BackPropagation (training examples, η, n_i, n_j, n_k)

* Create a feed-forward network with n_i inputs, n_j hidden units & n_k output units
* Initialise all n/w weights to small random nos. (−0.5 to 0.5)
* Until the termination condition is met, Do
  - For each <x, t> in training examples, Do

Propagate the input forward through the network:

1) Input the instance x to the network & compute the output y of every unit in the network

i.e.

I/p & o/p of Hidden layer

$$\boxed{\begin{aligned} X_j &= \sum x_i w_{ij} \\ y_j &= \sigma(X_j) = \frac{1}{1+e^{-X_j}} \end{aligned}}$$

$$\boxed{\begin{aligned} X_k &= \sum y_j w_{jk} \\ y_k &= \frac{1}{1+e^{-X_k}} \end{aligned}}$$

I/p & o/p of output layer

Propagate the errors backward through the network:

2) For each network output unit k, calculate its error term δ_k

$$\boxed{\delta_k \leftarrow y_k(1-y_k)(T-y_k)} \rightarrow \text{Error at Output layer}$$

T → target / Expected output

3) For each hidden unit j, calculate its error term δ_j

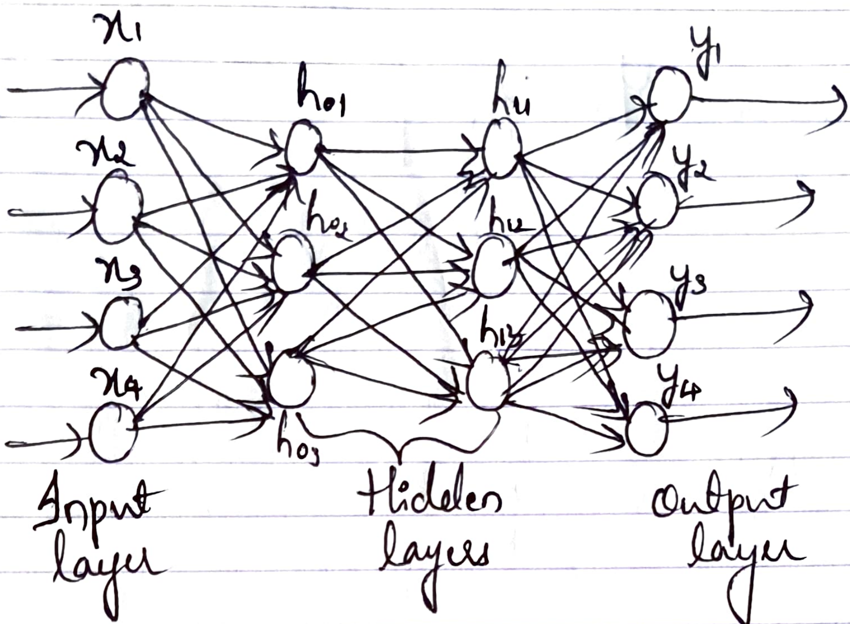
$$\boxed{\delta_j \leftarrow y_j(1-y_j)\sum \delta_k w_{jk}} \rightarrow \text{Error at Hidden layer}$$

4) Update each weight

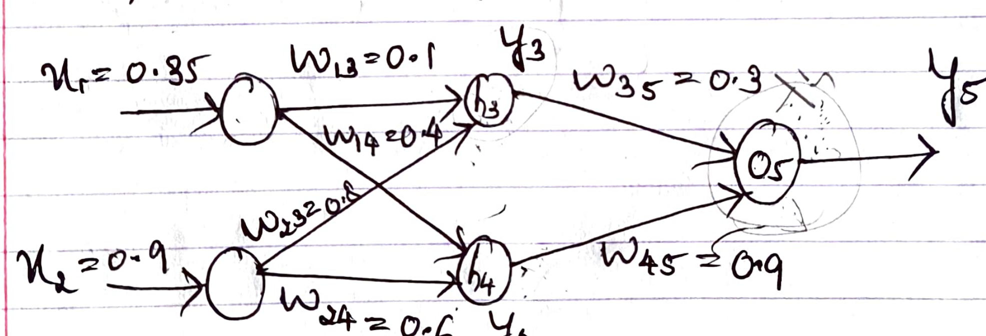
$$\boxed{w_{ij}(n) = w_{ij}(o) + \eta\delta_j x_i} \rightarrow \text{I/p to hidden layer}$$

$$\boxed{w_{jk}(n) = w_{jk}(o) + \eta\delta_k y_j} \rightarrow \text{Hidden to o/p layer}$$

