ECE542 – Backpropagation Assignment

Prathamesh Pandit
Student ID: 200311152
Unity ID: pppandi2

***Abstract* — This report contains information regarding the final network architecture and hyperparameter values that were selected for the multilayer perceptron, along with information about the performance of the model on training, validation and testing data.**

Keywords — neural networks, backpropagation, multilayer perceptron, hyperparameter tuning

I. Network Structure and Hyperparameters used

Upon training the network with the default values for hyperparameters i.e. number of nodes being [784, 20, 10], epochs = 100, mini-batch size = 128 and learning rate = 10^{-3} , our model provided us with training accuracy of 96.86%, validation accuracy of 90.53% and test accuracy of 90.14%.

A. Initial Selected Architecture:

Upon experimenting multiple network layers, I was able to generate a better model with two hidden layers. Although the performance of the network was not significantly better than a single hidden layer network, I noticed that having more than two hidden layers was leading to overfitting. My initial shortlisted architecture had the following number of nodes in each layer: [784, 60, 20, 10]

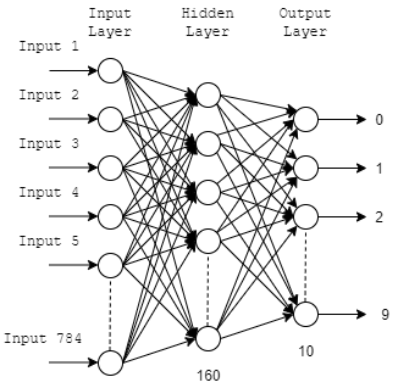
As I increased the complexity of the network, I noticed that by increasing the epochs slightly, I was able to achieve better performance. Therefore, I decided to set total epochs as 115 in the current architecture with other hyperparameters. Having mini-batch size of 64 helped improve the time required for model training.

Learning rates of the order of 10^{-4} were a bit too slow in converging and this was visible from validation loss plot, which was steeper for 10^{-3} values. Also, learning rate value in the order of 10^{-2} did not converge at the minima, and for higher values, I observed divergence. I found 10^{-3} to be a good value for the learning rate for the current network.

This architecture provided a Training Accuracy of 98.9%, a Validation Accuracy of 91.62% and a Test Accuracy of 91.24%.

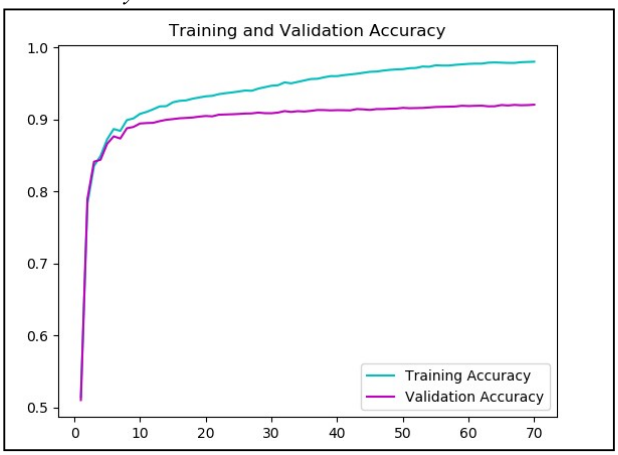
B. Final Network Architecture

A reason why I hesitated from using an architecture with a single layer was because adding more nodes to the middle layer significantly increased the time required to train the model. I finally trained my model with the following nodes in each layer: [784, 160, 10]. I trained the model for a total of 70 epochs. Mini-batch size was set to 64 and learning rate was set to 10^{-3} .



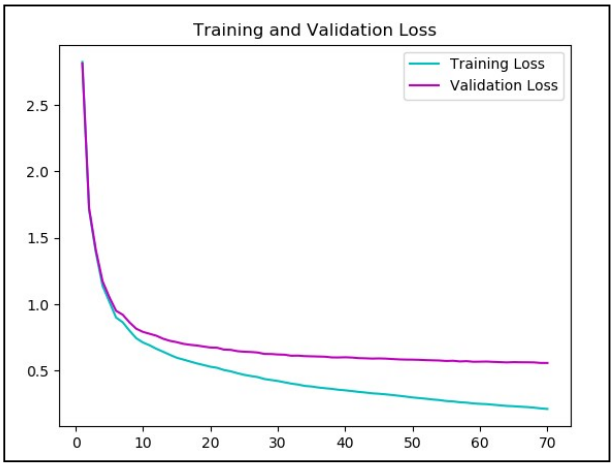
II. Learning Curves on Training and Validation Set

A. Accuracy Curves



At the end of the network's training, a Training Accuracy of 98% and a Validation Accuracy of 92.4% were observed.

B. Loss Curves



At the end of the network's training, a Training Loss of 21.13% and a Validation Loss of 55.77% were observed.

III. Final Test Accuracy

Our Model was able to achieve a Test Accuracy of 91.54