# ECE 595 Homework 6 Due: April 3, 6 PM

1. Consider the two-class problem with the training examples and targets as given:  , . These example patterns occur with equal probability of 0.5 each. (a) Calculate the quadratic cost function, , where the **expected values** for the cross correlation matrix and vector are calculated using the discrete probability values as  with *pi* = the probability of occurrence of . (Recall the definitions of and ). For example, . Do not use any bias *x0* or θ0. (b) From (a), obtain the optimal weight vector . (c) From the hessian, what is the maximum learning rate for iterative implementation of the LMS algorithm?
2. For the two-class problem of (1) above, implement the LMS algorithm in MATLAB with α = 0.2 and initial . (a) How many iterations did your code take to converge? (b) What are the theta values? (c) Show a plot of the decision boundary and the example (training) pattern vectors.
3. A four-class problem has the following inputs.

Class 1: , ; Class 2: , ; Class 3: , ; Class 4: , ; The output is coded as , , and  for the classes 1, 2, 3 and 4, respectively.

Train an ADALINE network with two input neurons and two biases and two outputs to classify the four classes i.e., obtain the weights and biases for each of the two output values of +1 or -1. Use MATLAB code in vectorized form of the LMS algorithm to iteratively obtain the weights and biases with initial weights (parameters) including the biases as  and a learning rate of 0.2. Verify that your final weights yield correct classification of each input. (a) How many iterations did your code take? (b) What are the final weights? (c) Show the decision boundaries on the plot of the input vectors. Include your code along with the figure.

1. Apply your implementation of the ADALINE with LMS learning algorithm for the two-class training data, *hm.mat* with a learning rate in the range of 0.1 to 0.5. (Use at least two learning rates.) Use +1 and -1 for target outputs (instead of binary 1 and 0). (a) How many iterations did your code take? (b) What are the final weights? (c) Show the decision boundaries on the plot of the input vectors. (d) What is the accuracy of classification of the test data, *hmtest.mat*? Include with your code figures and answers.