

Technical Report

The first layer is created from the pixels in the 28×28 pixel image which creates 784 nodes. Looking at the inside of the network there are two inner layers. The first one is activated with a linear function then proceeded by ReLU function followed by another linear layer composed of 10 nodes. These 10 nodes represent the 10 final options for what the number could be. The parameters are:

Batch size = 32

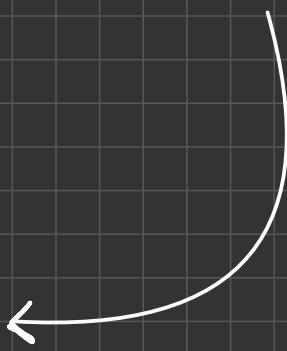
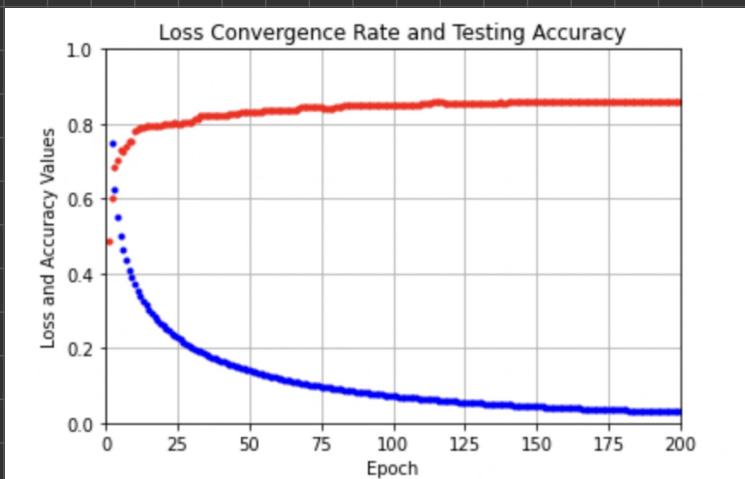
Epochs = 200

Learning rate = .01

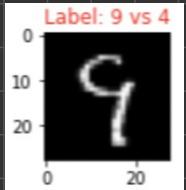
The training is done by using forward propagation to find outputs for each input at nodes then going to backward propagation. This finds gradients of all the layers. The loss values are calculated with MSE through the prediction of the outputs compared to true values.

In the first part of the assignment (appetizer) I got a test accuracy of .42 and a training accuracy of .721. Both are low especially compared to the neural network with a test accuracy of .86.

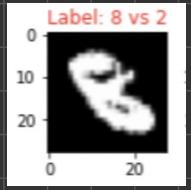
Graphing the loss values and testing values for the 200 epochs is down below.



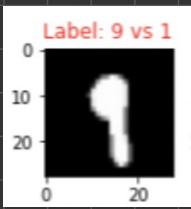
Below I give examples of errors and make predictions on why my network got them wrong.



This most likely was predicted wrong because there is a gap in the top of the nine. This likely caused for the prediction of the four.



This most likely was predicted wrong because the eight is super messy and almost looks like a blob. This causes for the system to be confused and have a poor guess.



This most likely was predicted wrong because the nine is filled in and not open like it should be. This causes the nine to be predicted as a one given it is one solid block.