Backpropagation Homework

For this homework you will work through training a multi-layer perceptron using the back propagation algorithm on **one** training instance (the second training instance below, 0 1 -> 0). The first training instance is done for you to give an example. You should **use the updated weights** after training on the first instance when training on the second instance. Assume a bias of 1.

Example with first training pattern:

BP-1) A 2-2-1 backpropagation model has initial weights as shown. Work through one cycle of learning for the following pattern(s). Assume 0 momentum and a learning constant of 1. Round calculations to 3 significant digits to the right of the decimal. Give values for all nodes and links for activation, output, error signal, weight delta, and final weights. Nodes 4, 5, 6, and 7 are just input nodes and do not have a sigmoidal output.

For each node calculate the following (show necessary equation for each). Hint: Calculate bottom-top-bottom.

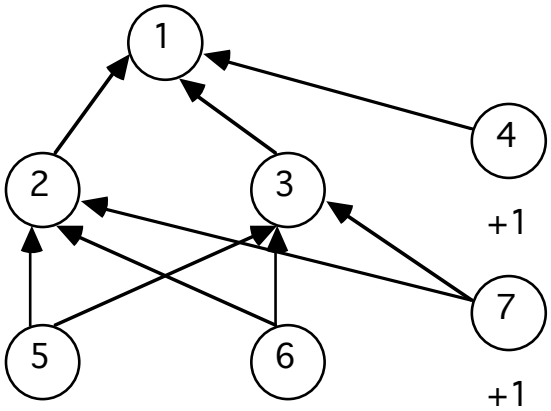
a =

o =

δ =

Δw =

w =



a) All weights initially 1.0

Training Patterns

1) 0 0 -> 1

2) 0 1 -> 0

BP-1)

net2 = Σ w_i x_i = (1*0 + 1*0 + 1*1) = 1

net3 = 1

o2 = 1/(1+e^{-net}) = 1/(1+e^{-1}) = 1/(1+.368) = .731

o3 = .731

o4 = 1

net1 = (1*.731 + 1*.731 + 1) = 2.462

o1 = 1/(1+$e^{-2.462}$)= .921

$\delta 1$ = (t1 - o1) o1 (1 - o1) = (1 - .921) .921 (1 - .92

$\Delta w21 = \eta \delta_j o_i = \eta \delta_1 o_2$ = 1 * .00575 * .731 = .

$\Delta w31$ = 1 * .00575 * .731 = .00420

$\Delta w41$ = 1 * .00575 * 1 = .00575

$\delta 2 = o_j (1 - o_j) \Sigma \delta_k w_{jk} = o_2 (1 - o_2) \delta_1 w_{21}$ =

.731 (1 - .731) (.00575 * 1) = .00113

$\delta 3$ = .00113

$\Delta w52 = \eta \delta_j o_i = \eta \delta_2 o_5$ = 1 * .00113 * 0 = 0

$\Delta w62$ = 0

$\Delta w72$ = 1 * .00113 * 1 = .00113

$\Delta w53$ = 0

$\Delta w63$ = 0

$\Delta w73$ = 1 * .00113 * 1 = .00113

Output: $O_j = f(net_j) = \frac{1}{1+e^{-net_j}}$

$f'(net_j) = \frac{\partial O_j}{\partial net_j} = O_j(1 - O_j)$

Δw_{ij} *(general node):* $C\ O_i\ \delta_j$

Δw_{ij} *(output node):*

$$\delta_j = (t_j - O_j) f'(net_j)$$

$$\Delta w_{ij} = C\ O_i\ \delta_j = C\ O_i\ (t_j - O_j) f'(net_j)$$

Δw_{ij} *(hidden node)*

$$\delta_j = \sum_k (\delta_k \cdot w_{jk})\ f'(net_j)$$

$$\Delta w_{ij} = C\ O_i\ \delta_j = C\ O_i\ \left(\sum_k (\delta_k \cdot w_{jk}) \right) f'(net_j)$$