Marshall Lanning

CECS 625 Parallel Programming Extra-cedit Project

October 29, 2019 (50 points)

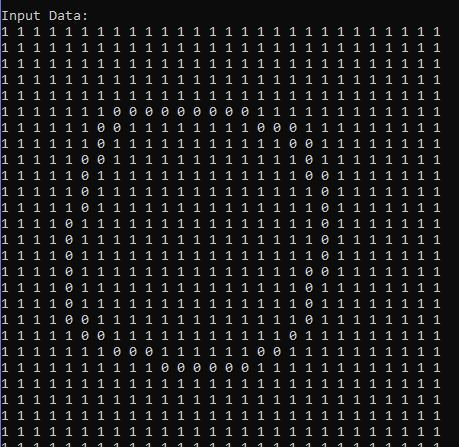
Due: November 7 (Wed) Demo your extra-credit project in class.

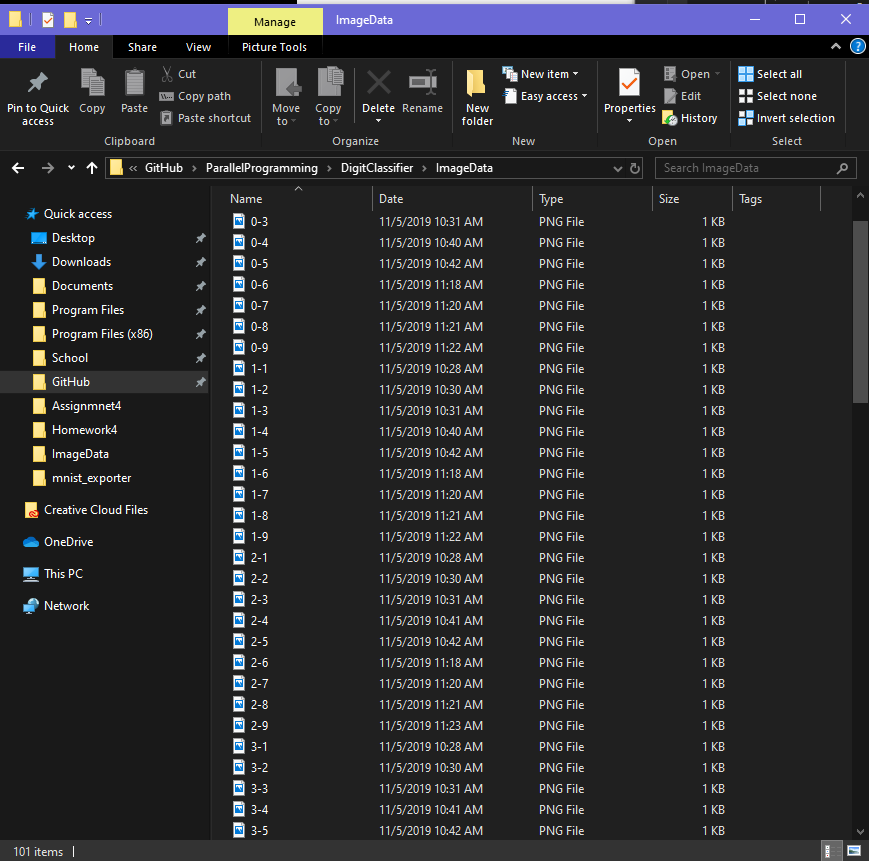
Based on the materials you learned from Section 6.3 and 6.5, write a C++ project to implement a written digit classifier as described below:

Input: An 28x28 float vector (i.e. an 28X28 grey-level image representin a written digit from 0-9)

