Example 01:

$\boxed{\text{Forward Propagation}}$

Input, $x_1 = 1$, $x_2 = 1$

Hidden Layer, $I_3 = (0.5 \times 1) + (-0.3 \times 1) + 0.6$

$= 0.8$

$I_4 = \sum w_i x_i + b$

$= (0.2 \times 1) + (0.5 \times 1) + (-0.4)$

$= 0.3$

Now, $O_3 = \frac{1}{1+e^{-0.8}} = 0.69$

$O_4 = \frac{1}{1+e^{-0.3}} = 0.5744$

Output Layer, $I_5 = (O_3 \times 0.1) + (O_4 \times 0.3) + b_5$

$= (0.69 \times 0.1) + (0.5744 \times 0.3) + 0.8$

$= 1.04132$

$O_5 = \frac{1}{1+e^{-1.04132}} = 0.7391$

$\therefore$ Error $=$ target $- O_5$

$= 0 - 0.7391$

$= -0.7391$

Back Propagation:

Output Layer Error, $E_5 = O_5 \times (1 - O_5) \times (\text{target} - O_5)$

$= 0.7391 \times (1 - 0.7391) \times (0 - 0.7391)$

$= -0.14252$

Hidden Layer Error, $E_3 = O_3 \times (1 - O_3) \times (E_5 \times W_{35})$

$= 0.69 \times (1 - 0.69) \times (-0.14252 \times 0.1)$

$= -0.00304$

$E_4 = O_4 \times (1 - O_4)(E_5 \times W_{45})$

$= 0.5744 \times (1 - 0.5744)(-0.14252 \times 0.3)$

$= -0.01045$

Update weight,

$W(\text{new}) = W(\text{old}) + \Delta W_{ij}$

$= W(\text{old}) + \eta \times \delta_j \times O_j$

Learning rate error term output

$W_{13}(\text{new}) = 0.5 + (0.5 \times x_1 \times E_3)$

$= 0.5 + (0.5 \times 1 \times -0.00304)$

$= 0.49848$

$W_{14}(\text{new}) = 0.2 + (0.5 \times x_1 \times E_4)$

$= 0.2 + (0.5 \times 1 \times -0.01045)$

$= 0.194775$

W_{23} (new) $= -0.3 + (0.5 \times \overset{x_2}{1} \times -0.00304) = -0.30152$

W_{24} (new) $= 0.5 + (0.5 \times \overset{x_2}{1} \times -0.01045) = 0.494775$

W_{35} (new) $= 0.1 + (0.5 \times 0.69 \times -0.14252) = 0.05083$

W_{45} (new) $= 0.3 + (0.5 \times 0.5744 \times -0.14252) = 0.259$

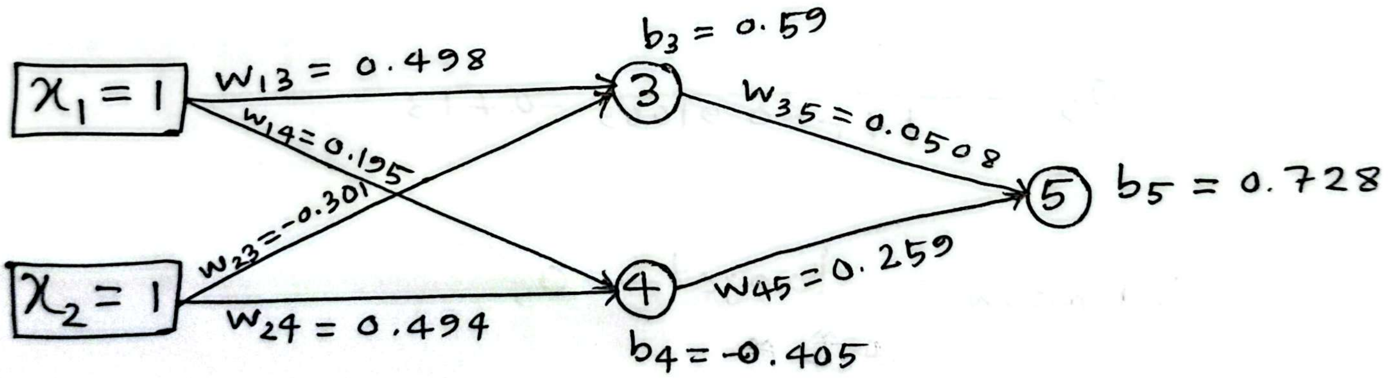
Update the bias $\quad b_j(\text{new}) = b_j(\text{old}) + \eta \times \delta_j$

