COGSCI 131 – Assignment 6
DUE: March 19 at class start

We are going to implement the perceptron algorithm on a very simple feature spaces: pixels in images of handwritten digits. To do this, we have provided a function to read image files (load_image_files) of a given digit (n), from a provided path (path=…), convert them to a numpy matrix, and finally flatten that numpy matrix into a vector. The function load_image_files returns all of the vectors corresponding to pixels in handwritten digits of n from the MNIST dataset. Note that the original images were 28x28 pixels, this function returns "images" of 28x28=784 zeroes and ones, corresponding to whether the image was white or black at a given location. But, convenient for the assignment, each "image" is a single vector of length 784.

1. [20pts, HELP] Write an implementation of the perceptron learning algorithm that first loads images for the digit "0" and then for the digit "1". Start with random weights from a normal distribution. Compute the average accuracy on blocks of 25 items and plot this accuracy until you think it won't get better.

2. [5pts, SOLO] Does your solution in Q1 converge on 100% accuracy or not? What does this mean in terms of the linear separability of "0" and "1" on this feature space?

3. [15pts, HELP] Reshape (numpy.reshape) your weight vector after training so that it is a 28x28 matrix. This corresponds to the weight assigned to each pixel in the image. Show a picture of this weight matrix and interpret it in a sentence or two. What do large negative and large positive values mean, intuitively? What do numbers near zero mean? Why does this matrix look the way that it does, in terms of where large positive and negative terms are located?

4. [10pts, SOLO] What should you expect to happen if you set the elements of the weight vector which are close to zero to be actually zero? Do this for the 10, 20, 30, … 780 weight values closest to zero (in absolute value) and plot the resulting accuracies on 1000 random classifications of "0" vs "1". What does this tell you about the proportion of the image which is diagnostic about "0" vs "1"?

5. [20pts, SOLO] Next show a matrix of the classification accuracy of each pair of digits after enough training. Make this a plot (with colors for accuracy rather than numbers). Does it match your intuitions about which pairs should be easy vs. hard? Why or why not?