CSE4214 Pattern Recognition Lab

Experiment No 2

"Implementing the Perceptron algorithm for finding the weights of a Linear Discriminant function."

Problem Description:

1. Take input from "train.txt" file. Plot all sample points from both classes, but samples from the same class should have the same color and marker. Observe if these two classes can be separated with a linear boundary.
2. Consider the case of a second order polynomial discriminant function. Generate the high dimensional sample points y, as discussed in the class. We shall use the following formula:

   $$y = [x_1^2 \quad x_2^2 \quad x_1 * x_2 \quad x_1 \quad x_2 \quad 1]$$

   Also, normalize any one of the two classes.
3. Use Perceptron Algorithm (both one at a time and many at a time) for finding the weight-coefficients of the discriminant function (i.e., values of w) boundary for your linear classifier in task 2.

   Here α is the learning rate and $0 < \alpha \leq 1$.

   $$\underline{\mathbf{w}}(\mathbf{i+1}) = \underline{\mathbf{w}}(\mathbf{i}) + \alpha\, \underline{\tilde{y}}_m^{(k)} \qquad \textbf{if } \underline{\mathbf{w}}^T(\mathbf{i})\ \underline{\tilde{y}}_m^{(k)} \leq 0$$

   $$\textbf{(i.e., if } \underline{\tilde{y}}_m^{(k)} \textbf{ is misclassified)}$$

   $$= \underline{\mathbf{w}}(\mathbf{i}) \qquad \textbf{if } \underline{\mathbf{w}}^T(\mathbf{i})\ \underline{\tilde{y}}_m^{(k)} > 0$$
4. Three initial weights have to be used (all one, all zero, randomly initialized with seed fixed). For all of these three cases vary the learning rate between 0.1 and 1 with step size 0.1. Create a table which should contain your learning rate, number of iterations for one at a time and batch Perceptron for all of the three initial weights. You also have to create a bar chart visualizing your table data.

Also, in your report, address these following questions:

a. In task 2, why do we need to take the sample points to a high dimension?

b. In each of the three initial weight cases and for each learning rate, how many updates does the algorithm take before converging?

Marks Distribution:

Task 1	2
Task 2	2
Task 3	3
Task 4	3

Sample Output (<u>Initial Weight Vector All One</u>):

Alpha (Learning Rate)	One at a Time	Many at a Time
0.1	6	102
0.2	92	104
0.3	104	91
0.4	106	116