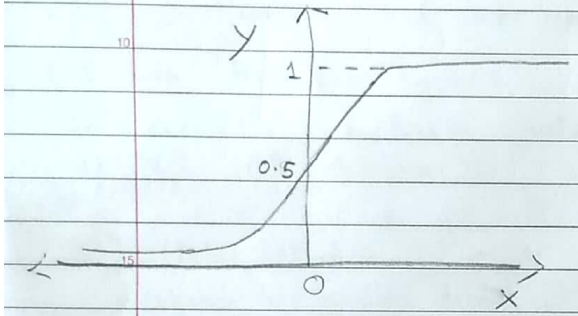
ACTIVATION FUNCTIONS

(1) SIGMOID

$$\text{Function} = \frac{1}{1 + e^{-x}}$$

Input = $Wx + b$

Output



It gives output between zero to one.

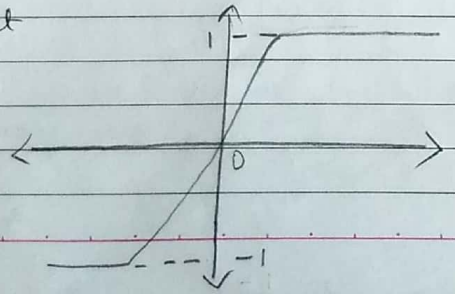
Drawbacks

a) It's an exponential function and hence high computation time
b) As value of X increases after a point y becomes constant known as vanishing gradient problem
c) It is not zero centric

(2) TANH FUNCTION

$$\text{Mathematical Expression} = \frac{e^{x} - e^{-x}}{e^{x} + e^{x}}$$

Output



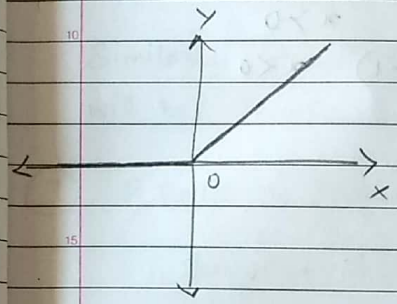
Advantages

a) It's zero centric function as it has values between -1 to +1
b) Normalized dataset

In binary class classification problems, we can use tanh function in hidden layer and sigmoid in ~~o/p~~ Output layer.

(3) ReLU (Rectified Linear Unit)

Function = $\max(0, x)$



Advantages

a) No vanishing gradient problem in positive direction
b) Computation is faster than sigmoid and tanh

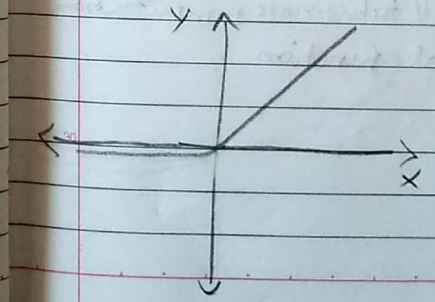
Disadvantage

a) In case of negative input there are no changes, O/P fixed
b) It is not a zero centric function

(4) Leaky ReLU

Function = +ve direction $\max(0, x)$

-ve direction = Input * 0.01



Advantages

a) Less computation time

b) In negative axis its goint to adjust gradient

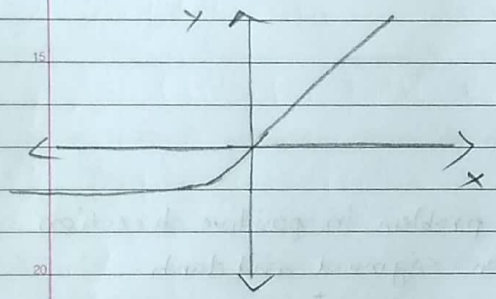
Disadvantage

a) Not so much gradient optimization in negative direction

(5) ELU (Exponential Linear Unit)

$$\text{Function } (f(x)) = \begin{cases} x & x > 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

Output



Advantages

a) No gradient clipping

b) Mean of output is close to zero and hence normalized and zerocentric

Disadvantage

a) For negative values, it will take more computation time as it's an exponential function

(6) Softmax (Generally used in multi class classification)

$$\text{function} = S(x_i) = \frac{e^{x_i}}{\sum_{i=1}^{n} e^{x_i}}$$

For example, in flattened array we have 2, 3, 7, 10

$$\text{Probability of 7 will be} = \frac{e^7}{e^2 + e^3 + e^7 + e^{10}}$$

Similarly it will be calculated for all and the data will be classified as per the highest value.

