Name: Fahad Iqbal **ID:220708016** **ML Assessment 1**

Assignment 1 Part 3(Neural Network)

Q1. Why is it important to use a random set of initial weights rather than initializing all weights as zero in a Neural Network? [2 marks]

The main reason for using random weights rather than zero can be covered by looking into the symettry problem.This is where if each weight is initially zero then each neuron will perform the same calculation in forward propagation.This prevents us from learning anything diverse and results in a network that just repeatedly learns the same patterns.If we have a random set of weights then we solve this problem and thus get different outputs for different neurons and when we do backpropagation we can get different gradients. These gradients then allow us to learn diverse features and use the network in a useful way.For gradient based methods random initialisation is especially important to adjust weights in the correct directions so we can converge on the solution

Q2) How does NN solve the XOR problem

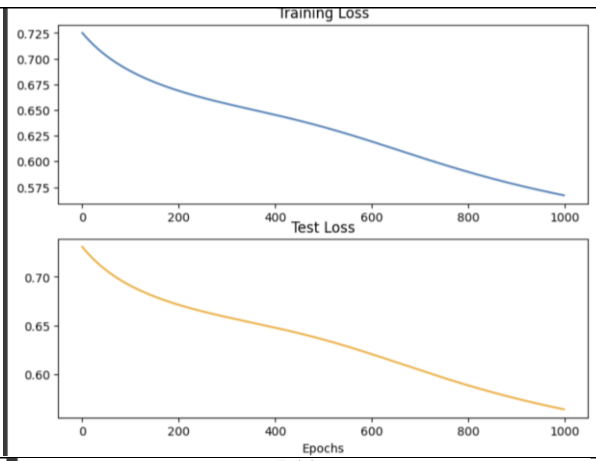
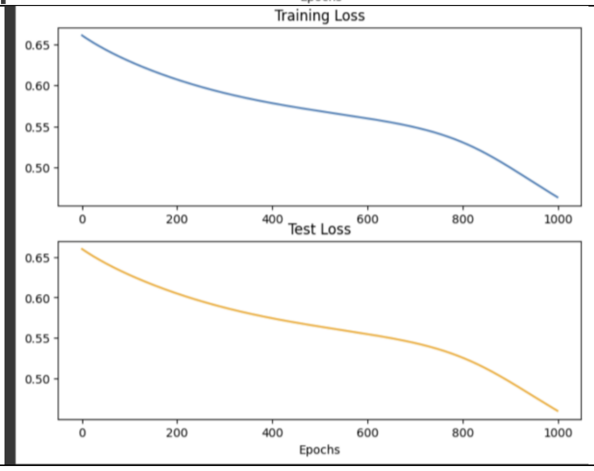
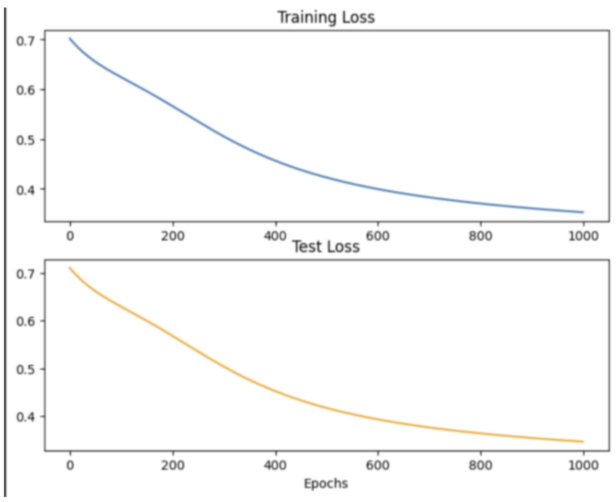
In the previous assignments we covered how logisitic regression cannot solve the XOR problen due to it only being able to create linear decision boundaries. Single Layer Neural Networks suffer from the same issue but this can be solved by adding a hidden layer. Hidden layers in a neural network enable us to create non linear boundaries.

