Mansai Conner
1231264114
CSE 575

Part 1

1. Introduction

In this first part of the project, I ran the baseline code that was given, then changed some of the parameters of the kernel and ran it a few more times. After this I plotted each of the errors for each epoch. My hypothesis was that having a larger kernel would help distinguish the different features.

2. Testing the Code

After the initial test with the default code, I got a test accuracy of about 98.3% after the 12 epochs. I noticed a large jump from the first to the second epoch in training accuracy. This makes sense because it would need to iterate through to train the model. Once after the first epoch each jump to the next epoch was very small. When I increased the kernel size to 5x5, the test accuracy was about the same at 98%. The error graph looked pretty similar as well. Then I started messing around with different values for the number of filters and was getting about the same result. This took a lot of time because every time I would run the code it took a few minutes to complete. After many attempts and still just getting around 98% test accuracy with a large jump after the first epoch, I was looking for other values I could modify.

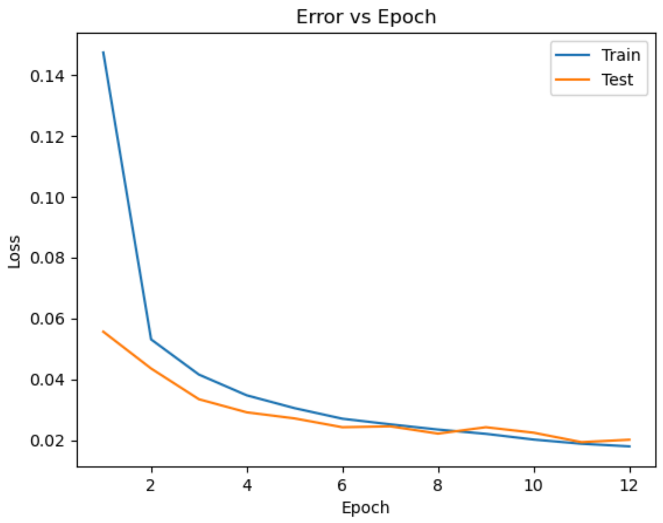
3. The Biggest Change

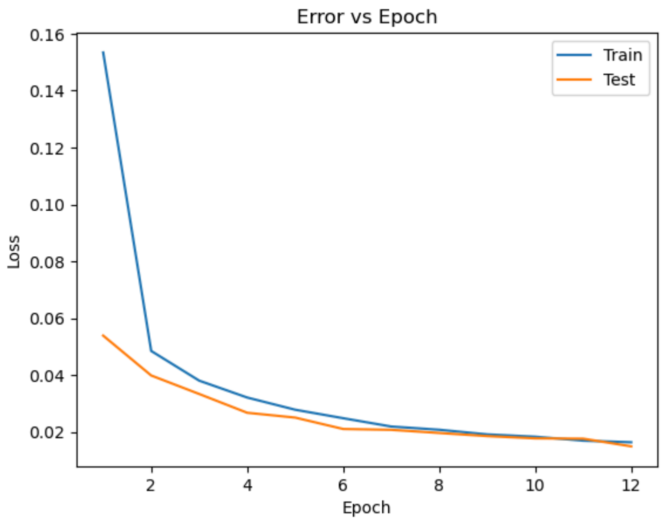
The modification where I noticed the largest change and was able to get the test accuracy to about 99% was with the number of neurons in the fully connected layers. I started increasing them little by little and saw a better accuracy in the first epoch. However there was still the initial jump between the first and second epoch. I started by changing the first fully connected layer to 200. Then I started to try larger increases in the neurons.

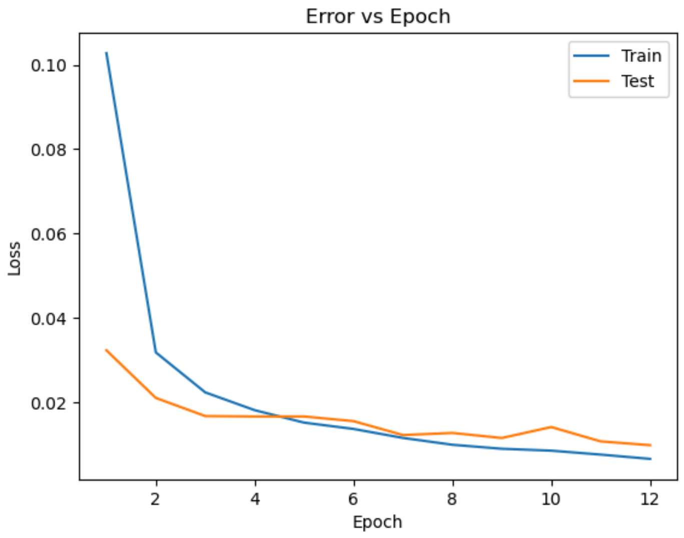
4. Analysis

For the baseline code and with the change of the kernel size to 5x5, the accuracy was about 98%. I believe this is because it was just a small change by increasing the kernel size, but it was still trying to extract the same number of features. The first and second convolutional layers had the same number of filters for these two tests. I believed that for the first two scenarios it plateaued around 98% because it was only able to store a max amount of features. With the test cases where the neurons were increased, the accuracy was able to increase. This is because as more features were extracted they were able to get stored in more neurons. However, 98% and 99% are great values for accuracy and there is not very much room to improve.

5. Plots







6. Conclusion

After altering the parameters for the CNN model's parameters for the MNIST dataset, we can see that with 2 convolutional layers we can get an accuracy of over 98% after 12 epochs. The more neurons the cnn has in the fully connected layers have the largest impact on the accuracy. Because the images were 28 by 28, I did not want to increase the kernel size too much, because then it would cover more of the image. I found that using the CNN model is a strong model to use for prediction, but it is also costly.