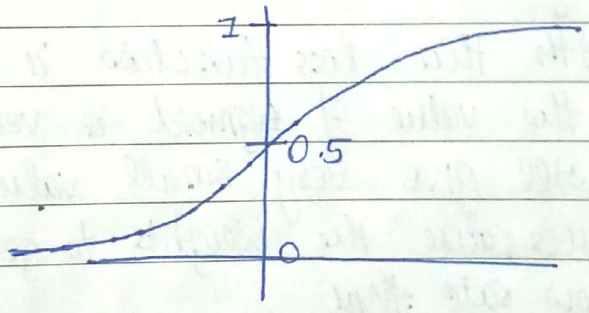
Abhi

DATE

ML- Assignment 3

Abhishek Agarwal
- 2016126

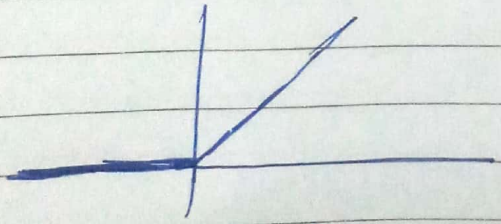
Q.3 X is unable to train the model successfully while using sigmoid activation because of the vanishing gradient problem of sigmoid activation, as the gradient becomes smaller and smaller and convergence cannot be achieved quickly.
The value never becomes equal to 0 or 1 and only tends to 0 or 1.
Sigmoid gradient becomes increasingly small as the value tends to high negative value, and it gets very close to 1 as the value becomes high positive value.
This can also be seen from the graph of sigmoid:



This gradient does not let training properly because it is not propogated in an appropriate ~~way~~ way as it vanishes (goes close to 0)

RELU activation function defined as:

$$Relu(x) = Max(0, x)$$



RELU can overcome this problem because it does not

DATE ___/___/___

2. have the vanishing gradient problem.
For input values >0, the gradient does not vanish to zero and has a constant value.
Thus we do not experience the problem of vanishing gradient

Also, relu has a spasses representation. i.e. for inputs <0, the gradient value is 0 and hence it is more spasse as compared to dense representation in case of sigmoid.

Batch Normalization technique can be used to tackle this problem. Here, each training batch is normalised seperately and all the layers learn independently of one another.
∴ Higher learning rates can be used while training

- Normalizing weights with zero can lead to problems.
This is a problem because since all weights are equal, all neurons will follow the same update rule.
The output of the first layer i.e. $wx+b$ will be zero as w, b both are initialized to 0.
A better method would be to initialize the weights randomly.

- Cross entropy loss function is a better one as compared to MSE in case of classification problems.
Cross entropy works purely on probability values and when we have 2 labels i.e. 0 and 1, we can get a true measure of how far is the result from the actual one.

Q.2

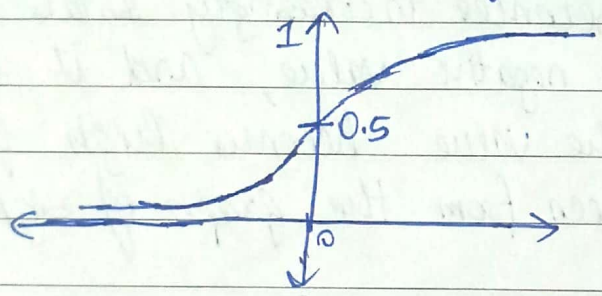
Cross entropy loss for M classes is given by:

$$L = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

where $y_{o,c}$ takes values 0 or 1 depending on whether it is in the correct or incorrect class.

In the case of quadratic loss, the loss contains "(sigmoid)(1-sigmoid)" term, that is multiplied by the derivative in the loss function.

As we know, the graph of sigmoid is:



The problem with this loss function is that, whenever the value of sigmoid is very near to 0 or to 1, it will give very small values.

These small values cause the weights to get updated at a very slow rate.

This causes a learning slowdown.

In the case of cross entropy loss, the term involved is the softmax for that particular class and does not cause slowdown in learning by any means.

Q.5 Consider a neural network having linear activation function.

For an input $\bar{x}$, the output is of the form

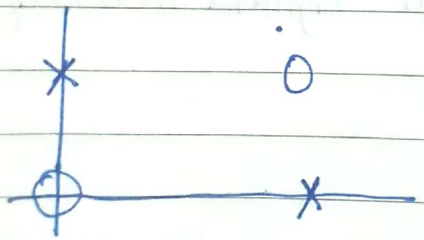
$$W_1^T x + b_1$$

For multiple such layers, the overall output will be:

$$Out = W_n^T (W_{n-1}^T \dots ((W_1^T(x) + b_1) + b_2) + \dots) + b_n$$

This is the combination of various linear layers, and can directly be represented as a ~~single~~ combination of single W and b.

∴ Since, XOR has a non linear decision boundary:



We will not be able to model XOR using multiple layers with linear activation.
We can achieve it by introducing non linearity or thresholding outputs to certain values.

This is analogous to a single layer perceptron. as explained above.