(7) PReLU (Parametric ReLU)

$$\text{Function} = f(x) = \begin{cases} x & x > 0 \\ \alpha x & x \leq 0 \end{cases}$$

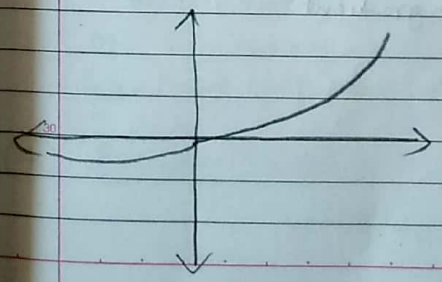
If $\alpha = 0.01$, it is leaky ReLU

If $\alpha = 0$, it is ReLU

If α is learnable parameter in negative direction, it is called PReLU.

(8) SWISH (Self Gated Function)

Output



$$\text{function} = x * \text{sigmoid}(x)$$

Advantages

a) Does not have vanishing gradient problem
b) Designed for LSTM networks where we need to train and memorize weights
c) It is smooth and zero centric
d) Decomposition is not happening to a low level like in sigmoid or ReLU functions.

(9) Maxout function

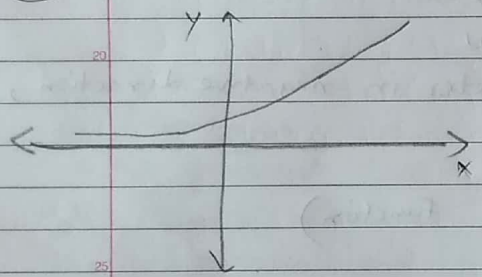
function = $\max(w_1 x_1 + b_1, w_2 x_2 + b_2)$

Advantages

a) Does not depend on predefined scenarios but depend on weights and biases
b) Computation time is low

Benefits of both ReLU and Leaky ReLU

(10) ~~Softmax~~ Softplus



$f(x) = \ln(1+e^x)$

Advantages

a) Smooth function, smooth gradient

LOSS FUNCTIONS

(1) L1 Loss Function (Least Absolute Deviation)

Function = $\sum_{i=1}^{N} |(y - \hat{y})|$

(2) L2 Loss Function (Least Squared Error)

Function = $\sum_{i=1}^{N} (y - \hat{y})^2$

L1 calculates the absolute deviation between actual and predicted value.
L2 calculates the residual squares between actual and predicted value

L2 has disadvantage in case of outliers as the same gets amplified by square terms i.e. a difference of 100 gets amplied to 10000 and so on..

(3) Huber Loss

Function $L(x, f(y)) = \begin{cases} \frac{1}{2}(y-\hat{y})^2 & y-\hat{y} < \delta \\ \delta \cdot |y-\hat{y}| - \frac{1}{2}\delta^2 & y-\hat{y} > \delta \end{cases}$

where δ is the threshold

This loss function tries to control the disadvantage of L2 where we can define the threshold above which losses will not get amplified due to outliers.

(4) Pseudo Huber Loss function

~~$L(x) = \delta^2(\sqrt{1 + (x/\delta)^2} - 1)$~~

$$L(x) = \delta^2\left(\sqrt{1+(x/\delta)^2} - 1\right)$$

(5) Hinge Loss (Used in classifier) (Support Vector Machine)

$$L(y) = \max(0, 1 - t \cdot y)$$

where t = Number of classes

~~Closer the y is to t, lesser the loss function~~

Closer the $t \cdot y$ is to 1, lesser the loss function value.

(6) Cross Entropy (Binary Classification)

$$\text{Function} = -\sum_{m=1}^{N} t_m(\log(y_m)) + (1-t_m)\log(1-y_m)$$

Either of one evaluates to zero, if $t_m = 0$ then $(1-t_m)\log(1-y_m)$ is calculated, if $t_m = 1$ then $t_m(\log(y_m))$ is calculated and value is always between zero and one.

(7) Sigmoid Cross Entropy

$$\text{function} = -\sum_{m=1}^{N} t_m\left(\log\left(\frac{1}{1+e^{-x}}\right)\right) + (1-t_m)\log\left(1 - \frac{1}{1+e^{-x}}\right)$$

(8) Softmax Loss function (Multi class classifier) (Cross Entropy)

$$\text{function} = \left(-\log\left(\frac{e^{xi}}{\sum_{i=1}^{N} e^{xi}}\right)\right)$$