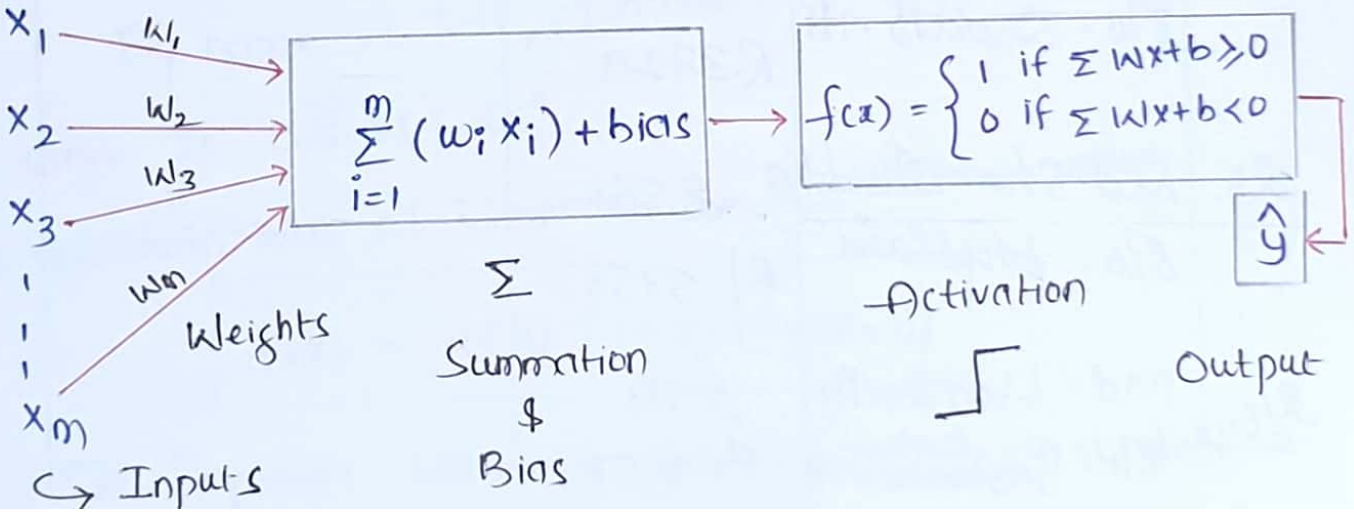
$x = 2 \times 4 + 1 = 9$

$f(x) = \frac{1}{1+e^{-9}}$

$f(9) = \frac{1}{1+e^{-9}} = 0.99$

Range of S.F is – (0 -1)

Procedure of a Single layer Network :-



Diffrent Types of Sigmoid functions :-

1) Hard Sigmoid
2) Sigmoid weighted linear units
3) Derivative Sigmoid linear Weighted units.

Hard Sigmoid :- it is also similar like Sigmoid but change in computation only. by using this we activate neuron.

$$f(x) = \text{Max}(0, \text{Min}(1, \frac{x+1}{2}))$$

Sigmoid weighted linear units :-

As compared to hard Sigmoid it is low computation cost. it is only used in reinforcement learning - research based.

$$f(x) = Z_k \cdot \alpha(Z_k)$$

Where Z_k – Input to hidden units $= (w1+b)$

α – Weights (0.8, 0.9, 0.1)

Derivative Sigmoid weighted linear units :-

By name it self, derivation of Sigmoid weighted linear unit is "D.S.W.L.U".

By the chain rule of derivation we got this equation.

$$f(z) = \alpha(z_k).(1 + z_k(1-\alpha(z_k)))$$

It Improves vanishing Gradient which is a deffeut

Tan H | Hyperbolic Tangent :-

"Tanh" function also called as hyperbolic tangent activation function, it is very similar to the Sigmoid function. range of this function is (-1 to 1).

Advantage of this function is - it works better in training for multi layer perceptron.

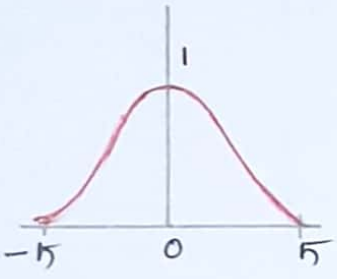
The activation gives strong decision of postive / Negative

Used in N.L.P / N.L.U. ⇒ for example Sentimental Analysis.

Disadvantage of this function is computation cost is very high.