b_3 (new) $= 0.6 + (0.5 \times E_3)$

$= 0.6 + (0.5 \times -0.00304) = 0.59848$

b_4 (new) $= -0.4 + (0.5 \times \underset{E_4}{-0.01045}) = -0.4052$

b_5 (new) $= 0.8 + (0.5 \times \underset{E_5}{-0.14252}) = 0.72874$



Forward Propagation

Hidden Layer, $I_3' = (1 \times 0.498) + (1 \times -0.301) + 0.59$
$= 0.787$

$I_4' = (1 \times 0.195) + (1 \times 0.494) + (-0.405)$
$= 0.284$

$$O_3' = \frac{1}{1+e^{-0.787}} = 0.687$$

$$O_4' = \frac{1}{1+e^{-0.284}} = 0.5705$$

Output Layer, $I_5' = (0.687 \times 0.0508) + (0.5705 \times 0.259) + 0.728$
$= 0.91065$

$$O_5' = \frac{1}{1+e^{-0.91065}} = 0.713$$

$\therefore$ Error $=$ target $- O_5'$
$= 0 - 0.713$
$= -0.713$

$\therefore$ Error is minimizing gradually.

Example 02:

Forward propagation

Input Layer, $x_1 = 0.35$, $x_2 = 0.9$

Hidden Layer, $I_3 = (0.35 \times 0.1) + (0.9 \times 0.8)$
$= 0.755$

$I_4 = (0.35 \times 0.4) + (0.9 \times 0.6)$
$= 0.68$

$$O_3 = \frac{1}{1+e^{-0.755}} = 0.680$$

$$O_4 = \frac{1}{1+e^{-0.68}} = 0.663$$

Output Layer,

$I_5 = (0.3 \times 0.68) + (0.9 \times 0.663) = 0.8007$

$$O_5 = \frac{1}{1+e^{-0.8007}} = 0.69$$

Error = target $- O_5$
$= 0.5 - 0.69$
$= -0.19$

Back Propagation

Output Layer Error

$$E_5 = O_5 \times (1 - O_5) \times (target - O_5)$$

$$= 0.69 \times (1 - 0.69) \times (0.5 - 0.69)$$

$$= -0.0406$$

Hidden Layer Error

$$E_3 = O_3 \times (1 - O_3) \times (E_5 \times W_{35})$$

$$= 0.68 \times (1 - 0.68) \times (-0.0406 \times 0.3)$$

$$= -0.00265$$

$$E_4 = O_4 \times (1 - O_4) \times (E_5 \times W_{45})$$

$$= 0.663 \times (1 - 0.663) \times (-0.0406 \times 0.9)$$

$$= -0.00816$$

Update the weights,

$$W_{13}(new) = 0.1 + (\overset{\eta}{1} \times \overset{input}{0.35} \times \overset{E_3}{-0.00265}) = 0.09907$$

$$W_{14}(new) = 0.4 + (1 \times \overset{input}{0.35} \times \overset{E_4}{-0.00816}) = 0.3971$$

