Activation Functions:

These are used to normalise the data.

- The value of these functions in neural network determines whether a neuron should be activated or not in forward propagation to determine the output
- Activation function also plays role in backward propagation to alter the weights and finding the gradients.

The following are the list of activation functions:

1) Sigmoid Function

2) Tanh Function

3) Relu Function

4) Leaky Relu Function

5) ELU function

6) Softmax function

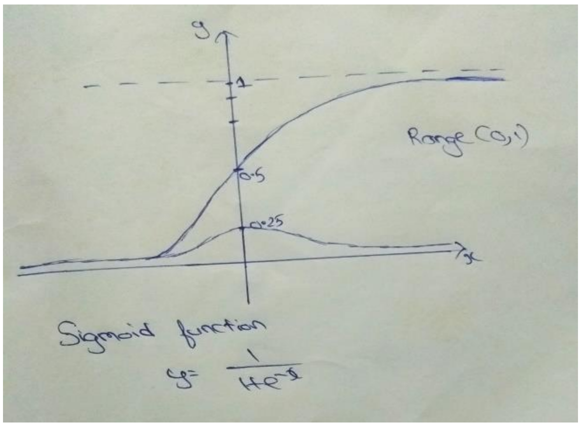
7) PRelu function

8) Swish function

9) Maxout function

10) Softplus function

1) **Sigmoid Function**

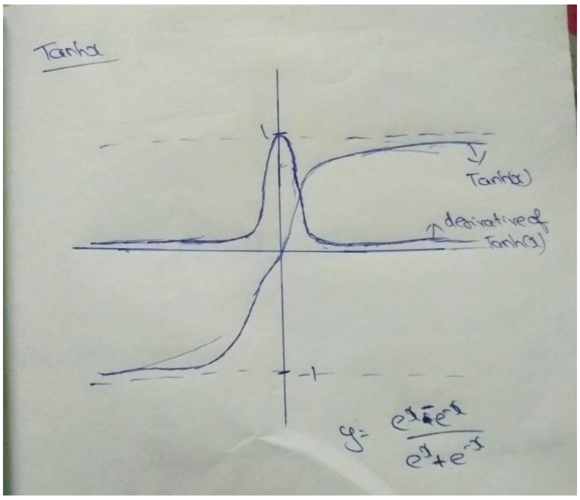


$$Y=(1+e^{-x})^{-1}$$

- Range of sigmoid function is (0,1) for all values of x.

- Gradients becomes smaller as we keep on moving backward Implies that neurons in earlier layers learn slowly as compared to neurons in the last layer (Vanishing gradient problem).
- This function is not zero-centric

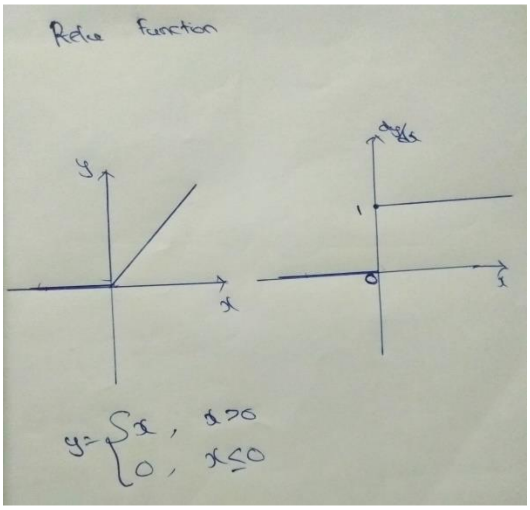
2) **Tanh Function**



$$Y= (e^{-x}-e^{x})/(e^{-x}+e^{x})$$

- Range of sigmoid function is (-1,1) for all values of x.
- Vanishing gradient problem exists in this function also.

3) **Relu Function**



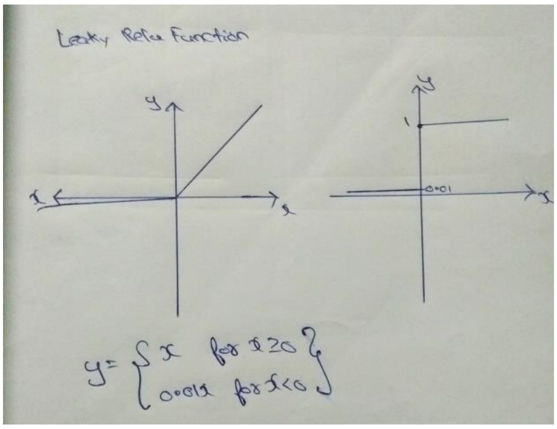
Y= max(x,0)

- Calculation is faster
- For values greater than zero , vanishing gradient problem do not occur.

