1. Backpropagation Algorithm,

for hidden layer,

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

for output layer,

$$f(x) = x \Rightarrow f'(x) = 1 \qquad -(1)$$

Calculating derivatives of activation function for hidden & output layer,

for hidden layer,

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$f'(x) = \frac{(e^x + e^{-x})(e^x + e^{-x}) - (e^x - e^{-x})(e^x - e^{-x})}{(e^x + e^{-x})^2}$$

$$= \frac{(e^x + e^{-x})^2 - (e^x - e^{-x})^2}{(e^x + e^{-x})^2}$$

$$= 1 - \frac{(e^x - e^{-x})^2}{(e^x + e^{-x})^2}$$

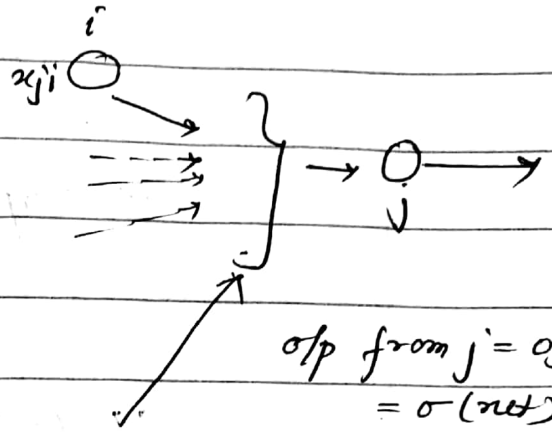
$$= 1 - (\tanh(x))^2$$

$$\Rightarrow f'(x) = 1 - (f(x))^2 \qquad -(2)$$

For output layer

$$f(x) = x$$

$$\Rightarrow \quad f'(x) = 1$$



Algorithm,

$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} \times \frac{\partial net_j}{\partial w_{ji}}$$

o/p from $j = o_j$
$= \sigma(net)$

net coming into $j = net_j$
$= \sum_i w_{ji} x_{ji}$

$\frac{\partial E_d}{\partial net_j} \rightarrow$ 2 cases: hidden unit / output unit

<u>Case 1, $j \in$ o/p unit</u>

$$\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial o_j} \times \underbrace{\frac{\partial o_j}{\partial net_j}} \rightarrow \text{Activation function}$$

$$\Rightarrow \quad E_d = \frac{1}{2} \sum_{k \in o/p} (t_k - o_k)^2$$

$$\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j}\left[\frac{1}{2} \sum_{k \in o/p} (t_k - o_k)^2\right]$$

(from (1))

$$\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j}\left[\frac{1}{2} (t_j - o_j)^2\right] \quad ; \quad \frac{\partial o_j}{\partial net_j} = 1$$

$$= -(t_j - o_j)$$

$$\Rightarrow \quad -\delta_j = \frac{\partial E_d}{\partial net_j} = -(t_j - o_j) \times 1$$

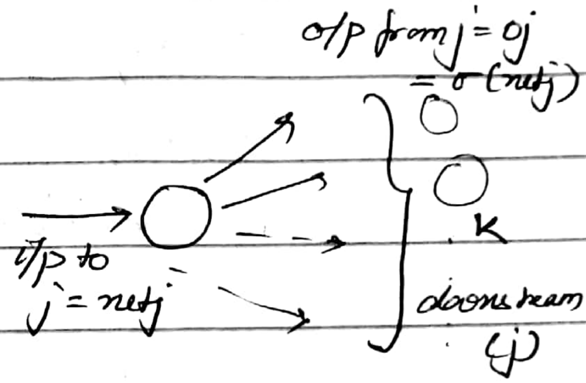
$$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}}$$

$$\Delta w_{ji} = \eta (t_j - o_j)\, x_{ji}$$

$$\Delta w_{ji} = \eta\, \delta_j\, x_{ji}$$

Case 2, $j \in$ hidden unit

$$\frac{\partial E_d}{\partial net_j} = \sum_{k \in downstream(j)} \frac{\partial E_d}{\partial net_k} \cdot \frac{\partial net_k}{\partial net_j}$$



