Computer Science 152, Fall 2010
Assignment 2
Due Tue. 14 September 2010

Similar to assignment 1, construct a program for training an Adaline. You may modify your code in assignment 1 in doing so, but keep a copy of the original, or keep both together. Use the same data sets as before, and run each of them on the basic Adaline, plus the two variations mentioned.

Recall that the Adaline trains on the error of the activation value, rather than on the error in classification output. But you need to set the target for the activation value in the two cases for classification. Use the targets {-1, 1} rather than the classification values {0, 1} for this. In other words, if the desired output is 1, make the target activation 1, but if it is 0, make the target activation -1. The reasons for this is that the output step function changes from 0 to 1 at an argument value of 0. So if you were to target 0, you might not reach the target during training, and there is a high chance of negative samples being mis-classified. If you instead target -1, there is a much better chance that you will have a negative activation and thus give a 0 output.

The learning rate will generally need to be less than 1 in the case of the Adaline, which also means that more epochs will be required for training. For the cancer example, you might try a learning rate on the order of 0.001, with an MSE goal of 1e-6. Remember that the MSE is the mean *squared* error, so the actual error is on the order of 1e-3. Experiment with a few different learning rates to see if you can determine the boundary between convergence and non-convergence.

Variation 1: Try batch learning: Instead of applying a weight update at each step, the would-be weight changes are accumulated and the weight update rule is applied to the sum across an entire epoch.

Variation 2: Try a logistic sigmoid ("logsig") activation function (with on-line, i.e. non-batch, learning). The formula for the logsig function and its derivative is given in the notes. The weight update formula for this case was also derived. In this case, however, you should be able to train on the overall Adaline output directly, rather than deriving a target activation, as the desired values are 0 or 1, which lie at the extremes of the logsig function's range. For classification purposes, consider the output to be 0 or 1 depending on whether the neurons output is below or above 0.5, the middle of the range.

The variations will help with experience when we implement multi-level networks.

What to submit (please use paper for this assignment):
Same as assignment 1, except make cross-comparisons between the main method and its variations. Also make comparisons between Perceptron and Adaline learning.