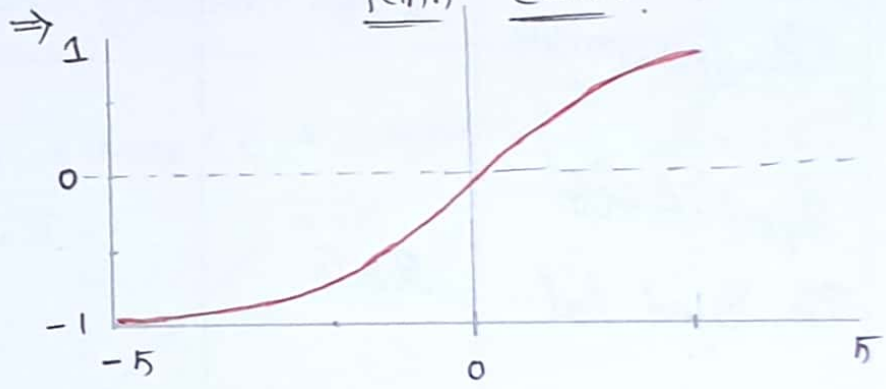
$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$\frac{d\,\text{Tanh}}{dx}$



⇒

"Tanh" Curve.



Ex: $f(2) = \frac{e^2 - e^{-2}}{e^2 + e^{-2}}$

$= \frac{7.38 - 0.13}{7.38 + 0.13}$

$= 7.22 / 7.51 = 0.96 \simeq 1$

Hard Tanh : it is One Type of Tanh Activation.

it takes Less Computation, hence works faster

Used in Speech Recognization, Sentiment analysis,

it learning text data., Accuracy Improvement.

$$f(x) = \begin{cases} -1 & \text{if } x < -1 \\ x & -1 \leq x \leq 1 \\ 1 & x > 1 \end{cases}$$

ReLU (Rectified linear Unit) :-

Performs near to linear function, hence it helps in linear model

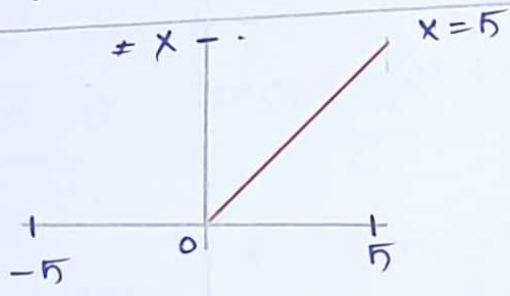
Advantage :- i) very easy Computation

ii) it introduces Sparsity in hidden Units.

Disadvantage :- i) The model easily overfits.

ii) Gradient die.

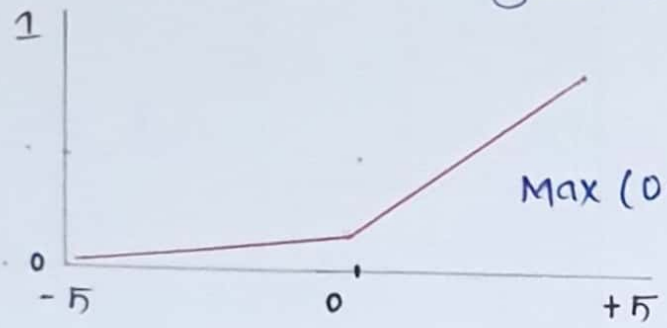
$$f(x) = \text{Max}(0, x) \text{ where. } \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$



Leaky Relu :- it is an activation function based on a. ReLU, but it has a small slope for negative values instead of a flat slope., The slope Coefficients is determind before training.

Leaky Relu — $f(x) = \alpha x + x$

Where α – is constant which makes negative gradient in range of 0.01 (learning rate)



$$\begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases}$$

Ex $x = -1 \Rightarrow f(-1) = 0.01 * (-1) + (-1) \Rightarrow -1.01$,

Parametric Relu (PReLU) :-

Parametric Relu is a type of leaky relu that, Instead of having a predetermind slope like 0.01, makes it a parameter for the neural network to figure out itself.

It has advantage of adaptive learning during back propagation

it is mostly used in CNN / Image recognization.

$$f(x) = \max(0, a) + a \cdot \min(0, x)$$

↳ Learning Rate

$$f(x) \begin{cases} x, & \text{if } x > 0 \\ \max(0, a) + a \cdot \min(0, x), & \text{if } x \leq 0 \end{cases}$$

