

## Deep Learning –HW#4

### About the Assignment

The main aim of the assignment is to make you familiar with the Multi Layer Neural Networks. Please solve the problems individually and cheating will be harshly punished.

## Backpropagation Algorithm

Error term for output units:

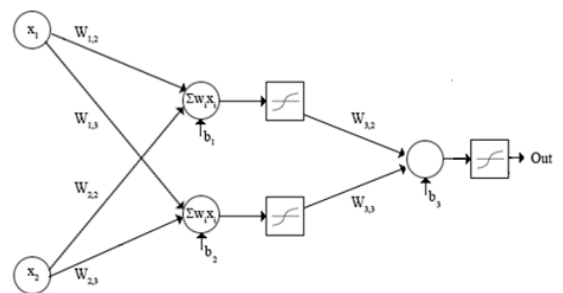
$$\delta_i = -\frac{\partial E_d}{\partial \text{sum}_{id}} = -\frac{\partial E_{id}}{\partial \text{sum}_{id}} = (t_{id} - o_{id}) o_{id} (1 - o_{id})$$

Error term for hidden units:

$$\delta_i = -\frac{\partial E_d}{\partial \text{sum}_{id}} = o_{id} (1 - o_{id}) \sum_{k \in \text{Outputs}} w_{ki} \delta_k$$

3. Compute all weights changes

$$\Delta w_{ij} = \eta \delta_i x_j$$



**Learning Rule**

$$w_{m+1} = w_m + \Delta w_m$$

*Hidden to output*

$$\Delta w_{kj} = \eta \delta_k y_j = \eta (t_k - z_k) f'(net_k) y_j$$

*Input to Hidden*

$$\Delta w_{ji} = \eta \delta_j x_i = \eta \left[ \sum_{k=1}^c w_{kj} \delta_k \right] \underbrace{f'(net_j)}_{\delta_j} x_i$$

$c = \text{number of connected weights}$

$c = 1 \text{ for this case}$

### Submit the Assignment

Ex: Name\_Surname\_NO\_HW#.py

### Hint

You can look the implementations given in Matlab.

### Step 1:

Implement both oh train and test functions for given Matlab codes about NN.

The test function should take only 1 sample\_test and required parameters.

Plotting iterations and errors in run time will be graded as 10 points.

You can use any code but you have to implement the same concept given in Matlab codes.