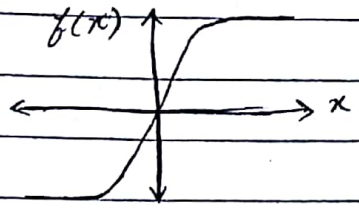
ML Assignment 3
Theory

Swathi S Nath
2016271

Date

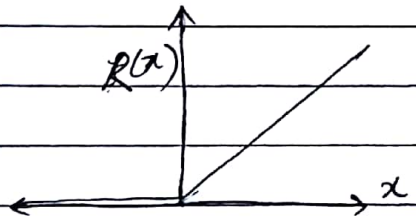
Q1 Sigmoid activation function is given by:

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



RELU activation function is given by:

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



The possible reason why one cannot train a model successfully using sigmoid activation function could be the vanishing gradient at high absolute values of the input. However, since $f(x)$ only tends to 0 at very ~~low abs~~ high negative values and only tends to 1 at very high positive values, it is never exactly 0 or 1. As a result of this, sigmoid activation always yields non zero values hence making it a dense representation.

RELU can overcome both these issues. When the input to RELU is positive, the vanishing gradient problem is tackled as even at high positive values, the gradient does not tend to 0 (vanish). When the input is negative, $R(x) = 0$ hence this tackles the density problem by providing a sparse representation which are considered better for training neural networks.

A suitable preprocessing technique to counter covariance shift (amount by which hidden units' values change) is batch normalization. Batch normalization not only reduces covariance shift, but also enables each layer to learn independent of other layers. As a result, higher LRs can be used while training. It also helps reduce overfitting by adding some amount of noise to each layer's activation.

While initializing the weights in the neural network, if they are all initialized as 0, firstly, no learning will happen since all neurons will follow same updations since all weights are initially equal. Also, they may get caught at local minimas ∴ the network should be provided several different values in the beginning. ∴ weights should be initialized randomly.

For a classification problem, cross entropy is a better loss function since true labels are in the form of 0s and 1s where 1s indicate the correct label. Cross entropy loss given by $\sum y_i \log(\hat{y}_i)$ ignores all 0s and only considers how far the result is from the desired class. MSE is more suitable for regression problems where we need to find the exact value for which difference b/w true and obtained value is a useful function for loss measurement.

Q2 While using the quadratic squared loss, the weight adjustments during backpropagation contain the sigmoid derivative term (sigmoid) × (1 − sigmoid). Clearly this results in very small values whenever the value of sigmoid is very near 0 or 1. This hence causes a very small update in weights under these conditions. ∴ there is a learning slowdown. However, when cross entropy loss is used, $L = \sum y_i \log(\hat{y}_i)$, the only term contributing to loss is softmax value for actual class. Hence it means how close the output for the actual class is to 1. This doesn't pose any

Date

limitation of ReLU learning.

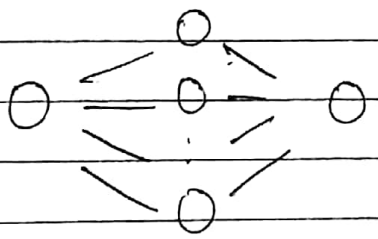
Q3 No matter however deep we make a neural network, XOR requiring a non linear decision boundary, (or multiple linear decision boundaries), cannot be modelled using only linear activation. The output will be of the form:

$$out = w_n(w_{n-1}(w_{n-2} \cdots (w_1 x + b) + b) + b \cdots + b)$$

which is still linear, since composition of linear functions is again linear. Such a model can hence only learn a linear decision boundary unless and unless some non linearity is introduced.

This model is analogous to the single layer perceptron since multiple hidden layers with linear activations are basically just another linear equation.

eg



linear activation
$\Rightarrow f(x) = x.$

$$\sum f\left(\sum w_i x_i + b_i\right) + b' = \sum \left(\sum w_i x_i + b_i\right) + b$$

$$= \sum_i w'_i x'_i + b'_i$$