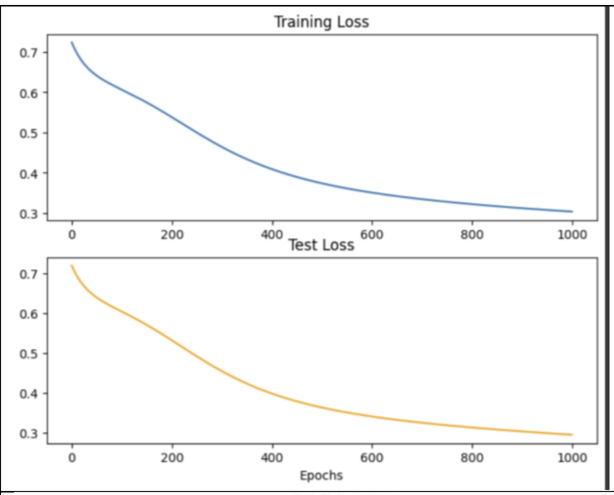
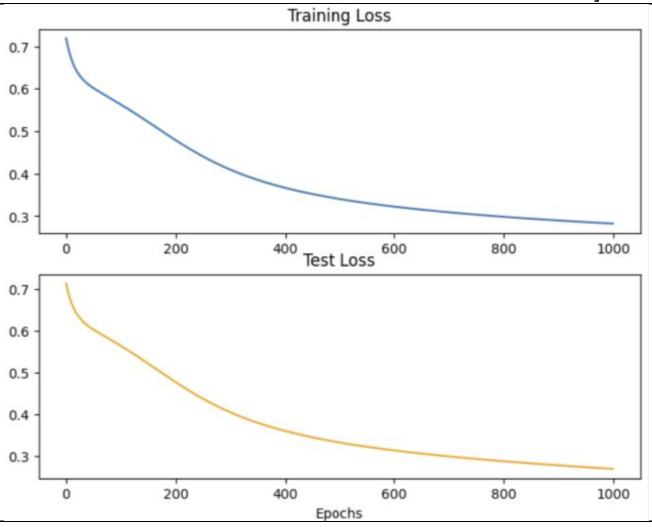
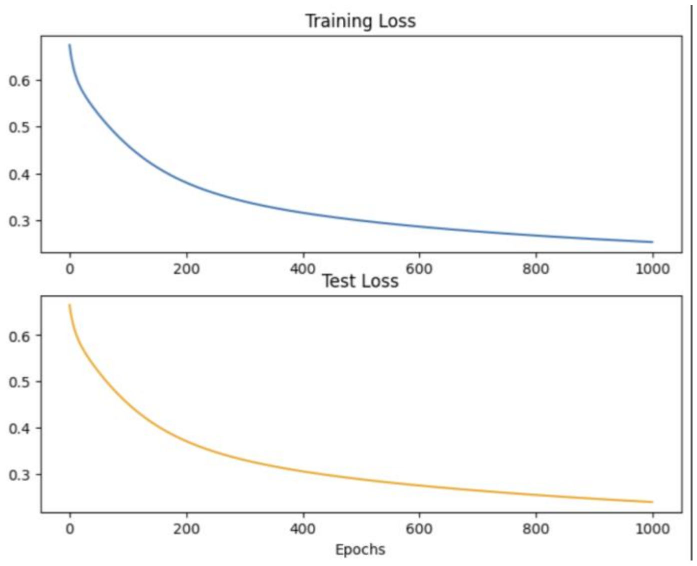
The XOR can be represented (A AND NOT B) OR (NOT A AND B), this can be modelled within a hidden layer . One neuron would learn to activate (A and not B) and th other would do the same for (Not A and B). The OR operation could be done on them to get the XOR.

Functionally this is performed by activation functions like sigmoid or ReLu) which when applied to each hidden neuron output enables us to solve the problems that are non linear.

Name: Fahad Iqbal ID:220708016 ML Assessment 1

Q3. Explain the performance of the different networks on the training and test sets. How does it compare to the logistic regression example? Make sure that the data you are refering to is clearly presented and appropriately labeled in the report. [8 marks]

Hidden Neurons	Iterations/epochs	Minimum train cost	Minimum test cost	Graph
1	1000	0.5670	0.563	
2	1000	0.4639	0.4602	
4	1000	0.3530	0.3464	

8	1000	0.3040	0.2941	
16	1000	0.2822	0.2688	
32	1000	0.2527	0.2387	

Analysing the table above we can observe that as we increase the number of neurons in the hidden layer, the training and test costs generally decrease,this is because increasing the hidden neurons allows the model to learn more complex patterns and relationship.

Firstly, we observe that for the networks with 1 and 2 hidden neurons respectiely the costs were higher with 1 having training and test costs of 0.5670 and 0.563 respectively.Likewise when we have hidden layer of 2 we get the train and test cost of 0.4639 and 0.4602. This indicates that the model cannot yet capture the complex patterns and we are underfitting.

Moving on to higher ammound of hidden neurons like 8 and 16 neurons we see the training and test loss are low but the graph does not indicate any overfitting yet. For example, hidden neuron 16 has train and test costs of 0.2822 and 0.2688.The gap between training and test losses is not too high and they are relatively close here.

Moving on to the highest number of hidden neurons it is likley we will start to overfit soon and the 32 hidden neurons model generalises less well compared to 8 and 16.Comparing to logistic regression from earlier the curves in logistic regression are much smoother and has a faster loss but limited convergence, plateauing at a higher loss, indicating its linear limitation.To conclude,logistic regression was unable to model more complex non linear relationships but with the addition of hidden layers the MLP is able to do so .