Backpropagation Algorithm



Expected ~~Actual~~ output $y = 0.5$ $\alpha = 1$



Forward Pass

Compute output for y_3, y_4 & y_5

$$X_3 = x_1 w_{13} + x_2 w_{23}$$

$$\boxed{X_j = \sum x_i w_{ij}}$$

$$\boxed{Y_j = \text{Sigmoid}(X_j) = \frac{1}{1+e^{-X_j}}}$$

$$X_j = \sum x_i w_i$$

$$y_j = \frac{1}{1+e^{-X_j}}$$

Forward Pass

$$\boxed{\text{Network Error} = Y_{t} - Y_{a}}$$

$X_3 = x_1 w_{13} + x_2 w_{23}$

$X_3 = 0.35 \times 0.1 + 0.9 \times 0.8 = 0.755$

$$y_3 = \frac{1}{1+e^{-X_3}} = \frac{1}{1+e^{-0.755}} = 0.68$$

$X_4 = x_1 w_{14} + x_2 w_{24}$

$X_4 = 0.35 \times 0.4 + 0.9 \times 0.6 = 0.68$

$$y_4 = \frac{1}{1+e^{-0.68}} = \frac{1}{1+e^{-X_4}} = 0.6637$$

$X_5 = y_3 \times w_{35} + y_4 \times w_{45}$

$X_5 = 0.68 \times 0.3 + 0.6637 \times 0.9 = 0.801$

$$y_5 = \frac{1}{1+e^{-0.801}} = \boxed{0.69}$$ Network output

$\boxed{\text{Error} = y_{target} - y_5} = 0.5 - 0.69 = -0.19$

Weight Adjustment Formula

$$w_{ji}(\text{old}) = w_{ji}(\text{new}) + \underbrace{\eta \delta_j y_i}_{\Delta w_{ji}}$$

Output unit

$$\boxed{\delta_j = y_j(1-y_j)(d_j - y_j)}$$

Hidden unit

$$\boxed{\delta_j = y_j(1-y_j)\sum \delta_k w_{kj}}$$

For output unit (Error)

$$\delta_5 = Y_5(1-Y_5)(t-Y_5)$$

$$\delta_5 = 0.69(1-0.69)(0.5-0.69) = \underline{-0.0406}$$

Error term at hidden layer

$$\delta_3 = Y_3(1-Y_3)\sum \delta_5 w_{35}$$

$$\delta_3 = 0.68(1-0.68) \times (-0.0406 \times 0.3) = \underline{-0.00265}$$

$$\delta_4 = Y_4(1-Y_4)\sum \delta_5 w_{45}$$

$$= 0.6637(1-0.6637) \times (-0.0406 \times 0.9) = \underline{-0.0082}$$

Adjusting the Weights

$$\Delta w_{ij}(n) = \eta \delta_j Y_i$$

$$w_{ij}(n) = w_{ij}(o) + \eta \delta_j Y_i$$

$$w_{35}(n) = 0.3 + 1 \times -0.0406 \times 0.68 =$$

$$w_{45}(n) = 0.9 + 1 \times -0.0406 \times 0.6637 = 0.8731$$

$w_{13} = 0.1 + 1 \times (-0.00265) \times 0.35 =$

$w_{14} = 0.4 + 1 \times (-0.0082) \times 0.35 = 0.3971$

2nd Epoch

$X_3 = 0.35 \times 0.0991 + 0.9 \times 0.7976 = 0.7525$

$Y_3 = \frac{1}{1+e^{-0.7525}} = 0.6797$ $X_3 = 0.6723$

$Y_4 = 0.6620$