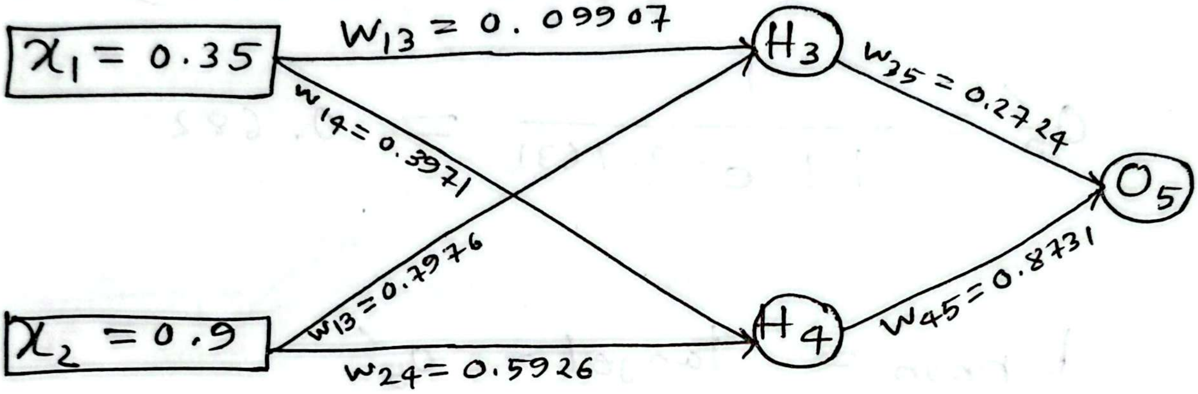
$$W_{23}(new) = 0.8 + (1 \times 0.9 \times -0.00265) = 0.7976$$

$$W_{24}(new) = 0.6 + (1 \times 0.9 \times -0.00816) = 0.5926$$

$W_{35}(new) = 0.3 + (1 \times 0.68 \times -0.0406) = 0.2724$

$W_{45}(new) = 0.9 + (1 \times 0.663 \times -0.0406) = 0.8731$

After updating weights:



$\boxed{\text{Forward Propagation}}$

Input Layer, $x_1 = 0.35$, $x_2 = 0.9$

Hidden Layer, $I_3' = (0.35 \times 0.09907) + (0.9 \times 0.7976) = 0.7525$

$I_4' = (0.35 \times 0.3971) + (0.9 \times 0.5926)$
$= 0.6723$

$$O_3' = \frac{1}{1+e^{-0.7525}} = 0.6797$$

$$O_4' = \frac{1}{1+e^{-0.6723}} = 0.6620$$

Output Layer, $I_5' = (0.6797 \times 0.2724) + (0.6620 \times 0.8731) = 0.7631$

$$O_5' = \frac{1}{1+e^{-0.7631}} = 0.682$$

Error = target − O_5'

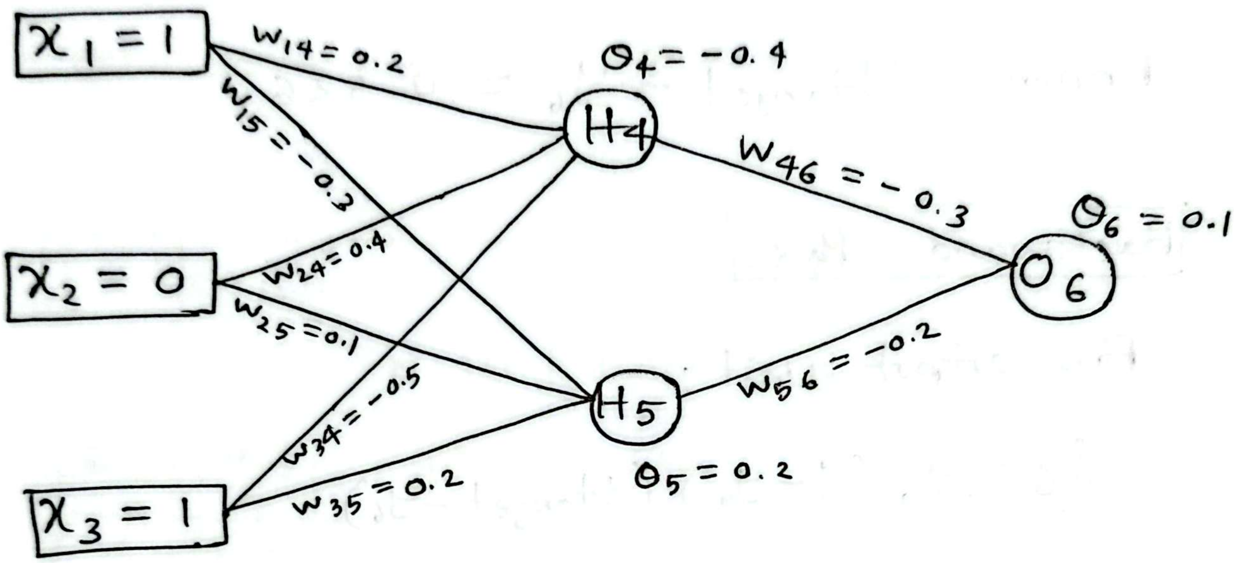
= 0.5 − 0.6820

= −0.182

∴ Error is minimizing.

