**Assignment -4a**

**Title:** To study and implement ADALINE (ADaptive LINear Element) Algorithm in MATLAB.

**Theory:**

**ADALINE** (**Adaptive Linear Neuron** or later **Adaptive Linear Element**) is a single layer [neural network](http://en.wikipedia.org/wiki/Neural_network). It was developed by Professor [Bernard Widrow](http://en.wikipedia.org/wiki/Bernard_Widrow) and his graduate student [Ted Hoff](http://en.wikipedia.org/wiki/Marcian_Hoff) at [Stanford University](http://en.wikipedia.org/wiki/Stanford_University) in 1960. It is based on the [McCulloch–Pitts neuron](http://en.wikipedia.org/wiki/McCulloch%E2%80%93Pitts_neuron). It consists of a weight, a bias and a summation function.

The difference between Adaline and the standard ([McCulloch–Pitts](http://en.wikipedia.org/wiki/McCulloch%E2%80%93Pitts_neuron)) [perceptron](http://en.wikipedia.org/wiki/Perceptron) is that in the learning phase the weights are adjusted according to the weighted sum of the inputs (the net). In the standard perceptron, the net is passed to the activation ([transfer](http://en.wikipedia.org/wiki/Transfer_function)) function and the function's output is used for adjusting the weights.

Adaline is a single layer neural network with multiple nodes where each node accepts multiple inputs and generates one output. Given the following variables:

* x is the input vector
* w is the weight vector
* n is the number of inputs
* \theta is a constant
* y is the output

The output is

y=\sum_{j=1}^{n} x_j w_j + \theta

Further assumption is that

*  x_{n+1} = 1
* w_{n+1} = \theta

then the output reduces to the dot product of x and w y=x_j \cdot w_j

Architecture of Adaline Network

W0

W1

Adjust weights

o/p

1

-1

P1

P2

W3

W2

..

∑ Summation

eta = W0+W1P1+W2P2+...WmPm

P3

Wm

Pn

Adaline is a simple perceptron like system that accomplishes classification by modifying weights in such a manner so as to minimize the MSE(Mean Square Error).

**LMS/Adaline/Widrow Hoff algorithm:**

1. Start: With a randomly chosen weight vector W0. Let K=1.
2. While mean square error (mse) is unsatisfactory and computational bounds are not exceeded,
3. Do :-
4. Let Pj = (P0, P1, P2…Pm) be an input vector.   
   (Chosen randomly or in some sequence).
5. For which tj is the desired output value, with P0 = 1;   
   update the weight vector t0.  
   Wk = Wk-1 + eta\*(tj – )\*Pj.
6. Increment K;
7. End While.