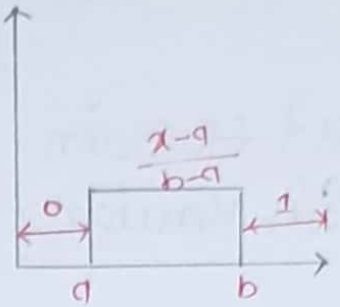
$$\Rightarrow a + a \cdot x$$

$$f(x) = a + a \cdot x$$

Randomized Leaky ReLU :-

- Dynamic varient of leaky relu where random number is sampled from a uniform distribution of $U(a,b)$

$$f(x) = \begin{cases} x & x \geq 0 \\ c.x & x < 0 \end{cases} +$$



$$c = \frac{a+b}{2}$$

Soft plus function :-

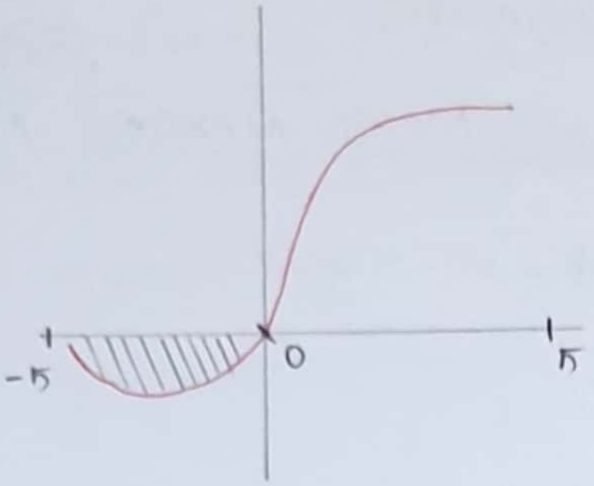
it is a smooth approximation to the relu function and can be used to contain the output of a machine to always be positive, for numerical stability the implimentation reverts to the linear function.

Note :- it is used for Smoothing & Non-zero gradient
it help in reduce epoch.

$$f(x) = \log(1 + \exp^x)$$

Exponential Linear unit [ELU] :-

It is an activation function for nueral networks. it contrast to ReLU's, ELU have negative values which allows them to push mean unit activations closer to zero like batch normalization but with lower complexity of computation.



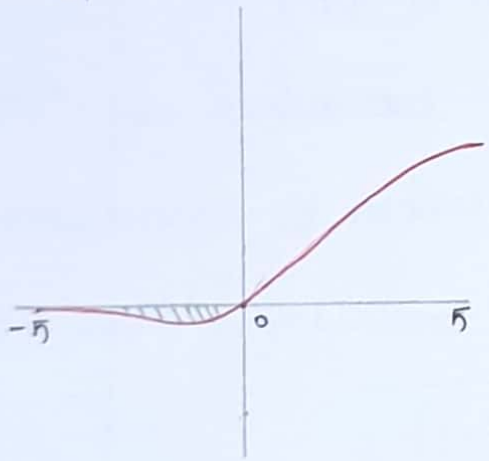
$$\Rightarrow \quad f(x) = \begin{cases} x & x > 0 \\ \alpha(\exp(x) - 1 & x \leq 0 \end{cases}$$

FLU is used to improve the training process in DL models.

It helps to remove vanishing gradient problem.

It reduce the bias shift.

Swish ReLU :- Swish is smooth, non-monotonic function that consistantly maches or outperforms ReLU on deep networks applied to a variety of challenging domains such as Image classification and machine translation.



$$f(x) = x * \sigma(x)$$

where $\sigma \rightarrow$ Sigmoid.

It used in LSTM (NLP | NLU)

LSTM Layer should be grater than 40.

Max out -

$$f(x) = \max(\hat{y}_1, \hat{y}_2, \hat{y}_3 \cdots \hat{y}_n)$$

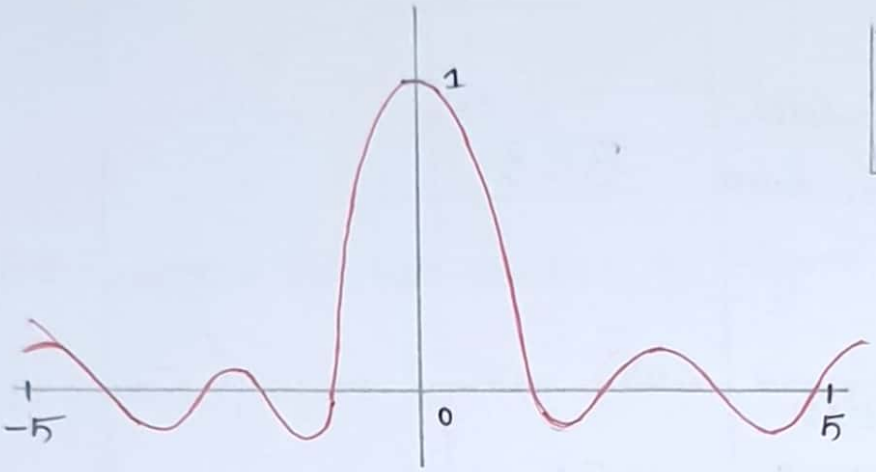
$$\hat{y}_1 = W_1^T x + b_1$$

$$\hat{y}_2 = W_2^T x + b_2$$

$$\vdots$$

$$\hat{y}_n = W_n^T x + b_n$$

Sinc function :- A Sinc function is an even function with unity area, a Sinc pluse passes through zero at all positive and negative intigers. (ie - ±1, ±2, ---) but at time, it reaches its maximum 1.



$$y = \frac{Sin(x)}{x} \quad \text{where } x \neq 0$$

This function is used by NASA

it helps in neuron diring.

