Daniel Katz
2/8/20
Advanced Topics in Computer Science
Mr. Bleckel

Digit Recognition Report

Initial Parameters				
Number Epochs	Min Initial Weight	Max Initial Weight	Learning Rate	Testing Set Accuracy
10	-1	1	.1	66.778%

While not awful, the testing set accuracy leaves a lot to be desired with a goal accuracy near 90%. My hypothesis of why the accuracy is low is because the initial weight limits allow for a wide range of possible random weights, which might cause there to be a non-even distribution of weights. For example, most of the weights could be close to -1, while I presume a more even initial distribution is desirable to attain a higher accuracy. Another factor could have been overtraining the model. The testing set accuracy is lower by 6 points than the 10th training epoch. After the 5th epoch there is only accuracy increases by about 2 or 3 points which also points to the fact that much more training past that point is ineffective for training the model to detect previously unseen data.

Tuned Parameters				
Number Epochs	Min Initial Weight	Max Initial Weight	Learning Rate	Testing Set Accuracy
7	-.4	.4	.1	91.263%

The accuracy after tuning the parameters is much closer to expectations. I lowered the number of epochs because during testing I noticed that after the 7th epoch the accuracy changes were mostly under 1 point, sometimes even decreasing, and often lowered the testing set accuracy. This is because the model was becoming overtrained to the training data instead of being trained to recognize digits in general. I tested with 5 and 6 epochs, but adding 1 or 2 epochs still resulted in an accuracy boost of about 5 points per epoch.

As for the weights, the reason I made the initial edge weight range smaller was to ensure that even if there was an uneven distribution of weights, there would not be a wide range between all the weights. When I tested a initial weight range smaller than -.4 to .4, I found that

the final accuracy went down. I believe this phenomenon is caused by backpropagation not being effective enough with the small learning rate and small initial range.

I kept the learning rate the same because it consistently gave results in the best accuracy. Even small changes made the testing set accuracy decrease. For example, a learning rate of .2 resulted in a testing accuracy of around 80%. Changing the learning rate and other parameters as well resulted in a mid 80% testing accuracy, but the above settings have proven most fruitful for a high testing set accuracy.

The drawing feature of my lab does not have as high a accuracy score. Qualitatively, I would assume around a 50% correct guess rate if numbers are centered and drawn as big as possible. The numbers that are identified correctly most of the time are 0, 2, 5, and 7. On the other hand, 3 and 8 are sometimes correctly identified if care is taken into drawing them and 1, 4, 6, and 9 are usually not correctly identified. I assume this has to do with the distinctiveness of the shape and if it uses the same pixels as other shapes. An example is how 1, 4, and 9 all have a straight line closer to the right. I experimented with the pen width and found that a thicker width of 30 pixels worked best.