Multilayer Preceptron Network

Example 03:



Actual Output = 1

Assume, $y = 1$ and Learning rate 0.9.

$\boxed{\text{Forward Pass}}$

$$a_4 = W_{14} \times x_1 + W_{24} \times x_2 + W_{34} \times x_3 + \theta_4$$

$$= 0.2 \times 1 + 0.4 \times 0 + (-0.5 \times 1) + (-0.4) = -0.7$$

$$O(H_4) = \frac{1}{1+e^{0.7}} = 0.332$$

$$a_5 = (-0.3 \times 1) + (0.1 \times 0) + (0.2 \times 1) + 0.2 = 0.1$$

$$O(H_5) = \frac{1}{1+e^{-0.1}} = 0.525$$

$a_6 = (-0.3 \times 0.332) + (0.2 \times 0.525) + 0.1 = -0.105$

$o(o_6) = \frac{1}{1+e^{0.105}} = 0.474$

Error = $y_{target} - y_6 = 0.526$

Backward Pass

For output unit:

$\delta_6 = y_6(1-y_6)(y_{target} - y_6)$

$= 0.474 \times (1-0.474)(1-0.474)$

$= 0.1311$

For Hidden unit:

$\delta_5 = 0.525 \times (1-0.525) \times (-0.2 \times 0.1311)$

$= -0.0065$

$\delta_4 = 0.332(1-0.332) \times (-0.3 \times 0.1331)$

$= -0.0087$

compute new weights,

$\Delta W_{46} = \eta \delta_6 y_4 = 0.9 \times 0.1311 \times 0.332$

$= 0.03917$

$W_{46}(new) = \Delta W_{46} + W_{46}(old) = 0.03917 + (-0.3)$
$= -0.261$

$\Delta W_{14} = 0.9 \times (-0.0087) \times 1 = -0.0078$

$W_{14}(new) = -0.0078 + 0.2 = 0.192$

Similarly,

i	j	w_{ij}	δ_i	x_i	η	Update w_{ij}
4	6	-0.3	0.1311	0.332	0.9	-0.261
5	6	-0.2	0.1311	0.525	0.9	-0.138
1	4	0.2	-0.0087	1	0.9	0.192
1	5	-0.3	-0.0065	1	0.9	-0.306
2	4	0.4	-0.0087	0	0.9	0.4
2	5	0.1	-0.0065	0	0.9	0.1
3	4	-0.5	-0.0087	1	0.9	-0.508
3	5	0.2	-0.0065	1	0.9	0.194

Similarly update bias,

θ_j	Previous θ_j	δ_j	η	Update θ_j
θ_6	0.1	0.1311	0.9	0.218
θ_5	0.2	−0.0065	0.9	0.194
θ_4	−0.4	−0.0087	0.9	−0.408

$\boxed{\text{Forward Pass}}$

$$a_4 = 0.192 \times 0.4 \times 0 + (-0.508 \times 1) + (-0.408)$$
$$= -0.724$$

$$O(H_4) = \frac{1}{1+e^{0.724}} = 0.327$$

$$a_5 = -0.306 \times 1 + 0.1 \times 0 + 0.194 \times 1 + 0.194$$
$$= 0.082$$

$$O(H_5) = \frac{1}{1+e^{-0.082}} = 0.52$$

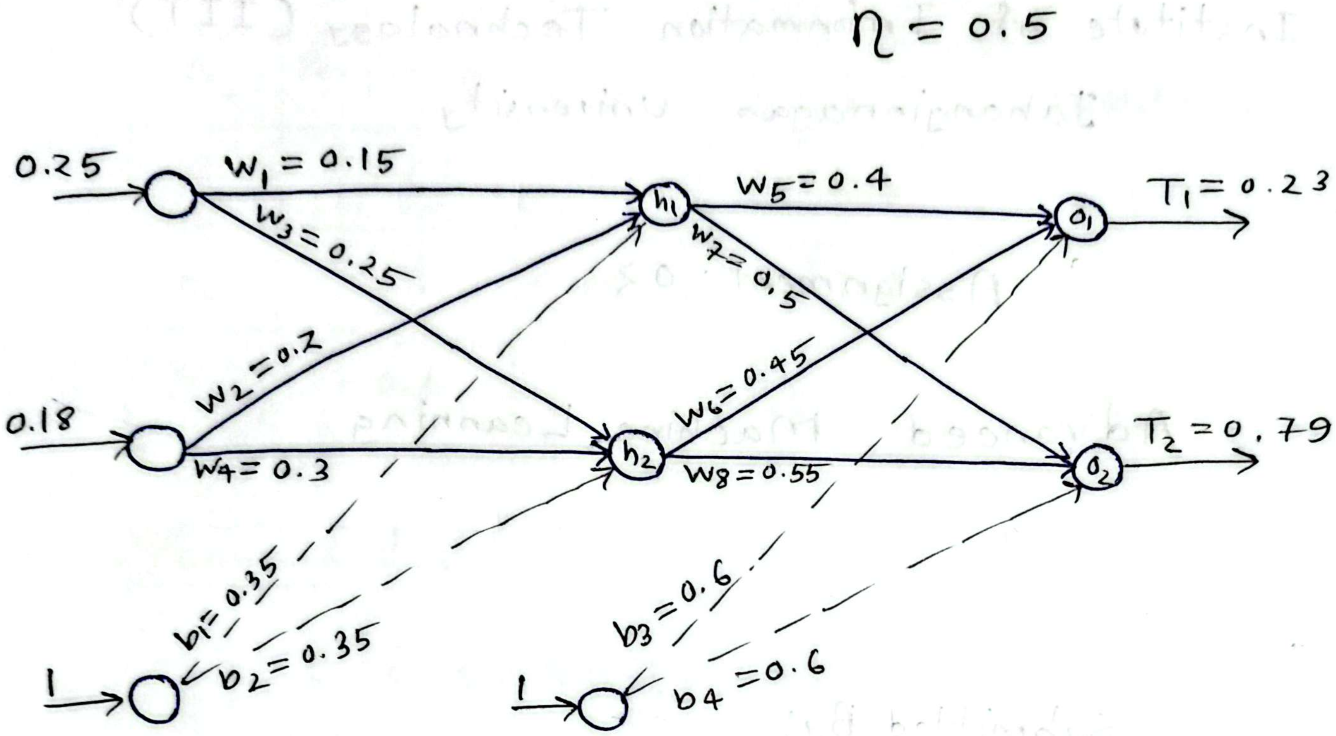
$$a_6 = -0.261 \times 0.327 + (-0.138 \times 0.520) + 0.218 = 0.6$$

$$O(H_6) = \frac{1}{1+e^{-0.061}} = 0.515$$

$$\text{Error} = y_{target} - y_6 = 0.485$$

Error is minimizing.

9.8.2 Example



Forward

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

$$out_{h_1} = f(w_1 \times i_1 + w_2 i_2 + b_1 \times 1) = f(0.15 \times 0.25 + 0.2 \times 0.18 + 0.35)$$
$$= f(0.4235) = 0.6043$$

$$out\, h_2 = f(0.25 \times 0.25 + 0.3 \times 0.18 + 0.35 \times 1) = f(0.4665)$$
$$= 0.6146$$

$$out_{o_1} = f(w_5 \times out\,h_1 + w_6 \times out\,h_2 + b_3 \times 1)$$
$$= f(0.4 \times 0.6043 + 0.45 \times 0.6146 + 0.6)$$
$$= f(1.11829) = 0.7537$$

$$out_{o_2} = f(0.5 \times 0.6043 + 0.55 \times 0.6146 + 0.6)$$
$$= f(1.24018) = 0.7756$$