Mish Activation function :-

Mish is mathematically defined as $f(x) = x \cdot \tanh(\text{softplus}(x))$. We evalute and find that mish tends to match or Improve the Performance of neural network architectures as Compared to that of swish, ReLU and leaky ReLU across different tasks in C.V.

$$f(x) = x \cdot \tanh(\text{Softpluce}(x))$$

Soft Max Activation function :-

It is used as activation function in the output Layer of neural network models that predict a multinominal Probability distribution.

The Softmax activation function is used in neural networks when we want to build a multiclass classifier, which solves the problem of assigning an instance to a class when the no of possible classes larger than two.

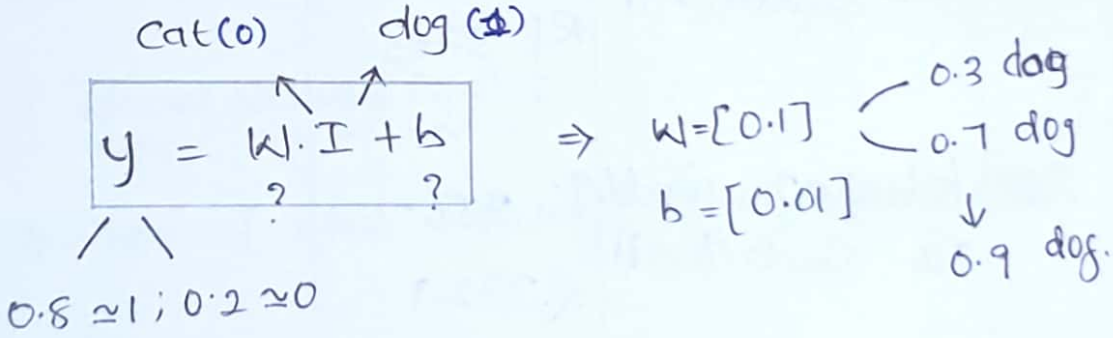
$$f(x) = \frac{e^x}{\sum e^{x_k}}$$

Where $x = WI + b$

For Suppose if we predict the figure of cat or dog by using Soft max activation function. where we give the weightage to threshold value, where the weight (W) value near to zero represent zero and if near to '1' represent '1' for Suppose if we train '1' as dog and '0' as cat if the Output as

$0.8 \simeq 1$ (near to 1) represent the Image is dog.

if output $0.2 \simeq$ "0" represent the Image is cat.



Weight (W) is always represent higher value ie 1 in Our Case that represent Image of dog. but the Output (y) defines the Image of input by threshold point.

Choosing Right Activation Function :-

① Sigmoid functions and their combinations generally work better in the case of classifier.

② Sigmoid & Tanh functions are sometimes avoided due to the vanishing problem. (vanishing Gradient)

③ ReLU function is a general activation function and is used in most cases these days.

④ if we encounter a case of dead neurons in our networks, the Leaky ReLU function is the best choice.

⑤ Always keep in mind that Relu function should only be used in the hidden layers.

⑥ As a rule of thumb you can begain with using ReLU function and then and move over to other activation functions in case ReLU doesn't provide with optimum result.

Training of a Neural Network (N.N) :-

i) Given X_i & Y_i think of what N.N. design and hyperparameter design might work.

ii) form a Neural Network "f(x)"

iii) Compute weights as estimating the 'y' for all samples.

iv) Compute the loss.

v) Tweak w to reduce the loss. & Repeat it.

Error and loss function :-

i) In General error / loss for a neural network is diffrence between actual and predicted.

ii) The Goal is to minimize the loss.

iii) By using loss-function we caliculate the error.

iv) loss-functions are different for classification and regression.

Backprop Algorithm :- this algorithm Searches for weight values that minimize the total error of the network over the Set of training

Back prop Consisting of a two passes _ ① forward pass
② Backward pass.

Forward pass :- In this Step the network is activated and error of (each neuron) the output layer is Computed.

Backward pass :- In this Step the network error is used for updating the weights, Starting at the output layer, the error is prapogated backwords through the network, layer by layer. this is done by recursively Computing the local gradient of each neuron.

* Backprapogation adjust the weights of N.N. in Order to minimize the network total mean Squared error.