EE 559 **Homework Week 5** Posted: Fri., 2/13/2020
Jenkins Due: Fri., 2/20/2020, 2:00 PM

1. Code up a 2-class perceptron classifier, for the case of 2 features (3 dimensions in augmented space). As in the previous homework problems, you are expected to code this up yourself rather than use a library or toolbox, so the same guidelines apply.

   Use basic sequential gradient descent.

   Hint: One way to randomly shuffle the data points, is to generate random permutations of numbers from 1 to *N*, where *N* = number of training samples. The result can be used as indices for the data points and their labels.

   It is recommended (but not required) that you use augmented space, and reflect the training data. Do not reflect the test data.

   Because the data you will use might not be linearly separable, you will need two halting conditions:

   (1) The usual perceptron halting condition: if there have been no updates to $\underline{w}$ in 1 epoch, then halt;

   (2) An ad hoc one in case condition (1) is not satisfied. A reasonable one is to set a maximum number of iterations or epochs (1000 epochs should be high enough), and during the last full epoch, at each iteration *i*, evaluate $J\left(\underline{w}(i)\right)$ over all training data points, and store each $J\left(\underline{w}(i)\right)$. Then, after the last iteration, choose the $\underline{w}(i)$ that had the smallest criterion value $J\left(\underline{w}(i)\right)$.

   For your initial weight vector, use $\underline{w}(0) = 0.1(\underline{1})$ (vector of 0.1 in each component).

   For your learning rate parameter, use $\eta(i) = 1$.

Run your classifier on 3 datasets: Synthetic1 (get from HW1 folder), Synthetic2 (also from HW1 folder), and Synthetic3 (from the current HW folder).

(a) For each dataset, report on the final weight vector, the classification error rate on the training set, and the classification error rate on the test set.

(b) Also for each dataset, give a 2D plot showing decision boundary and regions, as well as training data points (similar to plots of HW1). You may find it convenient to use the PlotDecBoundaries function from HW1 folder.

(c) For Synthetic1 and Synthetic2 datasets, compare your error rates of part (a) with the error rates obtained in HW1(a); you can compare with the posted solutions if your HW1 answer wasn't correct. Explain any significant differences between perceptron result and nearest means result.

Problem 2 on next page…

2. In a gradient descent procedure, let the "weight update" be defined as

$$\Delta \underline{w}(i) = \underline{w}(i+1) - \underline{w}(i).$$

Consider a given weight vector $\underline{w}_0$ which is the weight vector at (given) iteration i, so that $\underline{w}(i) = \underline{w}_0$.

Let the criterion function take the form $J(\underline{w}) = \sum_{n=1}^{N} J_n(\underline{w})$, in which $J_n(\underline{w})$ is the criterion function for one data point. In this problem, do not plug in for the perceptron criterion function; please solve the problem for the general case of J and J_n. Assume $\eta(i) = \eta = \text{constant}$.

(a) For stochastic gradient descent, variant 2, find $\mathrm{E}\{\Delta \underline{w}(i)\}$, in which $\mathrm{E}\{\ \}$ denotes expected value, and for each probabilistic experiment, n is randomly chosen from $\{1, 2, \cdots, N\}$ with uniform probability. Try to express your answer in terms of $\nabla_{\underline{w}} J(\underline{w}_0)$.

(b) Compare your result of (a) with $\Delta \underline{w}(i)$ for batch gradient descent. Describe in words how the batch GD update compares with the expected value of the stochastic GD variant 2 update.
Explain why the comparison between these two expressions makes sense, or does not make sense.