$$Error = \frac{1}{2}\left[(T_1 - out_{o_1})^2 + (T_2 - out_{o_2})^2\right]$$

$$= \frac{1}{2}\left[(0.23 - 0.7537)^2 + (0.79 - 0.7756)^2\right]$$

$$= 0.1372$$

Backward

computing adjusted weights o_1 and o_2

$$\delta_{o_1} = (T_1 - out_{o_1})\ out_{o_1}\ (1 - out_{o_1})$$
$$= (0.23 - 0.7537)\ 0.7537\ (1 - 0.7537)$$
$$= -0.0972$$

$$W_5^+ = W_5 + \eta\ \delta_{o_1}\ out_{h_1}$$

$$= 0.4 + 0.5 \times (-0.0972)\ 0.6043$$
$$= 0.3706$$

$$W_6^+ = W_6 + \eta\, \delta_{o_1}\, out_{h_2}$$

$$= 0.45 + 0.5 \times (-0.0972) \times 0.6146$$

$$= 0.4201$$

$$b_3^+ = b_3 + \eta\, \delta_{o_1} \times 1$$

$$= 0.6 + 0.5 \times (-0.0972) \times 1$$

$$= 0.5514$$

$$\delta_{o_2} = (T_2 - out_{o_2})\, out_{o_2}\, (1 - out_{o_2})$$

$$= (0.79 - 0.7756)\, 0.7756\, (1 - 0.7756)$$

$$= 0.0025$$

$$W_7^+ = 0.5 + 0.5 \times (0.0025) \times 0.6043$$

$$= 0.5008$$

$$W_8^+ = 0.55 + 0.5 \times (0.0025) \times 0.6146$$

$$= 0.5508$$

$$b_4^+ = 0.6 + 0.5 \times (0.0025) \times 1$$

$$= 0.6013$$

Computing adjusted weights h_1 and h_2

$$\delta h_1 = (w_5 \times \delta o_1 + w_2\, \delta o_2) \times out\,h_1 \times (1 - out\,h_1)$$

$$= [0.4 \times (-0.0972) + (0.5 \times 0.0025)] \times 0.6043\,(1 - 0.6043)$$

$$= -0.00899$$

$$w_1^{+} = 0.15 + \eta \times (-0.00899) \times 0.25$$

$$= -1.686 \times 10^{-4}$$

$$w_2^{+} = 0.2 + 0.5 \times (-0.00899) \times 0.18$$

$$= 0.1992$$

$$b_1^{+} = 0.35 + 0.5 \times (-0.00899) \times 1$$

$$= 0.3455$$

$$\delta h_2 = (w_6 \times \delta o_1 + w_8 \times \delta o_2)\, out_{h_2} \times (1 - out_{h_2})$$

$$= (0.45 \times (-0.0972) + 0.55 \times 0.0025) \times$$
$$0.6146 \times (1 - 0.6146)$$

