Neural Nets 2

Some further tutorial questions culled from various sources.

1. Consider a perceptron with a single input, x, and for which the following holds:

    - $g()$ is $sgn()$ and we will take $sgn'(x_m) = 1$
    - The initial weights on the neuron are: $w_0 = 1.5$ and $w_1 = 0$
    - The input/output training data contains two examples: input = ([2], [1]), output = ([3], [-1])
    - The learning rate, η is 0.1.

    By applying the perceptron learning rule show how the weight change over two epochs.

2. Figure 1 shows a back propagation netowork that is currently proscessing the training vector [1.0, 0.9, 0.9] and the associated target vector is [0.1, 0.9, 0.1].

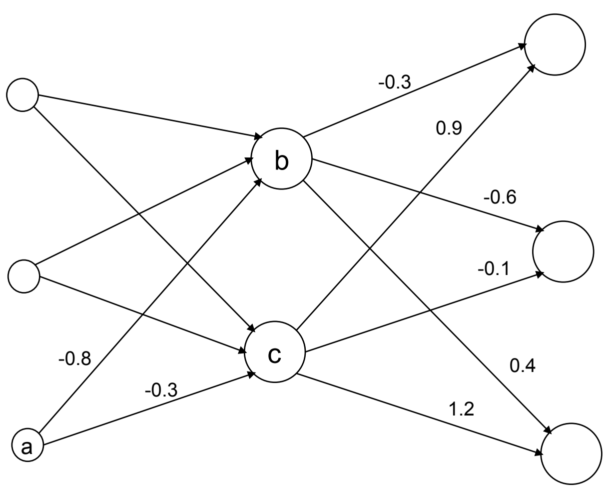

Figure 1:

Given that the output from unit b is 0.6 and from c is 0.8, and assuming that the logistic function is the activation function:

(a) Calculate the actual output vector.

(b) Calculate the error for each output unit.

(c) Calculate the error for each hidden unit.

(d) Calculate the weight changes for the weights connecting from unit a. Use a learning rate of 0.25.

3. Derive a suitable feed-forward network that models the logical AND.

4. A simple perceptron cannot represent XOR (or, generally, the parity function of its inputs). Describe what happens to the weights of a four-input, step function perceptron, beginning with all weights set to 0.1, as examples of the parity function arrive.

5. Suppose you had a neural net with linear activation functions. That is, for each unit the output is some constant c times the weighted sum of the inputs.

   (a) Assume that the network has one hidden layer. For a given assignment to the weights $\mathbf{W}$, write down equations for the value of the units in the output layer as a function of $\mathbf{W}$ and the input layer $\mathbf{I}$, without any explicit mention to the output of the hidden layer. Show that there is a network with no hidden units that computes the same function.

   (b) Repeat the calculation in part (i), this time for a network with any number of hidden layers. What can you conclude about linear activation functions?

6. Suppose that a training set contains only a single example, repeated 100 times. In 80 out of the hundred cases the single output value is 1; in the other 20 it is 0. What will a back-propagation network predict for this example, assuming that it has been trained and reaches a global optimum? (*Hint:* to find the global optimum, differentiate the error function and set to zero.)