**EECS 391 Introduction to Artificial Intelligence**
Fall 2017, Programming Assignment 2 (“P2”)

**Due**: Sat Dec 9 before midnight. **Total Points:** 100 + 20 extra credit

In this assignment you will train a linear classifier on the iris dataset. Like HW5, this will be convenient to do in a language with a good plotting library such as Matlab, Mathematica, or python using matplotlib. Submit both your code and your write-up (which should be a pdf file) via Canvas.

**Write-up**: Your write-up should answer the questions like in the written assignments. When relevant, include the source code in the write-up if it is sufficiently short (say half a page or less), otherwise refer to the file. For example, Exercise 2a asks you to write code to compute the mean-squared error, so you would just need to show the lines for that function.

**Exercise 1. Linear decision boundaries**

a) Inspect `irisdata.csv` which is provided with the assignment file on Canvas. Write a program that loads the iris data set and plots the 2nd and 3rd iris classes, similar to the plots shown in lecture on neural networks (i.e. using the petal width and length dimensions). 10 P.

b) Write a function that defines and plots a linear decision boundary (i.e. an arbitrary line) overlaid on the iris data. Choose a boundary (by hand) that roughly separates the two classes. 10 P.

c) Define a simple threshold classifier using the above decision boundary. Illustrate the output of the classifier using examples from each of the two classes. 10 P.

d) (Extra credit) Define a circle decision boundary using a single point as a center. Illustrate the output and performance of a classifier with three different centers. Explain how you performed the classification. +20 P.

**Exercise 2. Objective functions**

a) Write a program that calculates the mean-squared error for the iris data given a decision boundary. The function should take three arguments: the data vectors, the decision boundary parameters, and the pattern classes. 10 P.

b) Compute the mean squared error for two different decision boundaries that give large and small errors respectively. Plot both boundaries on the dataset as above. 5 P.

c) Give a mathematical derivation the gradient of the objective function above with respect to the decision boundary weights. Use $w_0$ to represent the bias term. You should show and explain each step. 10 P.

d) Show how the gradient can be written in both scalar and vector form. 5 P.

e) Write code that computes the summed gradient for an ensemble of patterns. Illustrate the gradient by showing (i.e. plotting) how the decision boundary changes for a small step. 10 P.

**Exercise 3. Learning a decision boundary through optimization**

a) Using your code above, write a program that implements gradient descent to optimize the decision boundary for the iris dataset. 5 P.

b) In your program, include code that shows the progress in two plots: the first should show the current decision boundary location overlaid on the data; the second should show the *learning curve*, i.e. a plot of the objective function as a function of the iteration. 5 P.

c) Run your code on the iris data set starting from a random setting of the weights. Note: you might need to restrict the degree of randomness so that the initial decision boundary is visible somewhere in the plot. In your writeup, show the two output plots at the initial, middle, and final locations of the decision boundary. 10 P.

d) Explain how you chose the gradient step size. 5 P.

e) Explain how you chose a stopping criterion. 5 P.