$$= \sum_{k \in down\ stream(j)} -\delta_k \frac{\partial net_k}{\partial net_j}$$

$$= \sum_{k \in down\ stream(j)} -\delta_k \frac{\partial net_k}{\partial o_j} \times \underbrace{\frac{\partial o_j}{\partial net_j}}_{\text{Activation function}}$$

$$= \sum_{k \in down\ stream(j)} -\delta_k\, w_{kj}\, (1 - o_j^2)$$

$$\delta_j = (1 - o_j^2) \sum_{k \in down\ stream} \delta_k\, w_{kj}$$

$$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}}$$

$$\Delta w_{ji} = \eta\, \delta_j\, x_{ji}$$

// [(o/p) Case 1 : $\delta_j = (t_j - o_j)$
(hidden) case 2 : $\delta_j = (1 - o_j^2) \sum_{k \in \text{down stream}} \delta_k \omega_{kj}$

2. Gradient Descent - training rule

$$o = \omega_0 + \omega_1(x_1 + x_1^2) + \cdots + \omega_n(x_n + x_n^2)$$

$x_1, x_2, \ldots, x_n$: Inputs

$\omega_1, \omega_2, \ldots \omega_n$: weights

Activation function, $f(x) = x$

Training rule,

$$\cancel{\nabla E(\omega) =} \quad \Delta \vec{\omega} = -\eta \nabla E(\omega)$$

$$\Delta \omega_i = \eta \frac{\partial E}{\partial \omega_i}$$

$$\frac{\partial E}{\partial \omega_i} = \frac{\partial}{\partial \omega_i} \frac{1}{2} \sum_d (t_d - O_d)^2$$

$$= \frac{1}{2} \sum_d \frac{\partial}{\partial \omega_i} (t_d - O_d)^2$$

$$= \sum_d (t_d - O_d) \frac{\partial}{\partial \omega_i} (t_d - O_d)$$

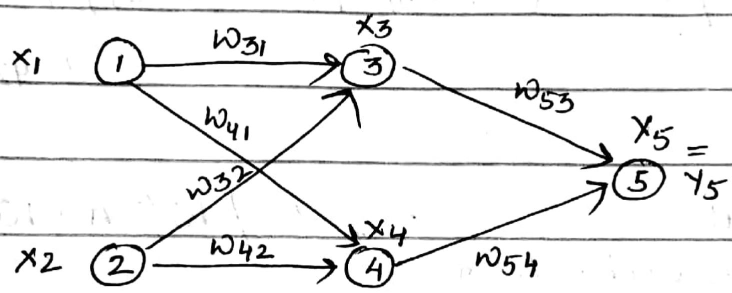
$$= \sum_d (t_d - O_d) \frac{\partial}{\partial \omega_i} (t_d - \vec{\omega} \cdot \vec{x_d})$$

$$\frac{\partial E}{\partial \omega_i} = \sum_d (t_d - O_d) (-(x_d + x_{id}^2))$$

$$\Delta w_i = \eta \sum_{d \in D} (t_d - O_d)(x_{id} + x_{id}^2)$$

η : learning rate,
t_d : target output,
O_d : Actual output,
x_{id} = value of i^{th} attribute of d^{th} training eg.

3.



I/P

Activation function, $f(x) = x$
hidden - h

a) Forward Pass

Node	Net	output
1	i_1	$x_1 = i_1$
2	i_2	$x_2 = i_2$
3	$net_3 = w_{31}x_1 + w_{32}x_2$	$x_3 = \cancel{f(net)}\ h(net_3)$
4	$net_4 = w_{41}x_1 + w_{42}x_2$	$x_4 = h(net_4)$
5	$net_5 = w_{53}x_3 + w_{54}x_4$	$x_5 = h(net_5)$

