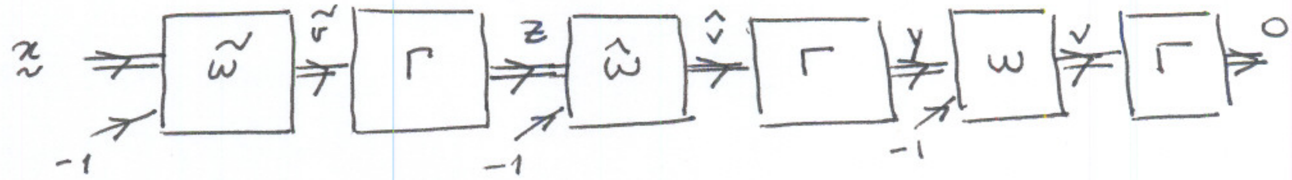
Back Propagation Algorithm



$x \in \mathbb{R}^m \qquad \tilde{v}, z \in \mathbb{R}^q \qquad \hat{v}, y \in \mathbb{R}^p \qquad v, o \in \mathbb{R}^R$

$x_e = \begin{pmatrix} x \\ -1 \end{pmatrix} \in \mathbb{R}^{m+1} \qquad z_e = \begin{pmatrix} z \\ -1 \end{pmatrix} \in \mathbb{R}^{q+1} \qquad y_e = \begin{pmatrix} y \\ -1 \end{pmatrix} \in \mathbb{R}^{p+1}$

m: # of input components (excluding treshold)

q: # of neurons in 1st layer

p: # of neurons in 2nd layer

R: # of neurons in 3rd = output layer

$\tilde{w} \in \mathbb{R}^{q \times m} \qquad \tilde{w}_e \in [\tilde{w} \,\vdots\, \widetilde{\text{treshold}}] \in \mathbb{R}^{q \times (m+1)}$

$\hat{w} \in \mathbb{R}^{p \times q} \qquad \hat{w}_e = [\hat{w} \,\vdots\, \widehat{\text{treshold}}] \in \mathbb{R}^{p \times (q+1)}$

$w \in \mathbb{R}^{R \times p} \qquad w_e = [w \,\vdots\, \text{treshold}] \in \mathbb{R}^{R \times (p+1)}$

$S_{Tr} = \{ x^1, x^2, \dots, x^N \}$

n : pattern index

$x(n) \rightarrow \tilde{v}(n) \rightarrow z(n) \rightarrow \hat{v}(n) \rightarrow y(n) \rightarrow v(n) \rightarrow o(n)$

$\boxed{e(n) = d(n) - o(n)} \qquad \boxed{e_i(n) = d_i(n) - o_i(n)} \quad i = 1, 2, \dots, R$

$\boxed{E(n) = \frac{1}{2} \| d(n) - o(n) \|^2 = \frac{1}{2} \sum_{i=1}^{n} e_i^2(n)} \qquad \boxed{E_{cum} = \sum_{n=1}^{N} E(n)}$

$\boxed{E_{ave} = \frac{1}{N} E_{cum}}$

Back Prop. : Componentwise Update.

→ Select learning coefficient
→ set $\tilde{w}(1)$, $\hat{w}(1)$, $w(1)$ RANDOMLY.

→ pick a pattern from training set. say $n^{\underline{th}}$ pattern
compute [FORWARD PROPAGATION]
$x(n) \Rightarrow \tilde{v}(n) \Rightarrow z(n) \Rightarrow \hat{v}(n) \Rightarrow y(n) \Rightarrow v(n) \Rightarrow o(n)$

→ Compute $e_j(n) = d_j(n) - o_j(n)$, $j = 1, 2, .., R$.