$$= -0.0101$$

$$w_3^+ = 0.25 + 0.5 \times (-0.0101) \times 0.25$$
$$= 0.2487$$

$$w_4^+ = 0.3 + 0.5 \times (-0.0101) \times 0.18$$
$$= 0.2991$$

$$b_2^+ = 0.35 + 0.5 \times (-0.0101) \times 1$$
$$= 0.3449$$

Now, we get,

$w_1 = w_1^+$, $w_2 = w_2^+$, $w_3 = w_3^+$, $w_4 = w_4^+$

$w_5 = w_5^+$, $w_6 = w_6^+$, $w_7 = w_7^+$, $w_8 = w_8^+$

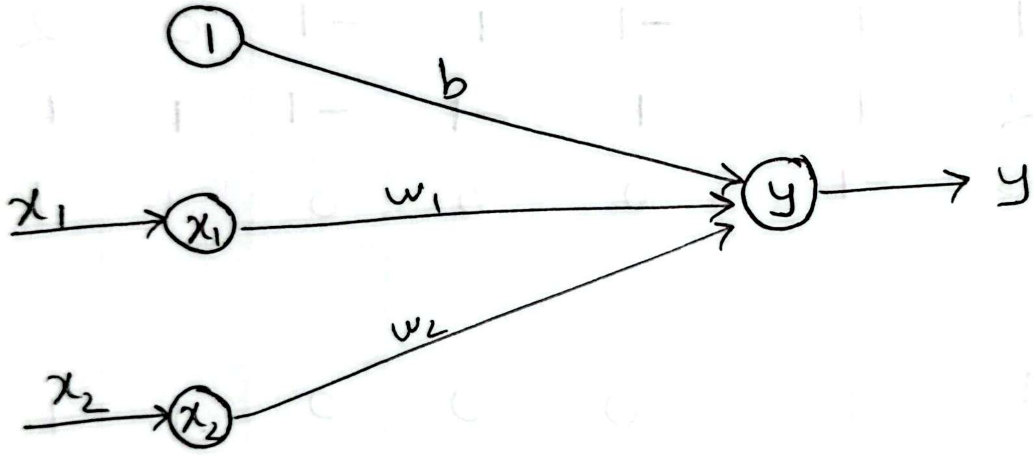
$b_1 = b_1^+$, $b_2 = b_2^+$, $b_3 = b_3^+$, $b_4 = b_4^+$

repeat.

The process is repeated until the root mean square of output errors is minimized.

Perceptron Learning Rule [AND Function]

x_1	x_2	t
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1



initial weights = 0

$b = 0$

Learning rate $\alpha = 1$

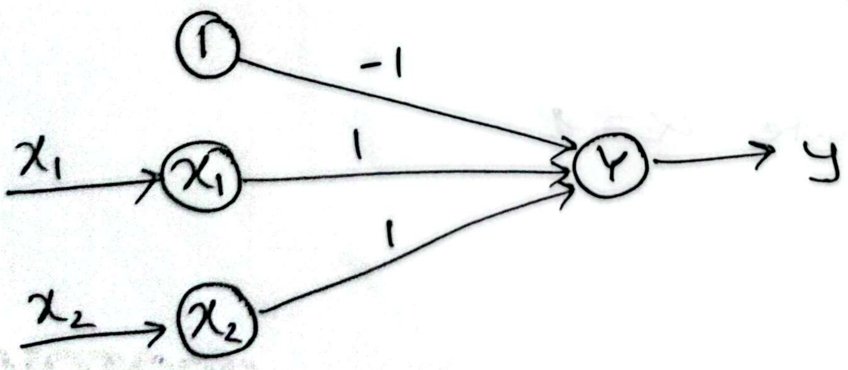
$\boxed{\alpha = 1}$

$$y_{in} = b + \sum_{i}^{n} w_i x_i$$

$$y = f(y_{in}) = \begin{cases} 1 & y_{in} > 1 \\ 0 & y_{in} = 0 \\ -1 & y_{in} < 0 \end{cases}$$

$$\Delta w_1 = \alpha t x_1$$
$$\Delta w_2 = \alpha t x_2$$
$$\Delta b = \alpha t$$

Input x_1	Input x_2	Target t	y_{in}	Calculated output y	Weight Changes Δw_1	Δw_2	Δb	Weights w_1	w_2	b
								0	0	0
Epoch 1										
1	1	1	0	0	1	1	1	1	1	1
1	-1	-1	1	1	-1	1	-1	0	2	0
-1	1	-1	2	1	1	-1	-1	1	1	-1
-1	-1	-1	-3	-1	0	0	0	1	1	-1
Epoch 2										
1	1	1	1	1	0	0	0	1	1	-1
1	-1	-1	-1	-1	0	0	0	1	1	-1
-1	1	-1	-1	-1	0	0	0	1	1	-1
-1	-1	-1	-3	-1	0	0	0	1	1	-1



Back Propagation Algorithm Derivation

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

E_d error on training example d.

$$E_d(\vec{w}) = \frac{1}{2} \sum_{k \in outputs} (t_k - o_k)^2$$

x_{ji} = the ith input to unit j

w_{ji} = the weight associated with the ith input to unit j

$net_j = \sum_i x_{ji} w_{ji}$

o_j = the output computed by unit j

t_j = the target output for unit j

σ = Sigmoid Function

Chain rule,

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

$$net_j = \sum_i w_{ji} x_{ji}$$

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

$$= \frac{\partial E_d}{\partial net_j} \cdot x_{ji}$$

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

Our remaining task is to derive a convenient expression $\frac{\partial E_d}{\partial net_j}$,

We consider two cases in turn,

Case 1: where unit j is an output of the network.

