CS3631 - Deep Neural Networks

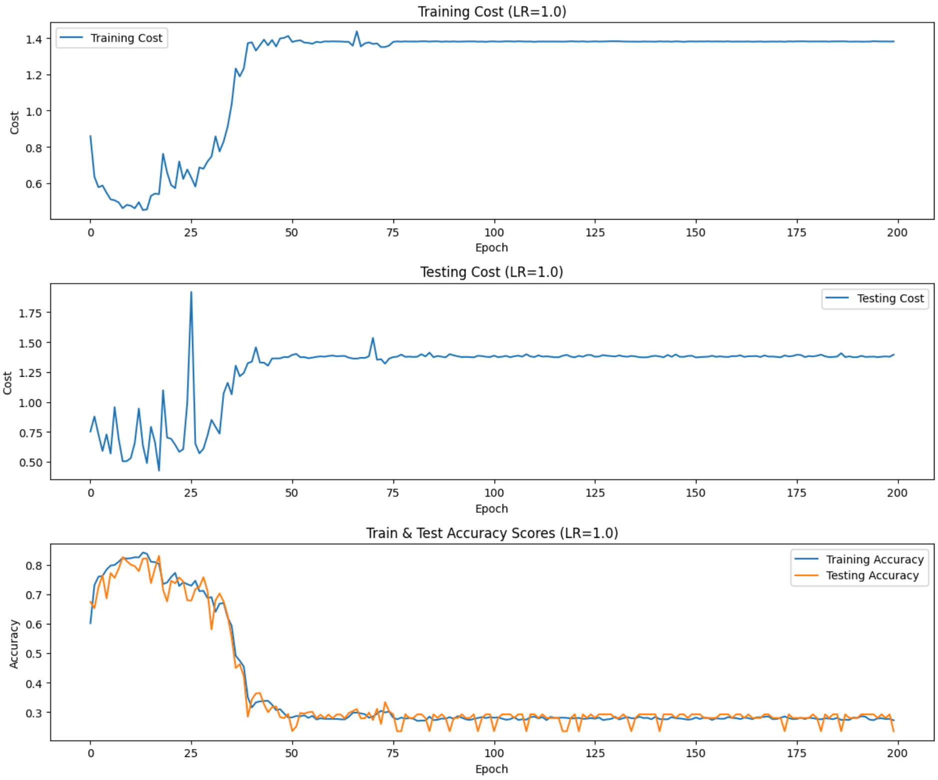
Assignment 1: Back propagation

210173T | GALLAGE D.

