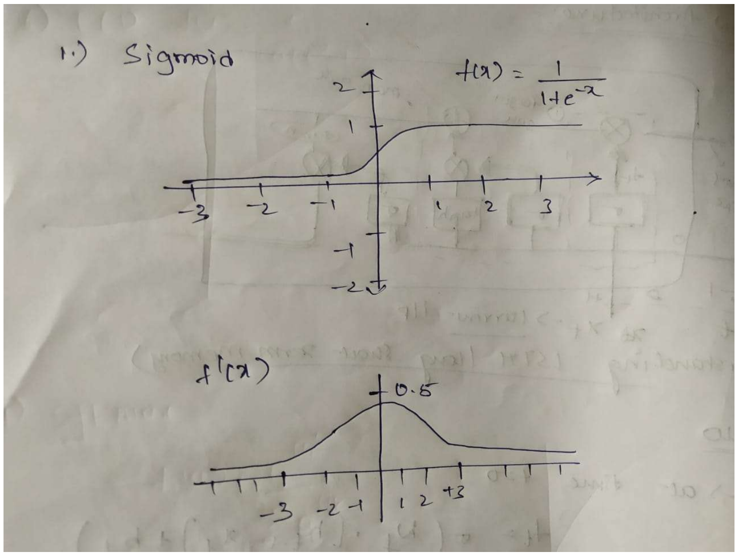
ACTIVATION FUNCTIONS:-

1. SIGMOID ACTIVATION FUNCTION

The sigmoid function is a non linear activation function wherein the output of the function lies in between 0 and 1.The drawback of this is that it has flat edges at the end which means that the gradient or the slope of the line at the end is 0 hence it has vanishing gradient problem during back propagation. The gradient values or only available from -3 to 3 after which it's a problem i.e. its constant

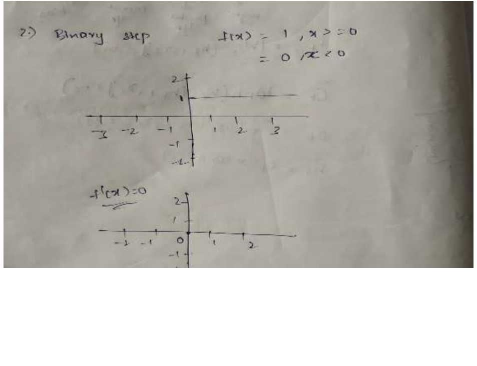


2. BINARY STEP FUNCTION-

In a binary step function the neurons get activated in a step fashion as in if the input is greater than 0 i.e. if it is positive the neuron is activated and the output is one on the other hand if the input is less than 0 or a negative number then in such case the neuron doesn't gets activated and the output is 0.One of the drawback of this function is that the derivative of this function is 0 which means during backpropagation while the optimizer comes into place the weight updating doesn't take place

$$f(x) = 0; if(x < 0)$$

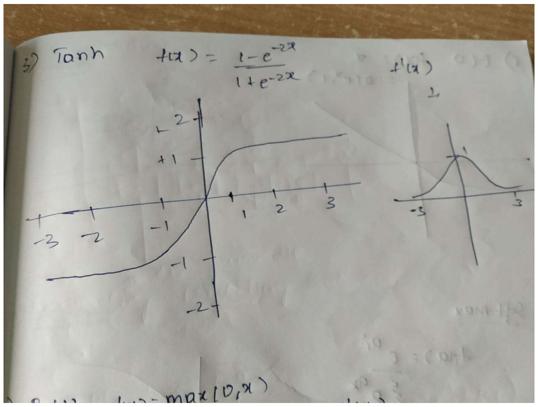
$$= 1\,; if(x > 0)$$



3.TANH ACTIVATION FUNCTION(Hyperbolic Tangent function)

The tanh activation function has the output centred around -1 to 1 , has an edge over the sigmoid activation function but it too still suffers the vanishing gradient problem which becomes a problem in the back propagation stage.

