ALDEL EDUCATION TRUST'S
ST. JOHN COLLEGE OF ENGINEERING AND TECHNOLOGY

Date

BPN :-

$$e_j(n) = d_j(n) - y_j(n) \quad - (1)$$

$$\mathcal{E}(n) = \frac{1}{2} \sum_{j \in c} e_j^2(n) \quad - (2)$$

$$\mathcal{E}_{avg} = \frac{1}{N} \sum_{n=1}^{N} \mathcal{E}(n) \quad - (3)$$

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n) \quad - (4)$$

$$y_j(n) = \varphi_j(v_j(n)) \quad - (5)$$

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \cdot \frac{\partial e_j(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} \cdot \frac{\partial v_j(n)}{\partial w_{ji}(n)} \quad - (6)$$

$$\frac{\partial \mathcal{E}(n)}{\partial e_j(n)} = e_j(n) \quad - (7)$$

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \quad - (8)$$

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \varphi_j'\left[v_j(n)\right] \quad - (9)$$

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n) \quad - (10)$$

Substituting (7), (8), (9) & (10) into (6)

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = -e_j(n)\, \varphi_j'(v_j(n))\, y_i(n). \quad - (11)$$

Now,

$$\Delta w_{ji}(n) \propto \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)}$$

$$\therefore \Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} \quad - (12)$$

Substituting (11) into (12)

$$\therefore \Delta w_{ji}(n) = -\eta \left[-e_j(n)\, \varphi_j'(v_j(n))\, y_i(n) \right]$$

$$= \eta\, e_j(n)\, \varphi_j'(v_j(n))\, y_i(n) \quad - (13)$$

ALDEL EDUCATION TRUST'S
ST. JOHN COLLEGE OF ENGINEERING AND TECHNOLOGY

Date

Now, the Local Gradient is defined by:

$$\delta_j(n) = -\frac{\partial E(n)}{\partial v_j(n)}$$

Again, applying chain rule:

$$= -\frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)}$$

Substituting their respective values:

$$\therefore \delta_j(n) = e_j(n)\, \varphi_j'(v_j(n)) \quad - (14)$$

Substituting (14) into (13):

$$\Delta w_{ji}(n) = \eta\, \delta_j(n)\, y_i(n) \quad - (15)$$

Case 1: Neuron j is an O/p node.

When neuron j is located in the o/p Layer, it is supplied with a desired response of its own. So, the computation of the error s/g is direct & rest of the equations ~~are as~~ follows.

Case 2: Neuron j is a Hidden Node.

As neuron j is in the hidden Layer, there is no specified disired sig for that neuron. Accordingly, the error sig for the hidden neuron would have to be determined recursively in terms of error signals of all the neurons to which the hidden neuron is directly connected.

The local gradient is redefined as

$$\delta_j(n) = -\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} \quad — \text{(16)}$$

Substituting from (4) into (16).

$$\delta_j(n) = -\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} \varphi_j'(v_j(n)) \quad — \text{(16a)}$$

The instantaneous energy for the k^{th} neuron is:

$$\mathcal{E}(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n) \quad — \text{(17)}$$

$$\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial y_j(n)} \quad — \text{(18)}$$

Substituting eq^{n}s (21) & (23) into (19).

$$\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} = -\sum_k e_k(n)\,\varphi'_k(v_k(n))\,w_{kj}(n). \quad (24)$$

From eq^n (19), we know that:

$$\delta_k(n) = e_k(n)\,\varphi'_k(v_k(n))$$

Substituting

$$\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} = -\sum_k \delta_k(n)\,w_{kj}(n) \quad - (24)$$

Substituting (24) into (16a)

$$\therefore \delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n)\,w_{kj}(n). \quad (25)$$

This is the formula for back propagation.

& the original weight correction formula can be given in statement as

$$\begin{pmatrix}\text{Weight}\\ \text{correction}\\ \Delta w_{ji}(n)\end{pmatrix} = \begin{bmatrix}\text{learning}\\ \text{rate parameter}\\ \eta\end{bmatrix} \cdot \begin{pmatrix}\text{Local}\\ \text{gradient}\\ \delta_j(n)\end{pmatrix} \cdot \begin{pmatrix}\text{i/p signal}\\ \text{of neuron } i\\ y_i(n)\end{pmatrix}$$

Stopping Criteria:

In general, BPN alg cannot be shown to converge. Rather there are some reasonable criteria

The BP Algo; is considered to have converged when ~~to~~ the absolute rate of change in the average squared error per epoch is sufficiently small, i.e., 0.1% to 1% per epoch

There are 2 phases to the algo

1] forward phase:-

During this phase, the free parameters of the networks are fixed & i/p is propagated layer by layer. The forwardward phase finishes with computation of error signal

$$e_i = d_i - y_i$$

2] Backward phase:-

During the second phase, the error signal is propagated through the n/w in the backward direction. Hence the name.

During this phase, adjustments are applied to the free parameters so as to minimize error e_i

Implementation by following mode:-

Sequential Mode:

Adjustments are made to the free parameters of the n/w on an example by example basis

$$N \Rightarrow O\left(\frac{W}{\varepsilon}\right)$$

$N \Rightarrow$ Size of training set
$O \Rightarrow$ Order of
$W \Rightarrow$ Total no. of synaptic weights including the bias weights
$\varepsilon \Rightarrow$ fraction of classification errors that are permitted eg:- with an error of 10% the number of training examples needed should be 10 times the number of synaptic weights