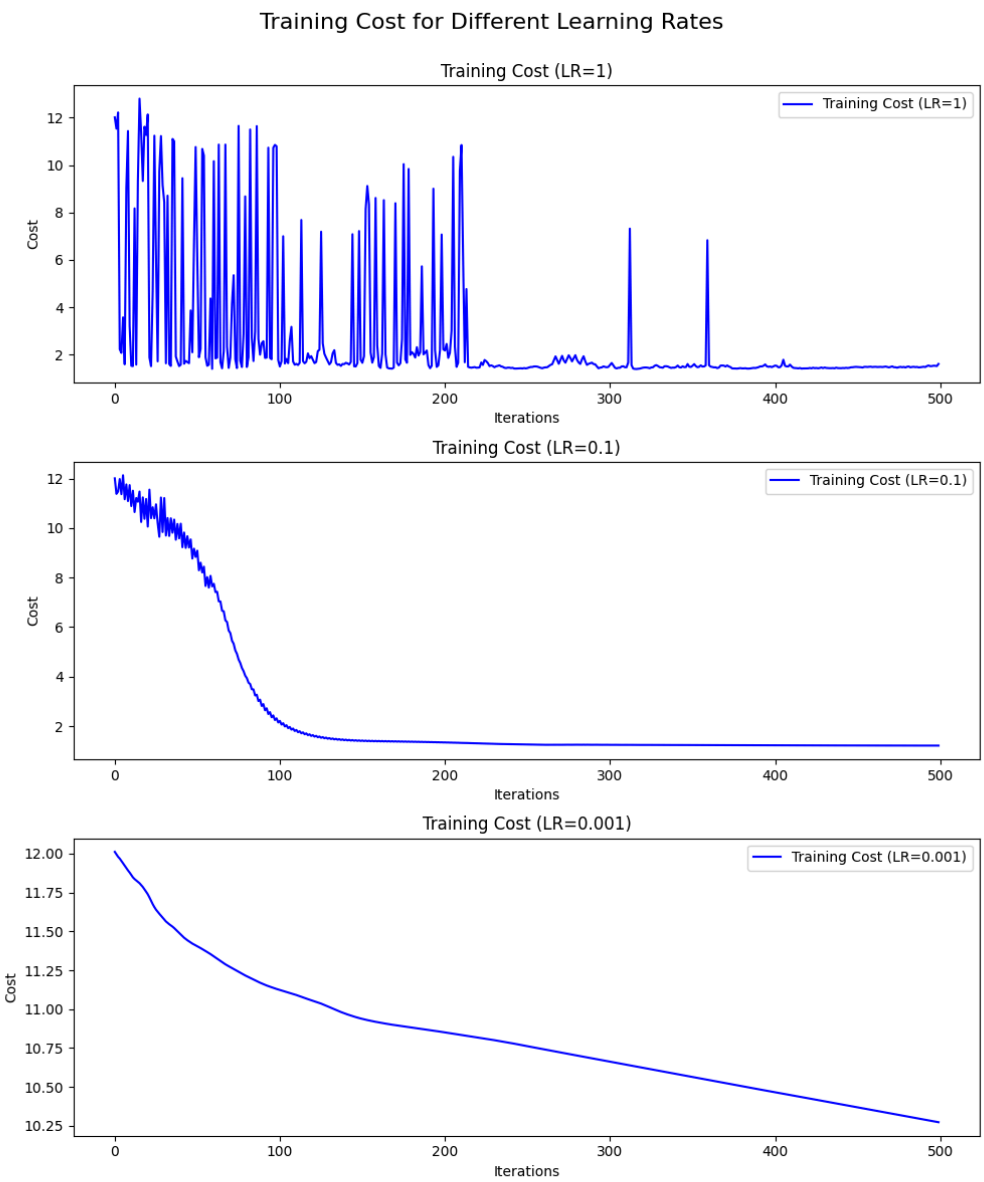
Assignment 1 - Backpropagation

210266G - Kahapola K. V.

Learning Rate controls the size of the steps that the optimization algorithm takes while attempting to find the minimum of the loss function. In this assignment, three neural networks were trained using learning rates of 1, 0.1, and 0.001 to observe the impact on model performance. The results of training and testing costs, as well as accuracy metrics, were visualized to understand the effect of different learning rates.

This report presents the graphs from these experiments and discusses the influence of learning rates on training and testing performance, with a focus on their effect on test accuracy.