→ Compute local gradients of output layer

$$\delta_j(n) = f'(v_j(n)) \cdot e_j(n) \qquad j = 1, 2, .., R$$

→ compute local gradients of 2nd layer

$$\hat{\delta}_j(n) = f'(\hat{v}_j(n)) \cdot \sum_{k=1}^{R} \delta_k(n) \cdot w_{kj}(n) \qquad j = 1, 2, \ldots, P$$

→ compute local gradients of 1st layer

$$\tilde{\delta}_j(n) = f'(\tilde{v}_j(n)) \cdot \sum_{k=1}^{P} \hat{\delta}_k(n) \cdot \hat{w}_{kj}(n) \qquad j = 1, 2, \ldots, q.$$

→ update output layer weights:

$$w_{ji}(n+1) = w_{ji}(n) + \eta\, \delta_j(n) \cdot y_i(n) \qquad \begin{array}{l} j = 1, 2, .., R \\ i = 1, 2, .., P, \underline{P+1} \Rightarrow \text{for treshold.} \end{array}$$

→ update 2nd layer weipts.

$$\hat{w}_{ji}(n+1) = \hat{w}_{ji}(n) + \eta\, \hat{\delta}_j(n) \cdot z_i(n) \qquad \begin{array}{l} j = 1, 2, .., P \\ i = 1, 2, .., q, \underline{q+1} \Rightarrow \text{for treshold} \end{array}$$

→ update 1st layer weights

$$\tilde{w}_{ji}(n+1) = \tilde{w}_{ji}(n) + \eta\, \tilde{\delta}_j(n) \cdot x_i(n) \qquad \begin{array}{l} j = 1, 2, .., q \\ i = 1, 2, .., m, \underline{m+1} \Rightarrow \text{for treshold.} \end{array}$$

→ compute error
$$E(n) = \frac{1}{2} \sum_{i=1}^{R} e_i^2(n)$$

→ is there a new pattern in the training set?

YES

NO
↓ compute Eave.
↓ if stopping criterion is satisfied: STOP
if not: Start a new epoch.

Back . prop: MATRIX UPDATE

(1) → same for the forward phase.

→ $e(n) = d(n) - o(n)$

→ compute the local gradient of output layer.

$$\delta(n) = \Gamma'(v(n))\, e(n) \qquad \Gamma'(v) = \begin{bmatrix} f'(v_1(n)) & & 0 \\ & \ddots & \\ 0 & & f'(v_R(n)) \end{bmatrix}$$

→ compute the local gradient of 2nd layer

$$\hat{\delta}(n) = \Gamma'(\hat{v}(n)) \cdot W^T(n) \cdot \delta(n)$$

→ compute the local gradient of 1st layer.

$$\tilde{\delta}(n) = \Gamma'(\tilde{v}(n)) \cdot \hat{W}^T(n) \cdot \hat{\delta}(n)$$

→ update output layer (extended) matrix.

$$W_e(n+1) = W_e(n) + \eta\, \delta(n) \cdot y_e^T(n)$$

→ update 2nd layer (extended) matrix

$$\hat{W}_e(n+1) = \hat{W}_e(n) + \eta\, \hat{\delta}(n) \cdot z_e^T(n)$$

→ update 1st layer (extended) matrix

$$\tilde{W}_e(n+1) = \tilde{W}_e(n) + \eta\, \tilde{\delta}(n) \cdot x_e^T(n)$$

→ compute error : $E(n) = \frac{1}{2} \sum_{i=1}^{R} e_i^2(n) = \frac{1}{2} \| d(n) - o(n) \|^2$

• Is there a new pattern in the training set?

YES → go to (1)

NO ↓

Find $E_{cum} = \sum_{n=1}^{N} E(n)$, $E_{ave} = \frac{1}{N} E_{cum}$.

If stopping criterion is satisfied: STOP

else START ~~A~~ NEW EPOCH.

Note: $f'(v) = \frac{\lambda}{2}(1 - o^2)$ for bipolar sigmoidal

$f'(v) = \lambda o (1 - o)$ for unipolar sigmoidal.