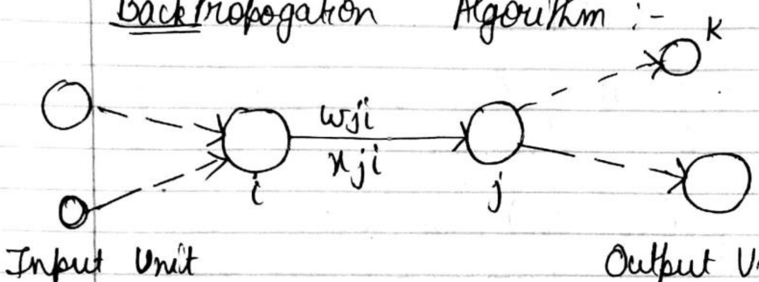
Ques 1.1

<u>Q-1.1</u>

a) Tanh activation function:

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

<u>BackPropogation</u> Algorithm :-



Consider above network with two intermediate nodes 'i' and 'j'.

$$\text{Error} = E_d(w) = \frac{1}{2} \Sigma (t_k - O_k)^2$$

Change in weight (Δw_{ji}) [w_{ji} is the weight from i^{th} node to j^{th} node].

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

$$\frac{\delta E_d}{\delta w_{ji}} = \frac{\delta E_d}{\delta net_j} \cdot \frac{\delta net_j}{\delta w_{ji}}$$

$$\frac{\delta net_j}{\delta w_{ji}} = x_{ji}$$

For $\frac{\delta E_d}{\delta net_j}$, there are 2 cases:-

(1) J is an Output Unit
(2) J is an hiddent unit.

(1) J is an Output unit.

$$\frac{\delta E_d}{\delta net_j} = \frac{\delta E_d}{\delta O_j} \frac{\delta O_j}{\delta net_j}$$

Error (E_d) is

$$E_d(W) = \frac{1}{2} \sum_{k \in output} (t_k - O_k)^2$$

Differentiate wrt O_j:

$$\frac{\delta E_d}{\delta O_j} = \frac{\delta}{\delta O_j} \left[\frac{1}{2} \sum_{k \in output} (t_k - O_k)^2 \right]$$

$$= \frac{\delta}{\delta O_j} \left[\frac{1}{2} (t_j - O_j)^2 \right] = -(t_j - O_j)$$

For activation function tanh.

$$\frac{\delta O_j}{\delta net_j} = \frac{\delta}{\delta net_j} \left[\frac{e^x - e^{-x}}{e^x + e^{-x}} \right]$$

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

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

$$\frac{\delta O_j}{\delta net_j} = 1 - \tan^2 h(x) = 1 - O_j^2$$

$$\frac{\delta E_d}{\delta net_j} = \frac{\delta E_d}{\delta O_j} \cdot \frac{\delta O_j}{\delta net_j}$$

$$= -(t_j - O_j)(1 - O_j^2)$$

$$\frac{\delta E_d}{\delta w_{ji}} = \left[-(t_j - O_j)(1 - O_j^2) \right] x_{ji}$$

$$\Delta w_{ji} = \eta (t_j - O_j)(1 - O_j^2) x_{ji}$$

If $\delta_j = (t_j - O_j)(1 - O_j^2)$.

$$\Delta w = \eta \delta_j x_{ji}$$

$$\frac{\delta E_d}{\delta net_j} = -\delta_j$$

<u>Case 2</u>: j is a hidden Layer.

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

$$= \sum -\delta_k \frac{\delta net_k}{\delta net_j}$$

$$= \sum -\delta_k \frac{\delta net_k}{\delta O_j} \frac{\delta O_j}{\delta net_j}$$

$$= \sum -\delta_k W_{kj} (1 - O_j^2).$$

$$\delta_j = (1-O_j)^2 \sum \delta_k W_{kj}$$

$$\Delta w_{ji} = \eta \frac{\delta E_d}{\delta w_{ji}} = \eta \delta_j X_{ji}$$

Relu Activation function

$$Relu(x) = \max(0, x).$$

$$\frac{\delta E_d}{\delta w_{ji}} = \frac{\delta E_d}{\delta net_j} \cdot \frac{\delta net_j}{\delta w_{ji}}$$

Case 1: j is an output Unit.

$$\frac{\delta E_d}{\delta net_j} = \frac{\delta E_d}{\delta O_j} \frac{\delta O_j}{\delta w_{ji}}$$

Differentiate wrt O_j;

$$\frac{\delta E_d}{\delta O_j} = -(t_j - O_j)$$

$$\frac{\delta O_i}{\delta net_j} = \frac{\delta(\max(0, x))}{\delta net_j}$$

$$= \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x > 0 \end{cases}$$

at $x = 0$ (function is not defined).

$$\frac{\delta E_d}{\delta w_{ji}} = \left[-(t_j - o_j) \right] x_{ji} \qquad \text{for } x > 0$$

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

Subsitute $t_j - o_j = \delta_j$

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

$$\frac{\delta E_d}{\delta net_j} = -\delta_j$$

Case 2 : j is a hidden unit.

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

$$= \sum_{k \in downstream(j)} -\delta_k \frac{\delta net_k}{\delta net_j}$$

$$= \sum_k -\delta_k \cdot \frac{\delta net_k}{\delta o_j} \frac{\delta o_j}{\delta net_j}$$

$$= \sum_k -\delta_k \frac{\delta net_k}{\delta o_j} \frac{\delta o_j}{\delta net_j}$$

$$= \sum -\delta_k \frac{\delta net_k}{\delta o_j} \qquad \text{for } x > 0$$

$$= -\sum_k \delta_k w_{kj}$$

$$\Delta w_{ji} = \eta \, \delta_j \, x_{ji} \qquad \eta > 0.$$

Summarize

$$\frac{\delta E_d}{\delta net_j} = \begin{cases} -\sum_k \delta_k w_{kj} & \text{for } o_j > 0 \\ 0 & o_j < 0 \end{cases}$$

Que 1.2

1.2 Gradient Descent

Single unit neuron with output O, can be defined as:

$O = w_0 + w_1(x_1 + x^2) \text{ --- } + w_n(x_n + x_n^2)$

where $x_1, x_2 \cdots x_n$ are inputs,

$w_1, w_2 \cdots w_n$ are the corresponding weights.

and w_0 is the bias weight.

Gradient :

$$\Delta E[\vec{w}] = \left[\frac{\delta E}{\delta w_0}, \frac{\delta E}{\delta w_1}, \cdots \frac{\delta E}{\delta w_n}\right]$$

Training Rule,

$$\Delta \vec{w} = -\eta \, \Delta E[\vec{w}]$$

$$\Delta \vec{W} = -\eta \frac{\delta E}{\delta w_i}$$

$$\frac{\delta E}{\delta w_i} = \frac{\delta}{\delta w_i}\left[\frac{1}{2d} \sum (td - od)^2\right]$$

$$= \frac{1}{2} \sum_d \frac{\delta}{\delta w_i} (td - od)^2$$

$$= \frac{1}{2} \sum_d 2(td - od) \cdot \frac{\delta}{\delta w_i} (td - od)$$

$$= \sum_d (td - od) \cdot \frac{\delta}{\delta w_i} (td - (w_0 + w_1(x_1 + x_1^2) + \text{---} \; w_n(x_n + x_n^2)))$$

$$= \sum_d (t_d - o_d) \; - \frac{\delta}{\delta w_i} [\vec{w}_d \; (x_d + x_d^2)$$

$$\frac{\delta E}{\delta w_i} = -\sum_d (t_d - o_d) \cdot (x_d + x^2 d)$$

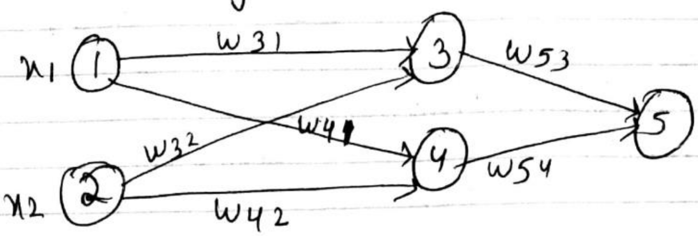
$$\Delta w = -\eta \frac{\delta E}{\delta w_i}$$