Case 2: where unit j is an internal unit of the network.

$\boxed{\text{Case 1}}$: Training Rule for Output Unit Weights

$$\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial o_j} \cdot \frac{\partial o_j}{\partial net_j}$$

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

$$= -(t_j - o_j)$$

$$\frac{\partial o_j}{\partial net_j} = \frac{\partial\, \sigma(net_j)}{\partial(net_j)} = \sigma(net_j)\left[1 - \sigma(net_j)\right]$$

$$= o_j(1 - o_j)$$

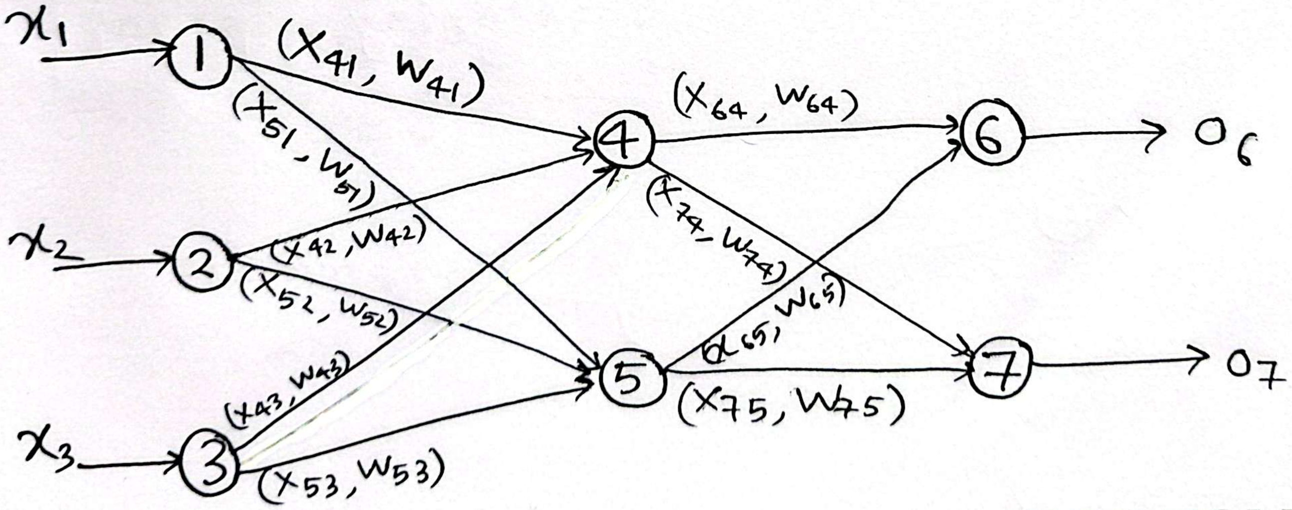
$$\therefore \frac{\partial E_d}{\partial net_j} = -(t_j - o_j)\; o_j(1 - o_j)$$

Now,

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

$$= -\eta\; \underline{-(t_j - o_j)\, o_j\, (1 - o_j)}\; x_{ji}$$

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



Case 2: Trainig Rule for Hidden Unit Weights

$$\frac{\partial E_d}{\partial net_j} = \sum_{K \in Downstream(j)} \frac{\partial E_d}{\partial net_K} \cdot \frac{\partial net_K}{\partial net_j}$$

$$= \sum_{K \in Downstream(j)} -\delta_K \frac{\partial net_K}{\partial o_j} \frac{\partial o_j}{\partial net_j}$$

$$= \sum_{K \in Downstream(j)} -\delta_K \, w_{Kj} \, o_j(1-o_j)$$

$$\Delta w_{ji} = -\eta \, \underbrace{o_j(1-o_j) \sum_{K \in Downstream(j)} -\delta_K w_{Kj}} \, x_{ji}$$

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