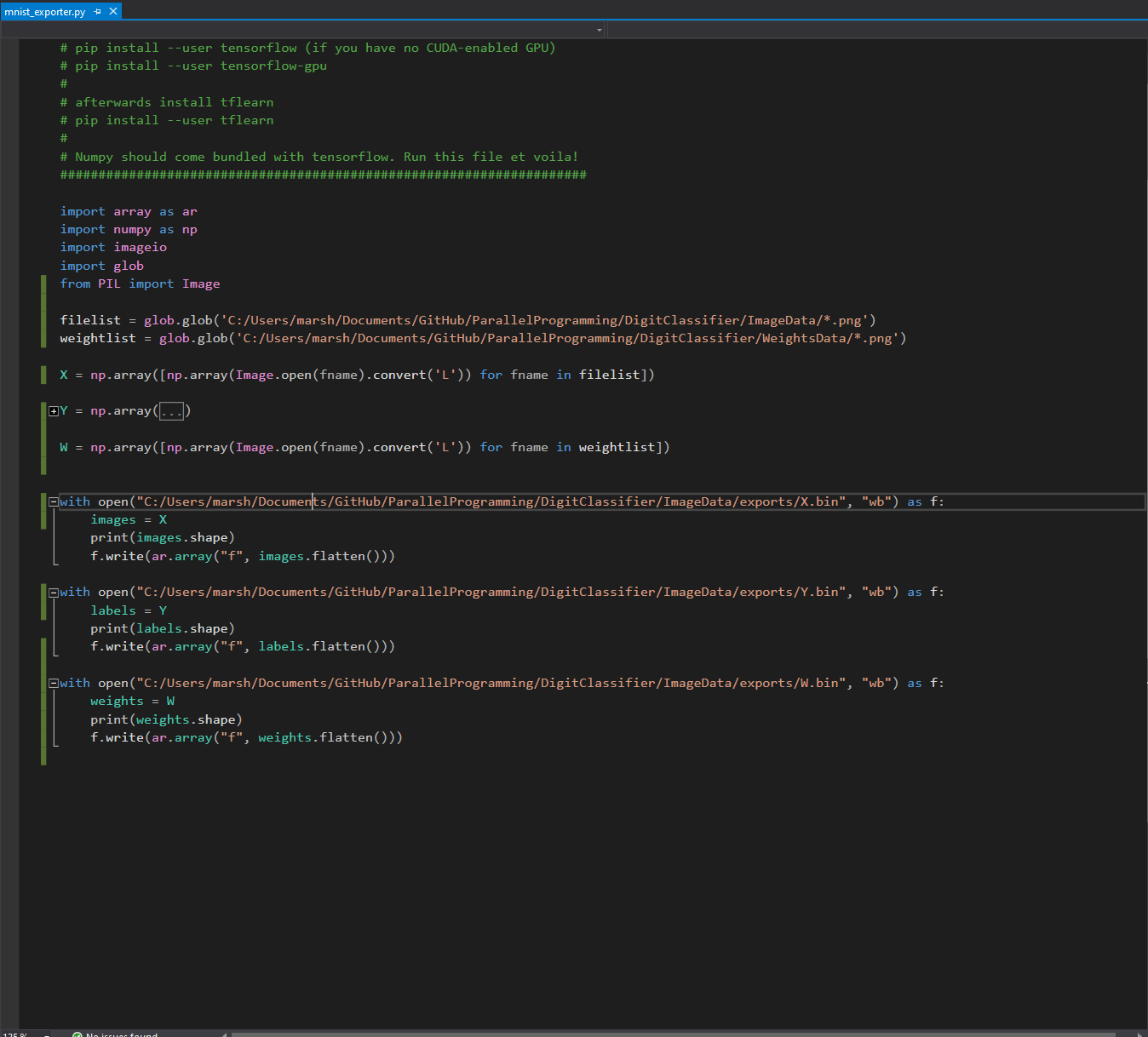
I used the Windows Paint application to write out 10 of each number in a 28X28 pixel fram so 100 numbers total. An example of 0 is below:





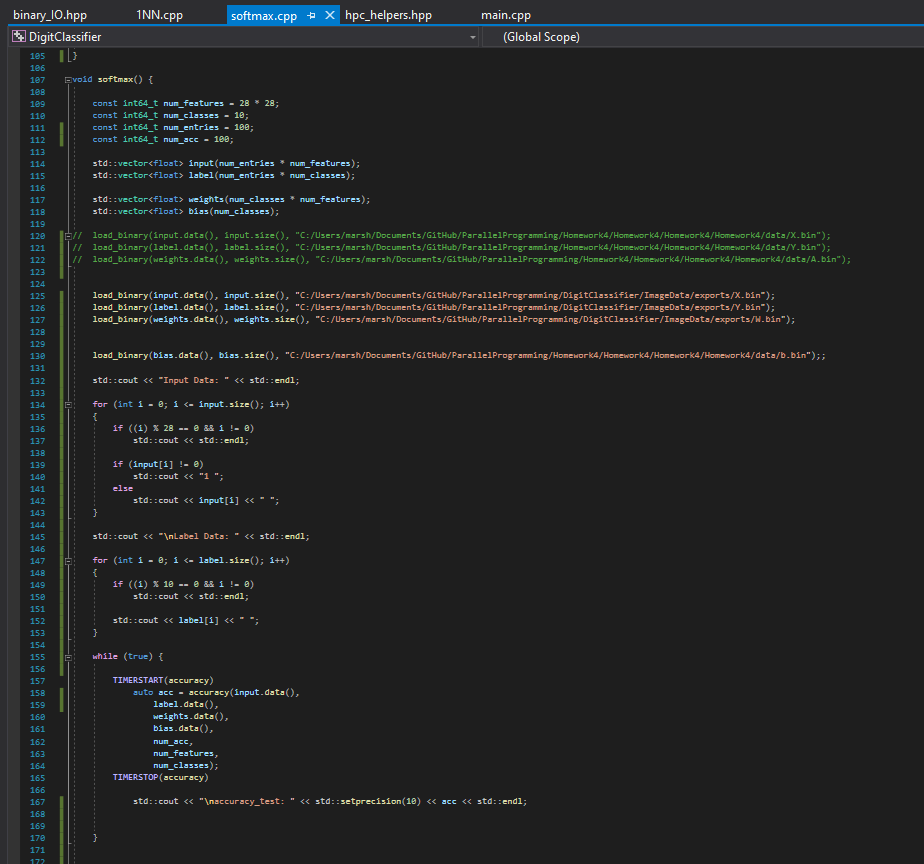


I used python like the mnist\_exporter to export my data set correctly and make my own weights to compare against.

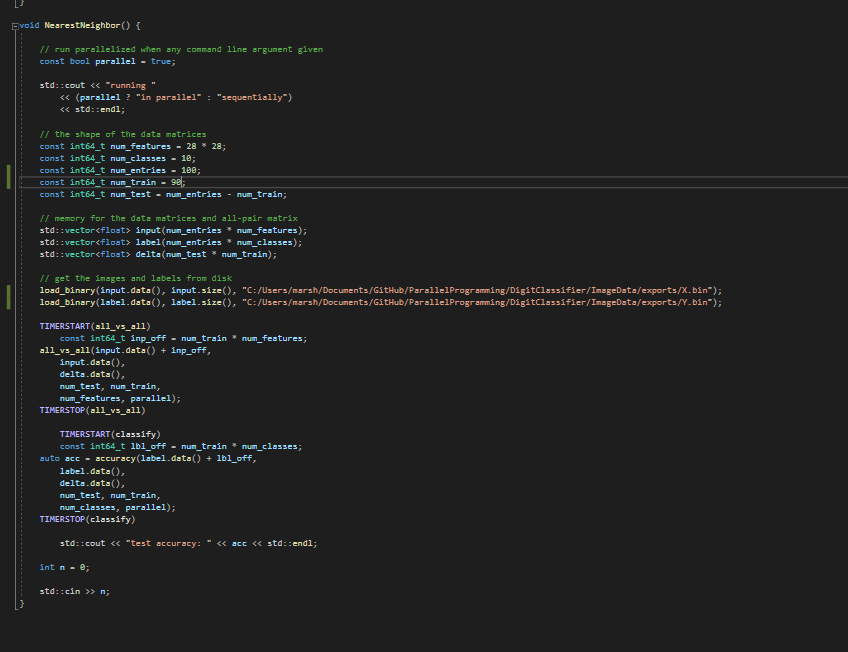


Output: The digit in the image (i.e. 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9)

