Homework #4, EE556, Fall 2018

Due: 10/11; hand in problems 4 and 5

Problem #1

6.6, Haykin's text

Problem #2

Consider a single LDF that, for augmented patterns from two classes (and with all patterns from class 2 multiplied by negative one) achieves $\underline{a}^T \underline{\tilde{y}} > b, \forall \underline{\tilde{y}}$. Show that the solution vector with minimum length is unique (this relates to uniqueness of the SVM solution). *Hint: suppose that there are two such solutions, and take their average.*

Problem #3

Consider the linear support vector machine mathematical framework developed in lecture. Specialize this development for the case where all the training vectors are orthonormal – it should be relatively easy to derive the solution in this case. How many training points are support vectors in this case ?

Problem #4

Consider M linearly independent data patterns $\underline{x}_i, i = 1, \ldots, M$, each N-dimensional, where $M < N$. Prove that there is a *linear* separator, i.e. a vector $\underline{w}$ such that $\underline{w}^T \underline{x}_i > 0 \forall i$. *Hint:* try a linear algebraic approach.

Problem #5: Multilayer Perceptron Computer Assignment

i) In this assignment, you are asked to design a multilayer perceptron to solve the XOR problem. The objective of XOR is to assign the patterns (0,0) and (1,1) to class ω_0 and the patterns (0,1) and (1,0) to class ω_1. The design will consist of three steps: i) choose an architecture capable of solving this problem (choose the number of inputs, layers, number of hidden units in each layer, activation function type for each neuron); ii) write a computer program that implements the back propagation algorithm (you can use built-in matlab functions if you prefer); iii) Choose initial parameter values and train the network using your program to minimize a sum-of-squared errors cost function over a training set consisting of the four patterns. At convergence, save the learned parameters; iv) demonstrate that your network correctly classifies all four input patterns. You should hand in your code and a plot showing the training cost function versus the number of batch gradient steps (to indicate gradient descent progress on the cost surface).

Note: you can use the Neural Network toolbox in Matlab to perform the design and to classify the patterns. Alternatively, you can implement your own back propagation algorithm (which I highly recommend as a learning experience).

ii) Now we will also investigate several real data sets from the UC Irvine repository: *glass* and *Pima Indians*. You are asked to do the following:

a) Read the given descriptions of these data sets.

b) Download these data sets from the UC Irvine machine learning web site.

c) Split into equal-sized training and test sets (Note that for *glass* this cannot be done by choosing the first half as the training set and the second half as the test set – why ?).

d) Build MLPs with different numbers of hidden units and then evaluate the training set and test set classification accuracies. Plot these performances as a function of number of hidden units.

e) Note any distinctive experimental observations.