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

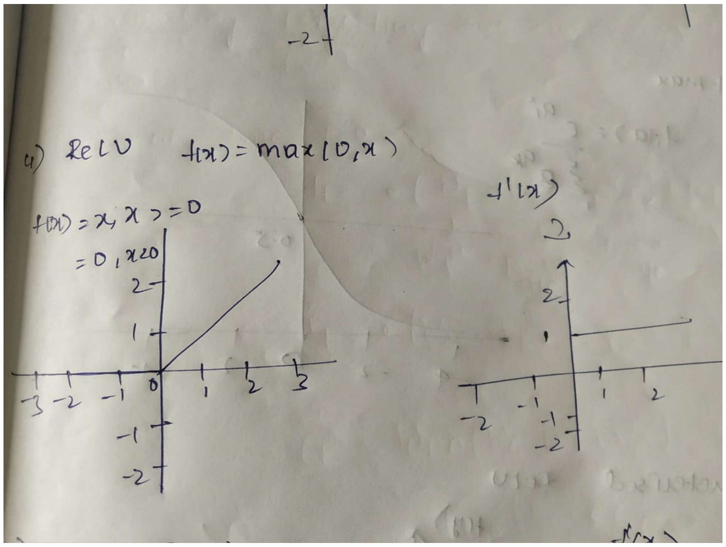


4.ReLU ACTIVATION FUNCTION

The ReLu (Rectified linear unit) function is one of the most popularly used activation function and it takes the maximum of the input that it only activates certain neurons.

$$R(x) = \max\,(0, x)$$

Hence it doesn't have the vanishing gradient problem but it is only used in the hidden layers and not in the output layers. And it completely neglects the inputs which are negative which is a major drawback of ReLu



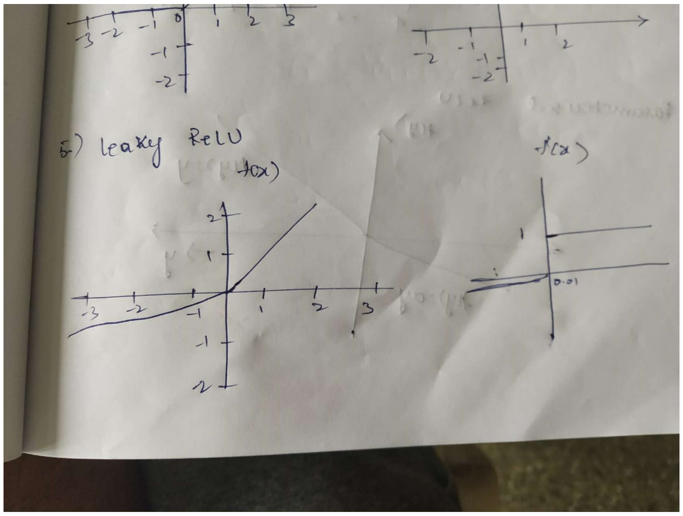
5.Leaky ReLu ACTIVATION FUNCTION

The drawback which was present in the previous ReLu is now answered by providing a small component of x in the negative region too. i.e. x*0.01 which helps in preventing the neurons to be completely dead.

$$f(x) = x \quad if\ x > 0$$

$= ax \quad if\ x < 0$ (where a=0.01)

Though it solves the problem of dead neurons it still doesn't completely justifies it.

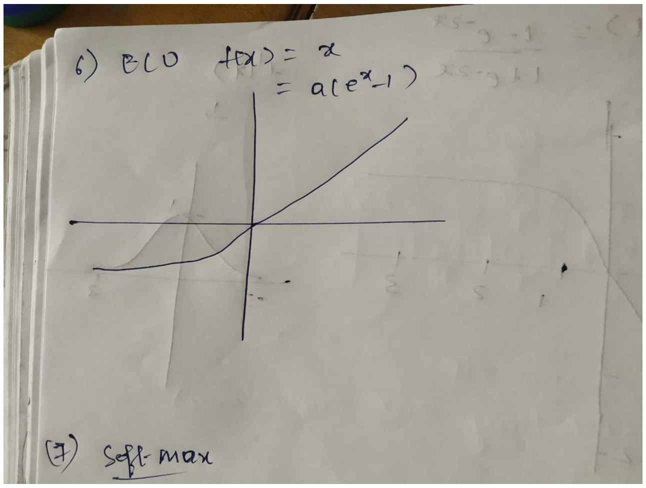


6.ELU ACTIVATION FUNCTION(Exponential Linear Unit)