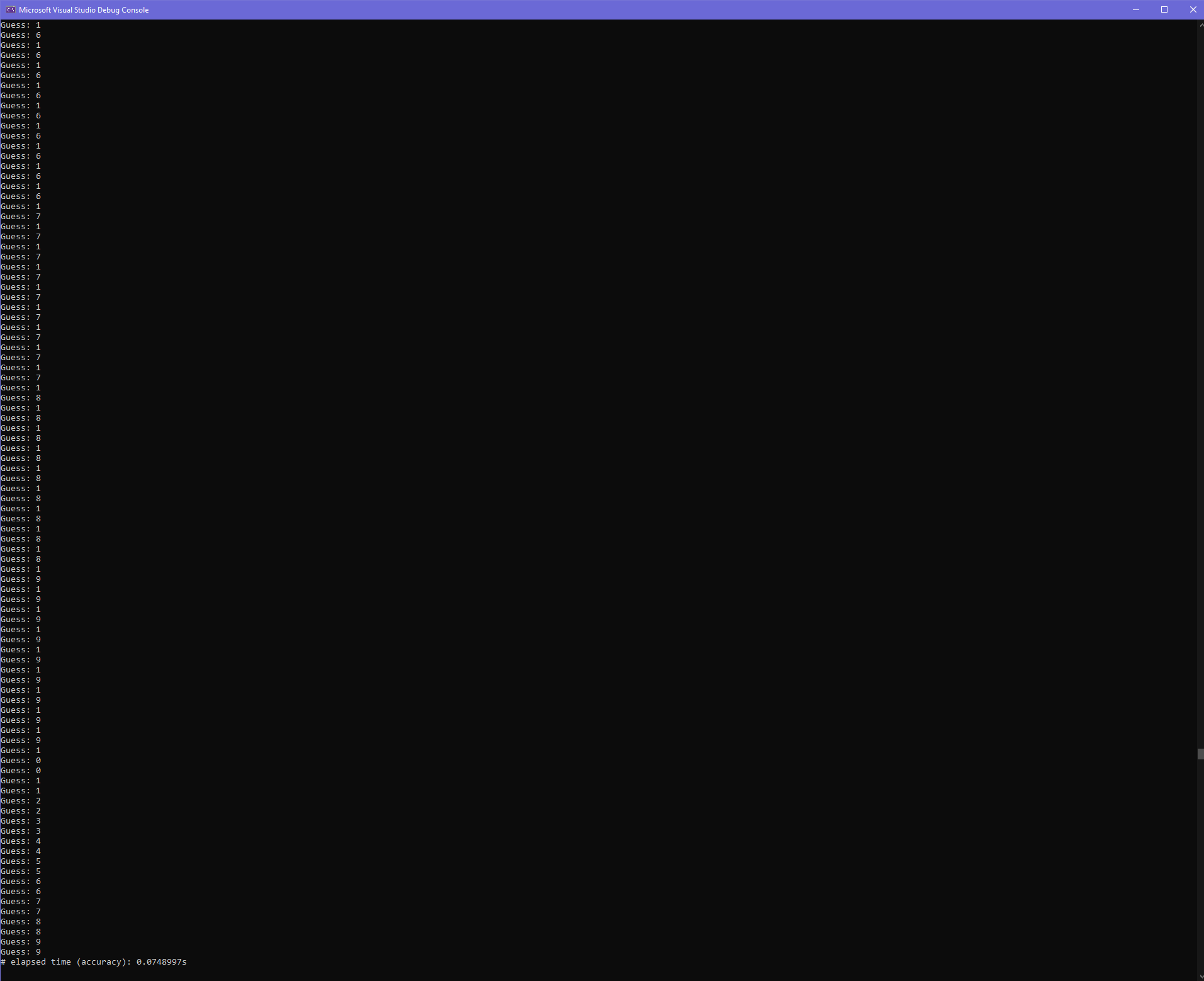
Softmax implementation:



Nearest Neighbor Implementation:

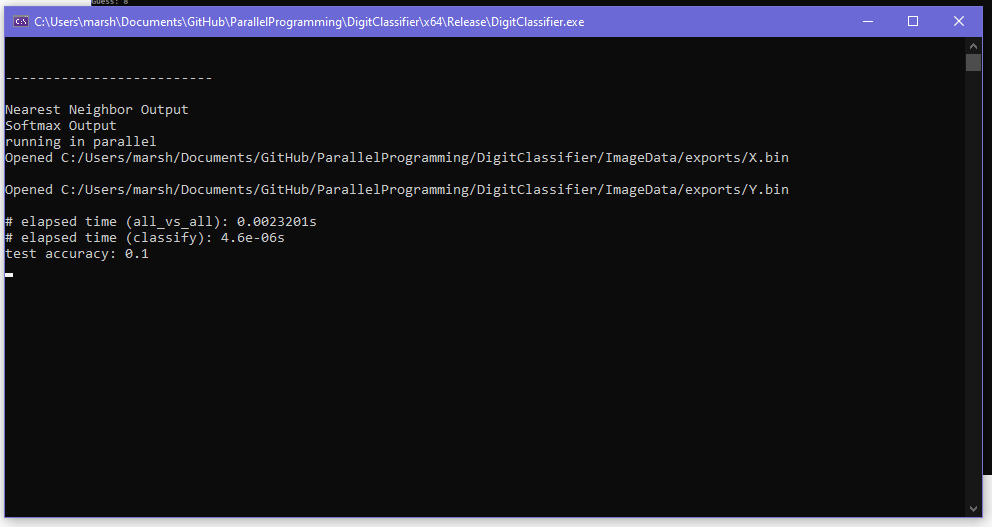


Output of Softmax guessing an iteration of my 100 numbers. The way the numbers are guessed seem somewhat accurate, the order that they are implemented in is 9 0’s, 9 1’s, 9 2’s, etc so you see the guesses increasing in value as it moves through the dataset.