$$\Delta w = - \left[-\eta \sum_d (t_d - o_d) \; (x_d + x^2 d) \right]$$

$$\Delta w = \eta \sum_d (t_d - o_d) \; (x_d + x^2 d)$$

Ques 1.3

1.3 Comparing activation functions :-



Node	Net	~~Output~~	Output
1	x_1		
2	x_2		
3	$net3 = W_{31}x_1 + W_{32}x_2$		$x_3 = \eta(net\ 3)$
4	$net4 = W_{41}x_1 + W_{42}x_2$		$x_4 = \eta(net\ 4)$
5	$net5 = W_{53}x_3 + W_{54}x_4$		$x_5 = \eta(net5)$

$$x_5 = Y_5 = \eta(net5)$$
$$= \eta(W_{53}x_3 + W_{54}x_4).$$

$$Y_5 = h(W_{53}\,\eta(W_{31} + W_{32}x_2) + W_{54}\cdot\eta(W_{41}x_1 + W_{42}x_2))$$

(b) Input Layer to hidden Layer

Input 1 $= w^{(1)} x$

Output from hidden Layer $\rightarrow \eta(w^{(1)} x)$

Input 2 $= w^{(2)} \cdot \eta(w^{(1)} \cdot x)$

Output $y_5 = \eta(w^{(2)} \cdot \eta(w^{(1)} x))$

$$\Rightarrow \eta\left(\begin{bmatrix} w_{53} & w_{54} \end{bmatrix} \cdot \eta\left(\begin{bmatrix} w_{31} & w_{32} \\ w_{41} & w_{42} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right)\right)$$

(c) Using Activation function

$$hs(x) = \frac{1}{1-e^{-x}} \quad \text{(Sigmoid)}$$

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

$$\Rightarrow \quad hs(2x) = \frac{e^{2x}}{1+e^{2x}}$$

$$h_T(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad \text{(tanh)}$$

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

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

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

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

~~2hs(2x)~~ from (1) & (2)

$$h_T(x) = 2hs(2x) - 1$$

The output of tanh(x) differs only by a linear transformation (2x) and a constant (-1).

Ques 1.4

1.4 $\mathcal{E}(\vec{w}) = \frac{1}{2} \sum_{d \in D} \sum_{k \in outputs} (t_{kd} - O_{kd})^2 + \gamma \sum_{ij} w_{ji}^2$

$\Rightarrow \frac{\delta E_d}{\delta w_{ji}} = \frac{\delta E_d}{\delta net_j} \cdot \frac{\delta net_j}{\delta w_{ji}} + 2\gamma w_{ji}$

$= \frac{\delta E_d}{\delta net_j} x_{ji} + 2\gamma w_{ji}$

$= \frac{\delta E_d}{\delta O_j} \cdot \frac{\delta O_j}{\delta net_j} \cdot x_{ji} + 2\gamma w_{ji}$

$\frac{\delta E_d}{\delta O_j} \frac{\delta O_j}{\delta net_j} = -(t_j - O_j)(1-O_j)O_j$

$\frac{\delta E_d}{\delta w_{ji}} = -(t_j - O_j)(1-O_j)O_j x_{ji} + 2\gamma w_{ji}$

$w_{ji}\ new = w_{ji} + \eta (t_j - O_j) O_j (1-O_j) x_{ji} - 2\eta\gamma w_{ji}$

$= (1-2\eta\gamma) w_{ji} + \eta (t_j - O_j) O_j (1-O_j) x_{ji}$

$= (1-2\eta\gamma) w_{ji} + \eta \delta_j x_{ji}$ (for output Layer)

For hidden Layer

$w_{ji}\ new = w_{ji}(1-2\eta\gamma) + \eta \delta_j x_{ji}$

here $\delta_j = O_j(1-O_j)$

From the above, it can be interpreted that each Rule can be implemented by Multiplying $(1-2\eta\gamma) \rightarrow$ weight update before Gradient descent update.