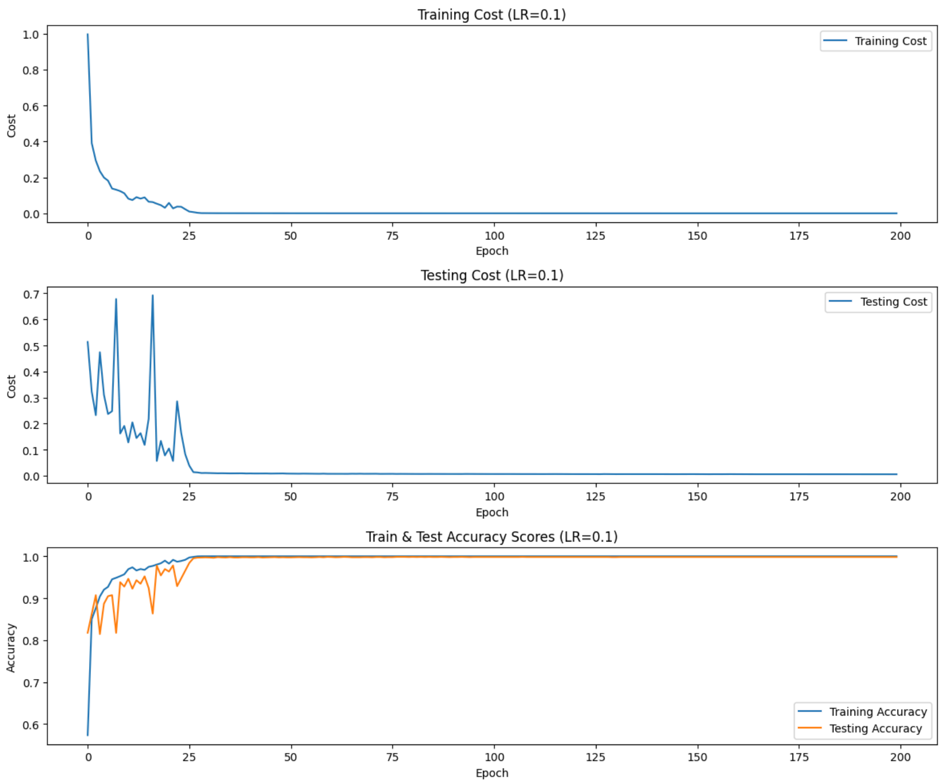
Task 2 - Report

The neural network was initialized with random weights and biases in each layer. The accuracy and cost of predictions for both the training and test datasets were evaluated over 200 iterations using three different learning rates: 1.0, 0.1, and 0.001. Below are the results and analysis for each learning rate.

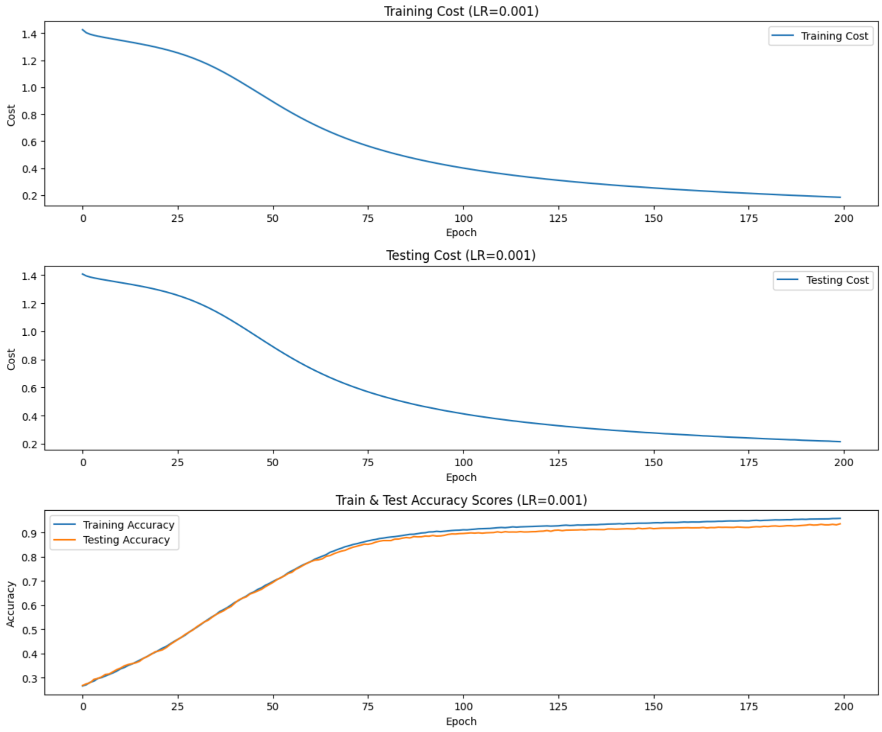
Learning Rate: 1.0



Learning Rate: 0.1



Learning Rate: 0.001



Discussion

At a high learning rate of 1.0, the model showed considerable volatility, with sharp spikes in both training and testing costs. It often overshot optimal points, leading to inconsistent performance and erratic accuracy improvements. The large step sizes made it difficult for the model to converge properly, resulting in unstable outcomes.

The learning rate of 0.1 offered the best balance between speed and stability. Both training and testing costs steadily decreased, with rapid accuracy improvements early on, followed by sustained high performance. This rate allowed the model to learn quickly without overshooting, leading to consistent convergence and strong results.

At a learning rate of 0.001, the model showed slow but steady progress. The costs decreased gradually with no significant fluctuations, but the smaller step sizes resulted in slower learning. Accuracy improvements were limited within the 200 iterations, indicating that this rate, while stable, was too conservative for achieving fast, meaningful progress.

Conclusion

The experiment demonstrated the importance of selecting an appropriate learning rate when training neural networks. A high learning rate, such as 1.0, causes instability and hampers convergence, leading to poor and unpredictable results. On the other hand, a low learning rate, such as 0.001, though stable, slows down the learning process to the extent that the model fails to reach high performance within a reasonable number of epochs. The optimal learning rate in this case was 0.1, striking a balance between speed and stability, enabling the model to quickly converge to low-loss values while maintaining high accuracy. This experiment underscores the need for careful tuning of the learning rate to ensure efficient and effective neural network training.