> **Please note:** new guidelines on how your homework may be submitted. Submit 2 pdf files: one of your solution (with "solution" or "soln" in its filename), and one of all your code, computer readable (with "code" in its filename). No other formats; and no compressed files (such as .zip or .rar files). Thank you!

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 $\underline{w}(0) = 0.1\left(\underline{1}\right)$ (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 HW2 folder), Synthetic2 (also from HW2 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 HW2). You may find it convenient to use the PlotDecBoundaries function from HW2 folder.

(c) For Synthetic1 and Synthetic2 datasets, compare your error rates of part (a) with the error rates obtained in HW2(a); you can compare with the posted solutions if your HW2 answer wasn't correct. Explain any significant differences between perceptron result and MDTCM result.

2. For 2-class perceptron *with margin* algorithm, using basic sequential GD, fixed increment, prove convergence for linearly separable training data by modifying the perceptron convergence proof covered in class. You may write out the proof, or you may take the 3-page proof from lecture (included in this HW folder, as updated with corrections), and mark up the proof to show all changes as needed. If you mark up the existing proof, be sure to mark everything that needs changing (*e.g.*, if a change propagates through the proof, be sure to make all changes for a complete answer).

3. **Extra credit.** [Comment: you will probably find that this problem is not difficult; it is extra credit because the regular-credit problems above are already sufficient for one homework assignment.]

In a gradient descent procedure, let $\Delta \underline{w}(i) = \underline{w}(i+1) - \underline{w}(i)$. Also, 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, there is no need to plug in for the perceptron criterion function; please solve the problem instead for the general case of J and J_n. Assume $\eta(i) = \eta = \text{constant}$.

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

   Hint: the answer is a simple linear function of $\nabla_{\underline{w}} J_n(\underline{w})$.

(b) Also for stochastic gradient descent, variant 2, find $E\left\{\sum_{i=0}^{N-1} \Delta \underline{w}(i)\right\}$, in which each $\Delta \underline{w}(i)$ is independent and identically distributed, and has the same distribution as in (a).

(c) Compare your result of (b) with $\Delta \underline{w}(i)$ for batch gradient descent. Describe in words what this means in terms of how batch gradient descent compares with stochastic gradient descent variant 2.