1. Training Cost w.r.t Iterations

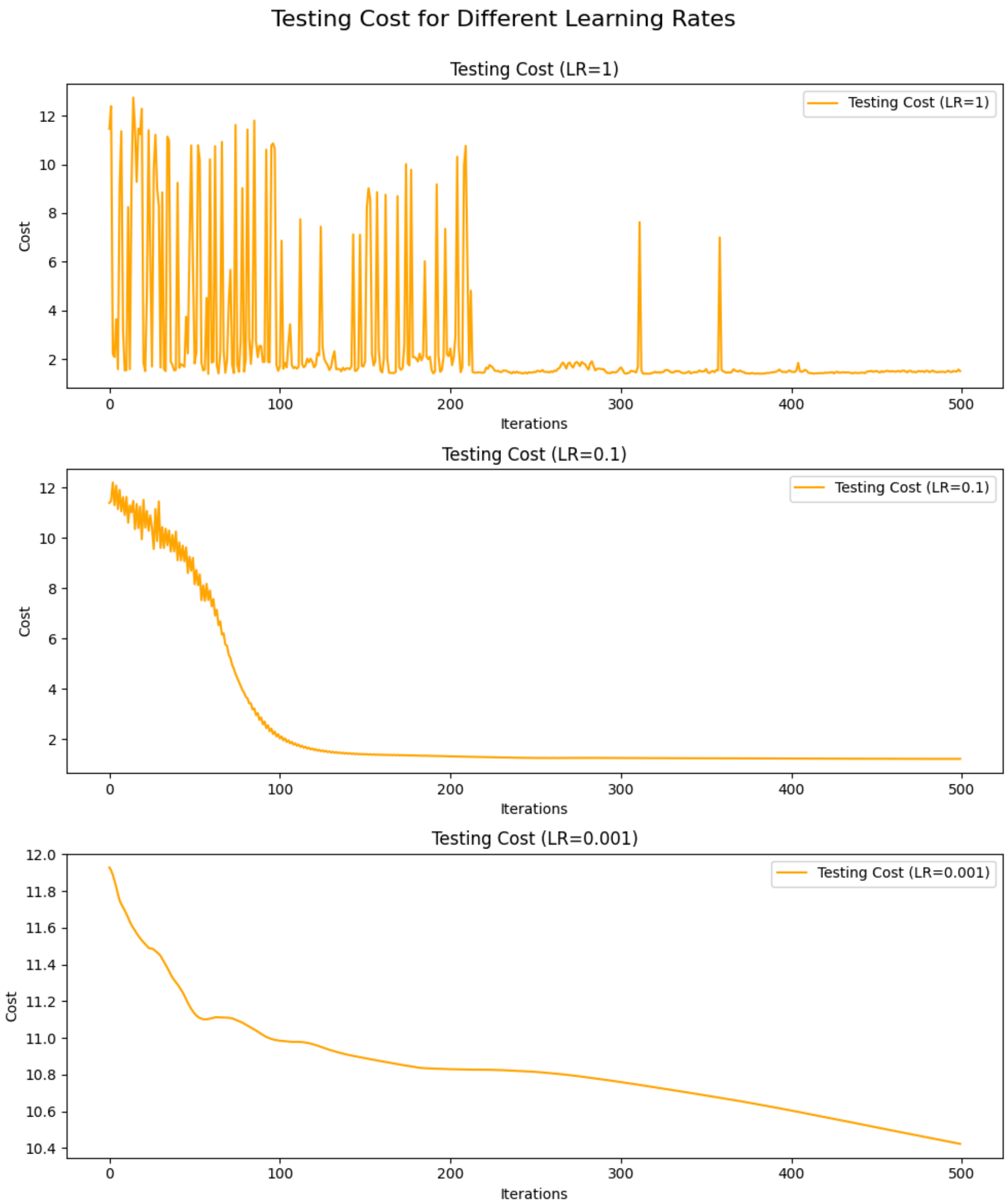


Learning Rate (1) - The graph shows how much the training cost varies. This suggests that the optimiser is doing too many steps and the learning rate is too high, which is leading to instability in the convergence process. The cost decrease is unpredictable because the model regularly overshoots the ideal solution.

Learning Rate (0.1) - The optimiser is approaching the minimum at a steady pace as the training cost drops smoothly. It seems that this learning rate is suitable for making consistent progress towards cost function minimisation.

Learning Rate (0.001) - The cost of training does go down, but very slowly. The model weights are only slightly updated as a result of the low learning rate, which causes the cost function to decrease slowly.

