Computer Science 152, Fall 2006

Assignment 3
Due Monday 25 September 2006

In preparation for constructing a more general back-propagation program, construct a 2-layer neural network as follows:

- Two input values
- First layer: 2 logistic functions $f(u) = 1/(1+e^{-u})$
- Second layer: 1 linear function $f(u) = u$
- One output value

1. First check the back-propagation operation using the example worked in class. That was a 1-input example, so set the other input to 0. Here you will manually set the weights.

2. Verify that your network can be trained to implement the 2-input XOR function. Here it might be easier to read the samples using your program from assignment 2. As with the Adaline assignment, convert the output of the linear unit to 1 if > 0.5 and 0 otherwise, so that the evaluation output is always 0 or 1. As with the Adaline, this evaluation plays no role in learning. You may use the same initial weights as in part1. A learning rate of 0.1 is suggested. Use on-line learning.

3. Repeat part 2 with random initial weights. These should be small signed values, around ≤ 0.1 magnitude. The IEEE standard function drand48 is suggested for generating random numbers.

4. [Extra credit] Repeat 3 with batch learning and compare the two versions.

5. Submit a draft of your design for a multi-layer network. This might include features such as:

    a. A variety of specifiable transfer functions (each with its own derivative). It can be assumed that all functions within a layer are the same.

    b. Specifiable number of layers and neurons per layer

    c. The usual input parameters on the command-line, or in a configuration file separate from the data

    d. Batch vs. on-line learning specifiable as a parameter

    e. Other features you might want to have

    The data format will be the same as on previous assignments.