Disadvantages:

- If weights become negative during forward propagation , the value is assigned to zero by relu activation function making the corresponding neuron inactive , termed as DYING neuron.
- To solve this problem another variation of RELU called as LEAKY RELU has been generated.

4) **Leaky Relu Function**



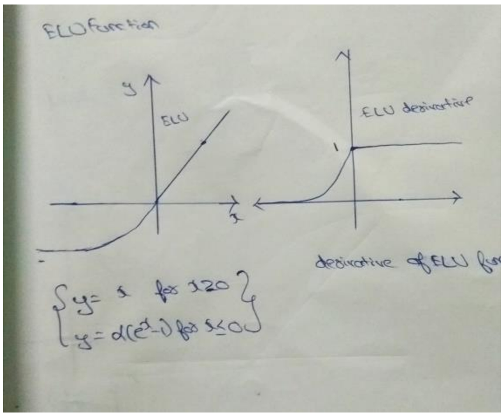
$$Y = x \text{ for } x>0$$

$$Y = 0.01x \quad \text{for } x<=0$$

Advantages:

- Calculation is simple.
- In RELU function for values less then zero , y is always constant, this has been solved here in leaky RELU .

5) **ELU function**



$$Y=x \text{ for } x>0$$

$$Y= \alpha(e^{x}-1) \text{ for } x<=0$$

Advantages:

- The issue of dying neuron in RELU for values less than zero has been solved here.

Disadvantages:

- Computation is expensive and time taking when compared to relu and its variants

6) **Softmax function**:

$$Y=(e^x)/(\sum e^x) \text{ for } x_i \text{ from (1 to n)}$$

- Gives probability of occurrence
- Mainly used in multi-class classification.
- Range of outpit of softmax activation function is (0,1)

7) **PRelu function**(Parametric Relu function):

Y= x for x>0

Y=αx for x<=0

Where α is hyper-meter and if α =0.01 it is Leaky relu and α is 0 then it is relu function.

8) **Swish Function:**

This is a self gated function.

Y= x*(sigmoid(x))

Advantages:

- This has practically been found working better than RELU
- Vanishing Gradient problem also does not occur in this activation function.

9)**Maxout Function**:

This function is generalised function for both RELU and leaky RELU functions.

This function does not have vanishing gradient problem but possess mor variables(hypermeter) for training making it longer to train.

$$Y=\max(w_1x+b_1,w_2x+b_2)$$

10) **Softplus Function:**

$$Y = \ln(1+e^{x})$$

$$dy/dx = sigmoid(x)$$

the derivative of softplus function is sigmoid function .So there is no vanishing gradient problem in softplus function.

General order for using Activation functions in our neural networks are:

ELU > Leaky ReLU > ReLU > tanh > sigmoid. Moreover, if we consider run-time then we may use Leaky ReLU as its computation is faster than ELU.

Loss Functions:

- Loss functions are used to regularize the model to avoid over fitting condition
- These Loss functions are inbuilt in keras module .
- Loss functions are used to make our neural networks perform better by optimising weights and bias .
- Loss function value should be minimum for our neural network to perform better.
- Loss functions are calculated based on predicted output(y^) on comparison with expected output(y).

Types of loss functions mainly used are as follows:

- L1&l2 loss function
- Huber loss function
- Hinge loss function
- Cross entropy loss function
- Sigmoid cross entropy loss function
- Softmax cross entropy loss function

1)**L1&l2 loss function:**

- L1 function also called as Least absolute deviation
- L2 function also called as Least square error
- L2 loss function is not advisable when the dataset contains outliers because error value becomes very large because of squaring the difference between expected and actual output.

$L1=\sum|y-y\hat{}|$

$L2=\sum|y-y\hat{}|^2$

2) **Huber Loss Function**:

- This loss function combines both L1 and L2 models and overcomes their limitations by creating a regularisation variable(δ)

Loss=0.5(y-y^)^2 for |y-y^| < δ
Loss= δ(y-y^)-0.5(δ)^2 for |y-y^| > δ

3)**Hinge Loss Function:**

- This function is more used for classification problems.

4) **Cross entropy Loss function**:

- This is used for binary classification problems

$$J= -\sum[t(\log(y)) + (1-t)(\log(1-y))]$$

5) **Sigmoid cross entropy loss function**:

- This is same as cross entopy loss function except that log value is replaced by sigmoid values

This is also used mainly during classification.

$$J= -\sum[t(\mathrm{sigmoid}(y)) + (1-t)(\mathrm{sigmoid}(1-y))]$$

Where ,

$$\mathrm{sigmoid}(y)=1/(1+e^{-x})$$

6) **Softmax Cross entropy loss function**:

- This loss function is used during multi class classification

$$J= -\log(e^{x}/\sum e^{x})$$