1. Introduction to the problem and a brief summary of your solution approach.

Pocket theory is the modified version of Perceptron learning algorithm. It solves the stability problem of PLA. PLA is only applicable for separable dataset where as Pocket algorithm also applicable for separable and non-separable dataset where the aim is to find a perceptron with small number of misclassifications.

Solution Approach: Pocket algorithm keeps the best result seen so far in its pocket (that is why it is called Pocket Learning Algorithm). The best result means the number of misclassifications is minimum. If the new weights produce a smaller number of misclassification than the weights in the pocket, then replace the weights in the pocket to the new weights; if the new weights are not better than the one in the pocket, keep the one in the pocket and discard the new weights. At the end of the training iteration, the algorithm returns the solution in the pocket, rather than the last solution.

Solution steps:

1. Dataset preparation
2. Initialize weight vector, w
3. Update w by running PLA
4. Calculation misclassification errors based one weight vector
5. Pick random dataset points for verifying classifications and update w if necessary

2. What was the average error in and out of sample for each N? (Test over 1000 runs of your algorithm)

---- Training Results N=50 ----------
Average In error: 3
Average Out error: 0
Average iteration: 2
---- Training Results N=200 ----------
Average In error: 6
Average Out error: 94
Average iteration: 5

3. Example graphs of your algorithm's results with the in-sample data.

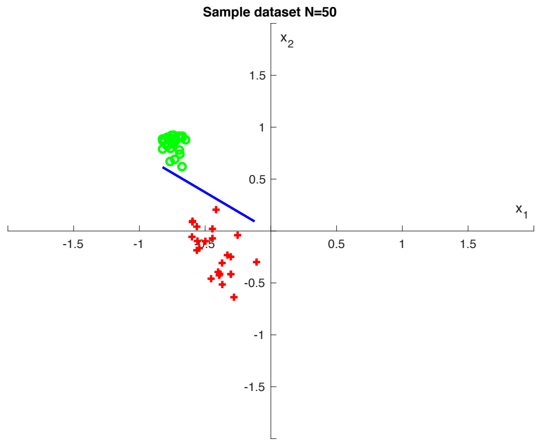

Fig 1: Sample result for N=50

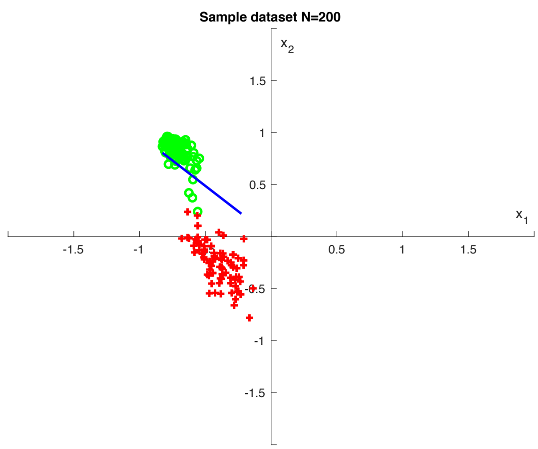

Fig 2: Sample result for N=200

4. Explanation of your choice of stopping criteria (i.e. fixed number of iterations, etc.).

   Pocket algorithm deals with linearly non-separable data. In worst case, the algorithm can fall into infinite loop. Therefore, we need a stopping criterion. In my case, I have used higher number of iterations for higher sample size i.e. N=50 and # iteration 100, N=200 & iteration 400. Also, iteration has been set using error rate.

5. What difference or changes were necessary when the training set was set to N=50 and N=200?

   Algorithm convergence time (iterations) has increased with sample size. The main difference we have observed in iterations number. For N=50, we have used #iteration 100 which is acceptable with error rate. When the sample size becomes four time larger (N=200), it is needed more iterations to convergence.