2. Testing Cost w.r.t Iterations

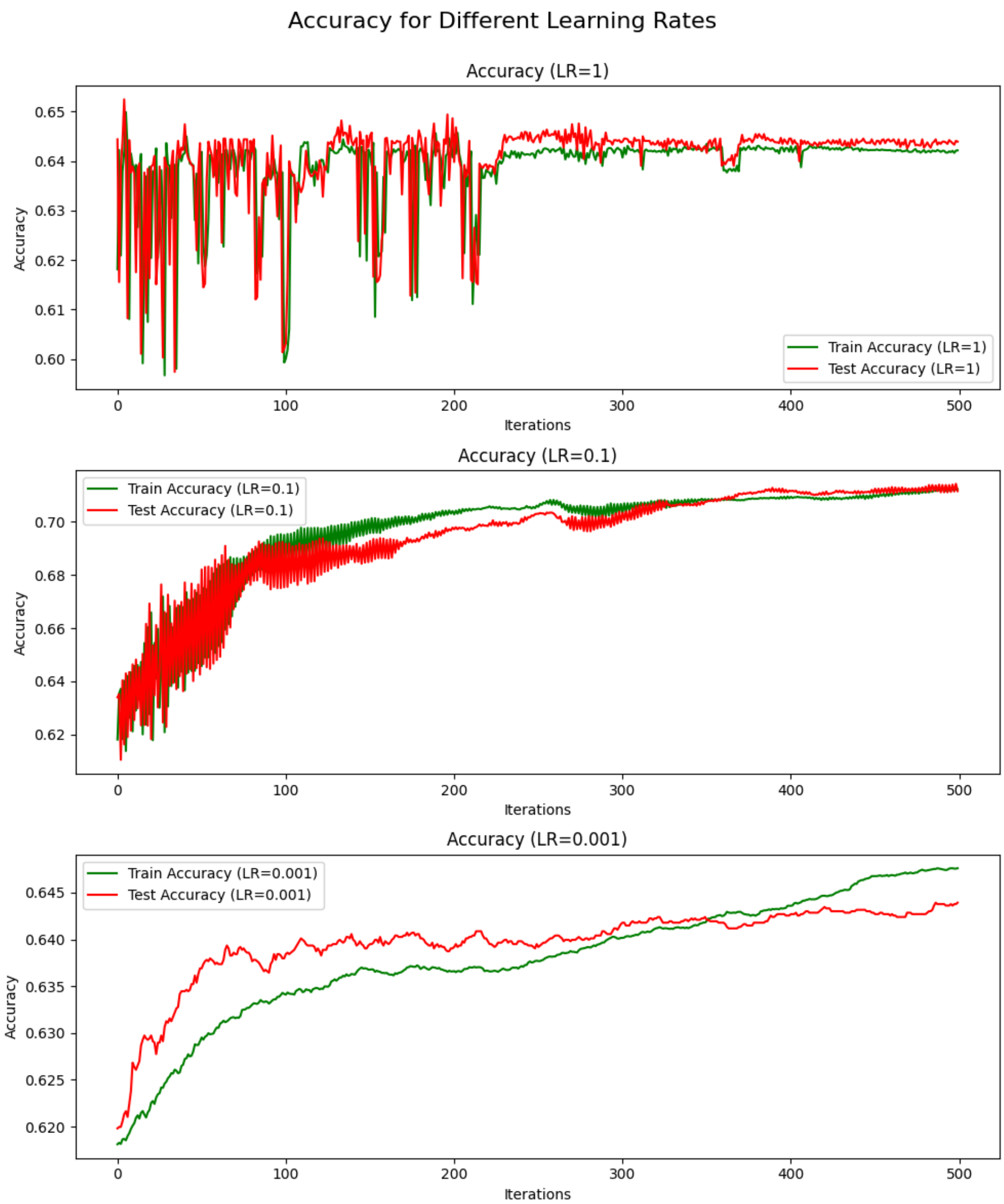


Learning Rate (1) - Similar unpredictability is seen in the testing and training costs. This suggests that there is a lack of good model generalisation to the test data. Poor performance on unseen data is the result of overadjustments brought on by the high learning rate.

Learning Rate (0.1) - Like training costs, testing costs decline gradually. The fact that the testing cost decreases steadily over time indicates that the model generalises successfully at this learning rate.

Learning Rate (0.001) - The cost of training does go down, but very slowly. The model weights are only slightly updated as a result of the low learning rate, which causes the cost function to decrease slowly.

3. Accuracy Score w.r.t Iterations



Learning Rate (1) - The test accuracy is very inconsistent, with both the train and test accuracy fluctuating dramatically. This is a blatant indication that the learning rate is very high, which is why the model performs differently on successive rounds.

Learning Rate (0.1) - Accuracy scores for the test and the train rise gradually and stay relatively close to one another. This suggests that the model is learning efficiently and making good generalizations to new data.

Learning Rate (0.001) - In comparison to 0.1, the accuracy ratings rise steadily but significantly more slowly. Although the model is improving steadily, many more iterations are needed before it reaches a comparable level of accuracy.

The learning rate plays a crucial role in determining the model's ability to generalize well to unseen data, which directly influences test accuracy. A high learning rate, such as 1, causes the optimizer to take large steps during gradient descent, leading to overshooting the optimal solution. This results in erratic performance and poor test accuracy, as the model fails to converge effectively.

On the other hand, a moderate learning rate, like 0.1, strikes a balance by allowing the model to converge steadily without overshooting, leading to improved test accuracy as it generalizes better. A low learning rate, such as 0.001, results in slower learning, with the model making minimal updates. While this prevents overshooting, the slow convergence leads to suboptimal test accuracy in fewer iterations, requiring more time to reach a higher accuracy level. Hence, finding the right learning rate is key to achieving good test accuracy while balancing learning speed.