Programming Assignment

Submission guidelines:

- Download the file `skeleton_sgd.py` from Moodle. In each of the following questions you should only implement the algorithm at each of the skeleton files. Plots, tables and any other artifact should be submitted with the theoretical section.

- In the file `skeleton_sgd.py` there is an helper function. The function reads the examples labelled 0, 8 and returns them with 0-1 labels. Case you are unable to read the MNIST data with the provided script, you can download the file from here:

  `https://github.com/amplab/datasciencesp14/blob/master/lab7/mldata/mnist-original.mat.`

- Your code should be written in Python 3.

- Make sure to comment out or remove any code which halts code execution, such as matplotlib popup windows.

- Your code submission should include one file: `sgd.py`.

1. **(25 points) SGD for Hinge loss.** We will continue working with the MNIST data set. The file template (`skeleton_sgd.py`), contains the code to load the training, validation and test sets for the digits 0 and 8 from the MNIST data. In this exercise we will optimize the Hinge loss (as you seen in the lecture) using the stochastic gradient descent implementation discussed in class. Namely, at each iteration $t = 1, \dots$ we sample i uniformly; and if $y_i w_t \cdot x_i < 1$, we update:

$$w_{t+1} = (1 - \eta_t) w_t + \eta_t C y_i x_i$$

   and $w_{t+1} = (1 - \eta_t) w_t$ otherwise, where $\eta_t = \eta_0 / t$, and η_0 is a constant. Implement an SGD function that accepts the samples and their labels, C, η_0 and T, and runs T gradient updates as specified above. In the questions that follow, make sure your graphs are meaningful. Consider using `set_xlim` or `set_ylim` to concentrate only on a relevant range of values.

   (a) **(10 points)** Train the classifier on the training set. Use cross-validation on the validation set to find the best η_0, assuming $T = 1000$ and $C = 1$. For each possible η_0 (for example, you can search on the log scale $\eta_0 = 10^{-5}, 10^{-4}, \dots, 10^4, 10^5$ and increase resolution if needed), assess the performance of η_0 by averaging the accuracy on the validation set across 10 runs. Plot the average accuracy on the validation set, as a function of η_0.

   (b) **(5 points)** Now, cross-validate on the validation set to find the best C given the best η_0 you found above. For each possible C (again, you can search on the log scale as in section (a)), average the accuracy on the validation set across 10 runs. Plot the average accuracy on the validation set, as a function of C.

   (c) **(5 points)** Using the best C, η_0 you found, train the classifier, but for $T = 20000$. Show the resulting w as an image.

   (d) **(5 points)** What is the accuracy of the best classifier on the test set?

2. **(20 points) SGD for multi-class cross-entropy.** The skeleton file contains a second helper function to load the training, validation and test sets for all the digits. In this exercise

we will optimize the multi-class cross entropy loss using SGD. Recall the multi-class cross-entropy loss discussed in the recitation (our classes are $0, 1, \ldots, 9$):

$$\ell_{CE}(w_0, \ldots, w_9, x, y) = \log\left(\sum_{i=0}^{9} e^{w_i \cdot x}\right) - w_y \cdot x$$

Derive the gradient update for this case, and implement the appropriate SGD function.

(a) **(10 points)** Train the classifier on the training set. Use cross-validation on the validation set to find the best η_0, assuming $T = 1000$. For each possible η_0 (for example, you can search on the log scale $\eta_0 = 10^{-5}, 10^{-4}, \ldots, 10^4, 10^5$ and increase resolution if needed), assess the performance of η_0 by averaging the accuracy on the validation set across 10 runs. Plot the average accuracy on the validation set, as a function of η_0.

(b) **(5 points)** Using the best η_0 you found, train the classifier, but for $T = 20000$. Show the resulting $w_0, \ldots, w_9$ as images.

(c) **(5 points)** What is the accuracy of the best classifier on the test set?