Homework 0

due Monday, February 26

Please submit all assignments as zip files titled [lastname]_assignment1.zip. Please include Python file (not a Jupyter notebook) with all code, as well as Word or PDF file with all results and discussion.

For this assignment, you will be implementing the k-nearest neighbors algorithm to perform digit OCR on the attached MNIST data set. Two CSV files have been provided, one with the training data, one with the test data. Each line of the csv file has 785 numbers. The first number is the label (i.e. what digit the image shows). The remaining are the greyscale values of the 784 pixels of the image.

Two functions have been provided for you in assignment1.py. The first reads all the data in to numpy format, and splits the features from the labels. You will need both the pandas and numpy packages to run this function. This function has 2 mandatory arguments (the file names for the two CSVs) and two optional arguments that allow you to specify the size of your training set and the size of your test set (as you will likely want to test your code on a small subset of the data before running it on the whole data set). The second function simply computes the Euclidean distance between two vectors using numpy. (I highly recommend you use this function, as it can do distance computations faster than other approaches.)

Implement the kNN algorithm, as specified in the lecture 1 slides, by looping over your test and training sets. You may use numpy as you like to manage your data and compute distances, but you may not use any packages that implement the kNN algorithm. Test your algorithm for four values of k: 1, 3, 5, and 7. While you will get more accurate results by training on more data, due to the slow nature of the algorithm, you may wish to run it on a subset of the data. You must use at least **6000** training points and at least **1000** test points. Additionally, you should use the same training and test sizes for all values of k and note these sizes in your write-up.

Compute test error rates for each of these four values of k, and present these results in your write-up. Additionally, for the best value of k (i.e. the one that resulted in the lowest test error rate), show the confusion matrix. (You may wish to use scikit-learn to compute errors or confusion matrices.) Briefly discuss which digits the algorithm most commonly confused and why.

Finally, propose one change you could make to improve the performance of this algorithm on this task. This improvement may be to either the data pre-processing or the algorithm itself. You do not need to implement this improvement (and you may suggest an improvement that you currently do not have the technical skills to implement), but it should be an improvement that could feasibly be implemented on this data, and you should provide enough details that someone could implement your improvement. (For example, if you suggest using a different distance metric, you do not need to provide the mathematical details of how to calculate distance, but you should present a clear description of what it captures.)