Similar to the leaky relu which is used but instead uses a logarithmic curve on the negative side which leads to no gradient clipping and the mean of the output is close to zero.

$$f(x) \quad = x \qquad if\ x > 0$$

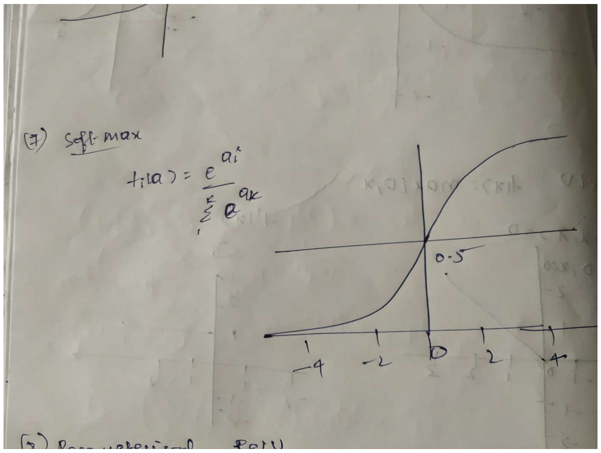
$= a(e^x - 1)$ if x < 0



7.SOFTMAX ACTIVATION FUNCTION

As in sigmoid activation function which is used for binary classification problem where in it has its output only concentrated as a 0 or a 1 and in a similar fashion the soft-max function which is present can be said as a group or a bunch of sigmoid functions which actually can be used for the multi class classification where there are multiple outputs for the given function.

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

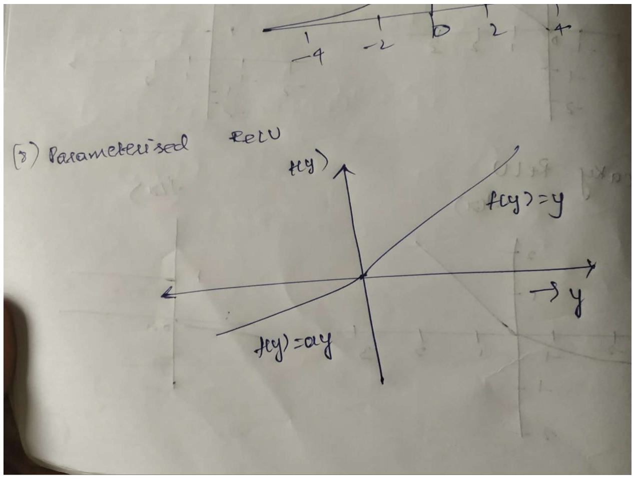


8.PARAMETERIZED ReLu ACTIVATION FUNCTION

The Parameterized Relu function is another variant of ReLu function where in this also solves the problem of dead neurons. Like the graph which is present down there in the left side of the axis hence instead of a 0.01 which is present in the leaky Relu this one replaces it with an ‘a’ which is a learnable parameter unlike the leaky Relu.

$$f(x) = x\ for\ x > 0$$

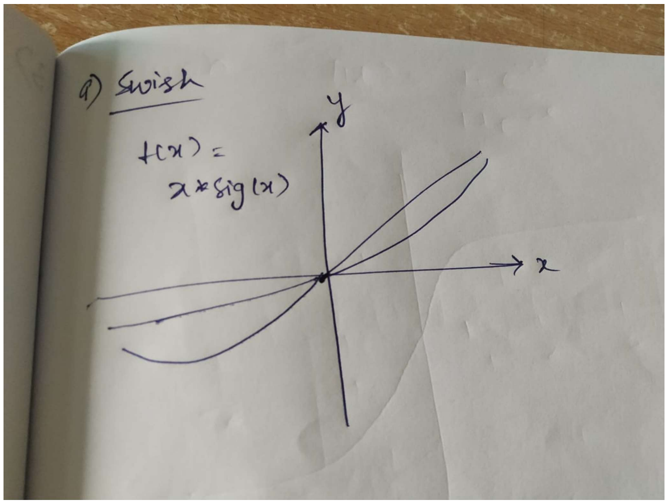
= a*x for x < 0(where ‘a’ is a learnable parameter)



9.SWISH ACTIVATION FUNCTION

The swish activation function is not so familiar activation function which was founded by google and it ranges from -∞ to ∞ .

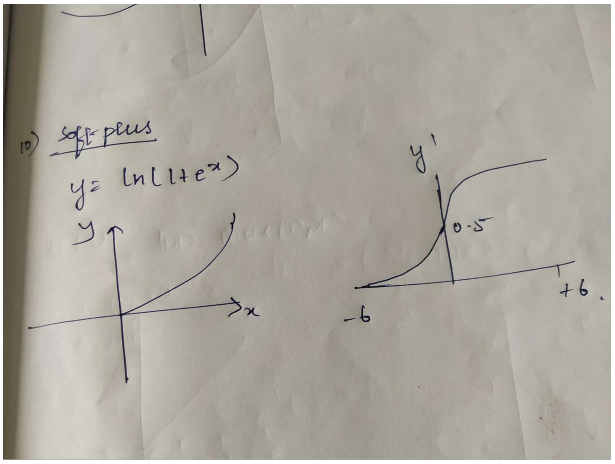
$$y = sigmoid(x)$$



10.SOFTPLUS ACTIVATION FUNCTION

The soft plus activation function unlike other activation functions which have upper and lower bounds the outputs of the soft-plus activation function range in between (0,∞). It is similar to the ReLu activation function but has its difference in the differentiability of x at 0.

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



LOSS FUNCTIONS

(I) REGRESSION LOSSES

1.MEAN SQUARE ERROR(MSE)

It is the average or the mean of the squared values i.e. of the original and the predicted values.

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

2.MEAN ABSOLUTE ERROR (MAE) [L1 LOSS]

It computes the mean of the absolute difference between the predicted and the original values.

$$MAE = \frac{\sum_{i=1}^{n} \mid y_i - \hat{y}_i \mid}{n}$$

3.MEAN BIAS ERROR

It is similar to the mean absolute error but instead of computing the absolute values it just computes the mean of the actual and the predicted values though this is not practiced well in machine learning this helps us to know whether the model has a positive bias or a negative bias

$$MBE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n}$$

4.HUBER LOSS

The huber loss combines the properties of both the MSW and MAE and is quadratic for smaller losses otherwise is linear.

Quadratic

$$L_\delta = \begin{cases} \frac{1}{2}(y - f(x))^2, & if\ |y - f(x)| \le \delta \\ \delta|y - f(x)| - \frac{1}{2}\delta^2, & otherwise \end{cases}$$

Linear

(II) CLASSIFICATION LOSSES

5.HINGE LOSS

The hinge loss is most commonly seen in the support vector machines (svms) where it is used as a maximum margin classifier as in how correctly the object is classified. A hinge loss of 0 indicates that the model has a very good accuracy and vice-versa.

$$SVMLoss = \sum_{j \neq y_i} max(0, s_j - s_{y_i} + 1)$$

6.CROSS ENTROPY LOSS

The cross entropy loss is the one which is most commonly used in any classification problems the loss increases as the model deviates from its original model .

$$CrossEntropyLoss = -(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i))$$

MULTI CLASS CLASSIFICATION LOSSES

7.Multi Class Cross Entropy Loss-

The multi-class cross-entropy loss is a generalization of the Binary Cross Entropy loss. The loss for input vector x and the corresponding one-hot encoded target vector y.

8. Psudo huber loss

The psudo huber loss combines the properties of both the MSW and MAE and is quadratic for smaller losses otherwise is linear

9.Softman Cross Entropy loss

Cross entropy indicates the distance between what the model believes the output distribution should be, and what the original distribution really is. Cross entropy measure is a widely used alternative of squared error.

H(y,p)=−∑iyilog(pi)H(y,p)=−∑iyilog(pi)

10. KL-Divergence

The Kullback-Liebler Divergence is a measure of how a probability distribution differs from another distribution. A KL-divergence of zero indicates that the distributions are identical.