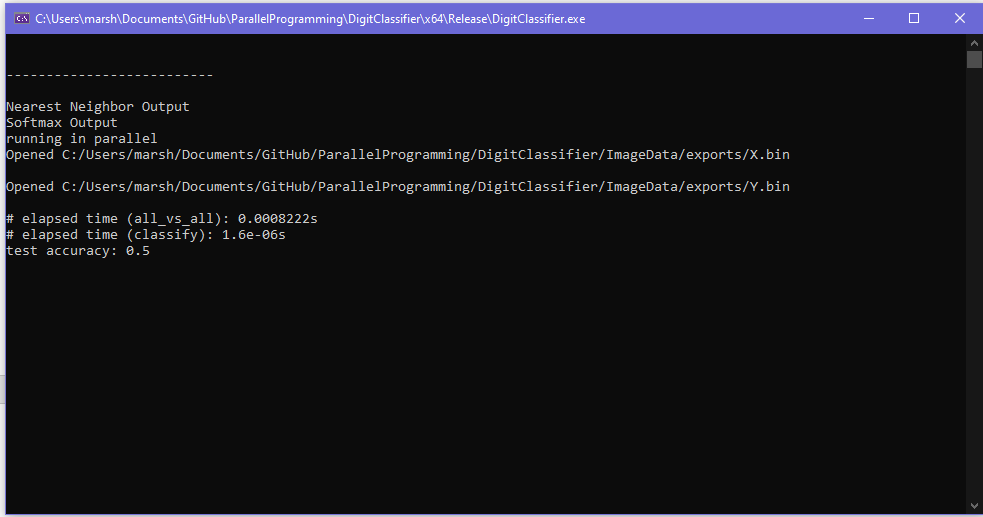


1NN Implementation:

Half of the values are used for training:



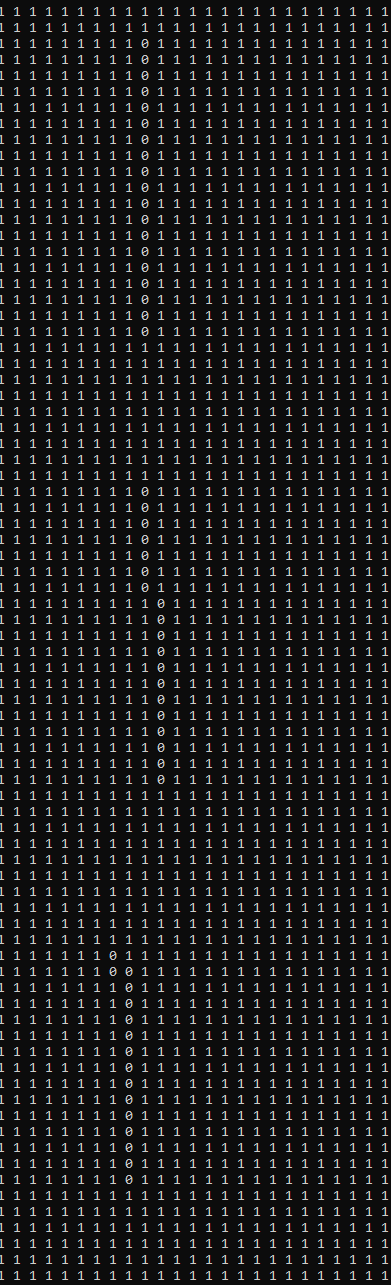
90 of the values are used for training:



Better accuracy when you train more!

I believe my Softmax and 1NN implementations are not very accurate. This is because of my dataset. It is not consistent and a lot of room for ambiguity because the utility used with paint draws a line that covers a 1X1 pixel dimension. In other words, the written digits take up a very small amount of the solution space.

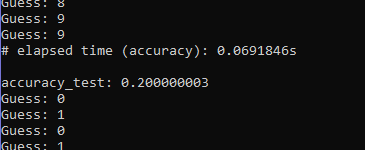
I also was not very careful about making sure they all looked a like and were in the same area. For example, look at my 1’s:



Compare the accuracy and computing times of these two classifier algorithms.

My Softmax implementation can be better by improving the dataset. Espically the weights data set which is just example images of 0-9. If I made them more consistent and possibly even had more data to use I would see more accuracy. The computing times are really not much different and my dataset is not big enough to have significance. In the original dataset for Homework 4 the timing of Softmax was better than 1NN, Softmax can classify these images in about half the time.

Softmax:



1NN:

