Math 354: HW #4
Final Project Preparation (prework) Machine Learning
Due Monday 12/11/2017
Instructor: Dr. Fred Park

The goal for machine learning, in this context is, for the computer to be able to learn from data without being programmed. An example would be for the computer to learn a model for pass/no-pass based on exam scores and labeled data. By labeling, we mean that a value of 1 means pass and 0 no-pass. Thus, if two midterm scores were input that were not part of the training set, the computer would be able to tell you if they passed or not without ever knowing the specific grading criteria. (In fact, it can also learn the grading criteria if needed.) An example: if scores were 40 and 80 for midterms 1 and 2 respectively, the computer would tell you that the student did not pass. However, if the scores were 40 and 90, it would tell you that they did in fact pass. Thus, the computer learned a decision boundary as to what scores would allow a pass or not. Based on this, we can make predictions on pass/no-pass of future students.

1. Decide if you want to take on the final project as solo work or group work. If group work, you can have a group of 2-3 students total. No groups larger than that will be allowed. Once you have a general idea of the group, let the instructor know and the instructor will determine if the group is appropriate or not. You will need instructor approval for your group before you begin.

2. Find a real world data set that interests you that has 2 or more features and 2 class labels. For example, the exam scores data set we saw in class has 2 features: exam 1 and exam 2. It has 2 labels: passed and not-passed. You can collect data from any reputable source. Best thing to do is to find something that interests you and you will have fun collecting the data. The data set needs to have at least 30 examples that we will use for both training and testing. You can save the data as a .txt file or as a .mat file. The latter if you use matlab to enter the scores into a matrix. If you feel that the data must have more than 2 labels, this can also be done via Logistic Regression or Neural Networks as well.

3. (a) If you have 2 features, you can use the scatter plot from class from the template code found on the course website (`classify_train_test_final.m`). If you have more than 2 features, you can still visualize the data via a 3-D scatter-plot for 3 features or a 4-D bubble plot for 4 features. More than that requires dimension reduction which is beyond the scope of the course. Nonetheless, even if you cannot visualize the data, you can still classify it via Logistic Regression. Answer the following question: is my data linearly separable? i.e. can I separate the 2 classes with a line? If not, what type of curve seems to separate the classes best? A circle? A parabola? This can be inspected visually.

   (b) Download the file (`logistic_regression_exam_final.m`) and make sure it runs on your computer. The goal now is to tweak this template code to your data and to classify your data. Try to understand this code as much as possible.

   (c) In the (`logistic_regression_exam_final.m`) code, adjust the stochastic gradient descent to randomly shuffle the training examples during each outer iteration. Currently, they are sequentially being used to update the parameters without randomization. This could lead to a cycle behavior and getting stuck in a local minima.

   (d) Run your single layer perceptron except where the activation function is now the step function discussed in class, not the sigmoid function. This is actually the original perceptron model. Do this via stochastic gradient descent. How much does the result differ from logistic regression? How different are the θ and b values? Visually, are the decision boundaries more or less the same?