$$Y_5 \ (X_5) = h(net_5)$$

$$= h(w_{53}x_3 + w_{54}x_4)$$

$$Y_5 = h(w_{53}(h(w_{31}x_1 + w_{32}x_2)) + w_{54}(h(w_{41}x_1 + w_{42}x_2)))$$

Scanned by CamScanner

(b) $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

$w^{(1)} = \begin{pmatrix} w_{31} & w_{32} \\ w_{41} & w_{42} \end{pmatrix}$

$w^{(2)} = (w_{53} \quad w_{54})$

from prev. question,

$y_5 = h\left[w_{53}\left(h(w_{31}x_1 + w_{32}x_2)\right) + w_{54}\left(h(w_{41}x_1 + w_{42}x_2)\right)\right]$

output in vector form $= h\left[W^2 h(W^1 x)\right]$

(c) $h_1(x) = \frac{1}{1+e^{-x}}$

$h_2(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

$$h_2(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \times \frac{e^{-x}}{e^{-x}}$$

$$h_2(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

$$h_2(x) = h_1(2x)\left[(1 - e^{-2x}) + 1 - 1\right]$$

$$h_2(x) = h_1(2x)\left[2 - \frac{1}{h_1(2x)}\right]$$

$$\underline{h_2(x) = 2h_1(2x) - 1}$$

$$h_1(x) = \frac{1}{1+e^{-x}} \quad \& \quad h_1'(x) = h(x)\,(1 - h_1(x)) \leftarrow \text{sigmoid}$$

$$h_2(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad \& \quad h_2'(x) = 1 - (h_2(x))^2$$

substituting $h_1(x)$ & $h_2(x)$ to output functions,

For sigmoid,

$$\underline{\delta_j = (t_j - o_j)\, o_j (1 - o_j)} \quad \text{— (1)}$$

for $\tanh x$,

$$\delta_j = (t_j - o_j)\,(1 - (o_j)^2)$$

$$\underline{\delta_j = (t_j - o_j)\,(1 + o_j)(1 - o_j)} \quad \text{— (2)}$$

Thus we observe that the two output functions are same (1) & (2) with parameters differing only by linear transformations & constants.

4. ~~$\Sigma(\omega)$~~

$$E(\vec{\omega}) = \frac{1}{2} \sum_{d \in D} \sum_{k \in o/p} (t_{kd} - O_{kd})^2 + \gamma \sum_{i,j} \omega_{ji}^2$$

Gradient Descent update for E

Update rule :

$$\Delta \omega_{ji} = -\eta \frac{\partial E(\vec{\omega})}{\partial \omega_{ji}}$$

$$\omega_{ji}^{new} \leftarrow \omega_{ji}^{old} + \Delta \omega_{ji}$$

$$\frac{\partial E(\vec{\omega})}{\partial \omega_{ji}} = \frac{\partial}{\partial \omega_{ji}} \left[\frac{1}{2} \sum_{d \in D} \sum_{k \in o/p} (t_{kd} - O_{kd})^2 + \gamma \sum_{ij} \omega_{ji}^2 \right]$$

$$= -(t_{kd} - O_{kd})(1 - O_{kd}) O_{kd} x_{ji} + 2\gamma \omega_{ji}$$

$$\omega_{ji} \leftarrow \omega_{ji} + \eta \delta_{kd} x_{ji} - 2\gamma\eta \omega_{ji}$$

η : learning rate

Since,

$$(t_{kd} - O_{kd})(1 - O_{kd}) O_{kd} = \delta_{kd}$$

$$\omega_{ji} \leftarrow \eta \delta_{kd} x_{ji} + \omega_{ji} (1 - 2\gamma\eta)$$

For hidden layer,

weight update,

$$\omega_{ji} \leftarrow \eta \delta_{kd} x_j + \omega_{ji} (1 - 2\gamma\eta)$$

Here, ~~δ_j~~ $\delta_{kd} = O_{kd}(1 - O_{kd}) \sum_{k \in \text{down stream}} \delta_k \omega_{kj}$

Scanned by CamScanner

The above equation thus shows that the update rule can be implemented by multiplying each weight by some constant before standard Gradient Descent Update.