Dylan Wulf

CSC380: Artificial Intelligence

Project 3: Handwritten Digit Recognition

Stopping Criterion

To decide on stopping criterion for the Neural Network, I ran the program with a varying number of epochs and varying set size. Increasing both of these numbers did increase the accuracy. However, I found that after the perceptrons had approximately 30,000 to 40,000 training sessions (set size \* epochs), increasing the number of epochs seemed to have little effect on the accuracy. So when gathering the test results, I set the number of epochs so that the perceptrons would get approximately 30,000 to 40,000 training sessions.

For the Nearest Neighbor program, there are less variables so finding stopping criterion is a bit easier. Nearest Neighbor’s accuracy peaks at approximately 96% using 10,000 training images. The accuracy does not decrease very much unless you go below 7,000 images, so this is probably a good stopping point.

Learning Rate

To decide on the learning rate, I ran the Neural Network several times while varying the learning rate. The accuracy seemed to be the best between 0.03 and 0.3, so I ran the program with learning rates 0.03, 0.1, and 0.3 while varying the set size. The results of this can be seen on the graph below.

With LR=0.03, the accuracy started off higher than the other two. However, LR=0.03 had a slightly lower accuracy overall.   
With LR=0.1, it started off in the middle and varied significantly in accuracy. It peaked at 93% with 4000 training images.  
With LR=0.3, it started off the lowest, and varied significantly.

Overall, these three learning rates appear to be very similar. More testing is needed to determine what is best. It seems like lower learning rates perform slightly better with a smaller set size, while larger learning rates perform slightly better with a larger test size. However, this could just be coincidence.

I also attempted to use multithreading for the neural network training. I found out that the final result is different depending on the order that the training data is given, so multithreading introduced a slightly random element into the program. When running the program with the same exact configurations using multithreading, the accuracy varied by about 1.5%. However, training with multithreading still wound up with fairly accurate results overall.

This graph describes the learning curve of the Nearest Neighbor program. It does not appear to be nearly as sharp as the Neural Network graph, but it does result in noticeably better accuracy (96.6%) at 10,000 training images.