Perceptron learning rule numericals

① Use perceptron learning rule to the train the network. The set of input training vectors are as follows:

$$X_1 = \begin{bmatrix} +1 \\ -2 \\ 0 \\ -1 \end{bmatrix} \quad X_2 = \begin{bmatrix} 0 \\ 1.5 \\ -0.5 \\ -1 \end{bmatrix} \quad X_3 = \begin{bmatrix} -1 \\ 1 \\ 0.5 \\ -1 \end{bmatrix}$$

and the initial weight vector W_1 is,

$$W_1 = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0.5 \end{bmatrix}.$$ The learning constant $c = 0.1$.

The desired response for X_1, X_2 and X_3 are $d_1 = -1$, $d_2 = -1$ and $d_3 = 1$ respectively. Calculate the weights after one complete cycle.

Soln:-

for perceptron learning rule,

$$net_i = w_i^t X$$

$$O_i = \text{Sign}(net_i)$$

$$\Delta W_i = c(d - O_i)X$$

Step 1: Taking the first training pair $X = X_1$ and $d = d_1$

$$X = \begin{bmatrix} 1 \\ -2 \\ 0 \\ 1 \end{bmatrix} \quad d = -1$$

$$net_1 = w_1^t X = \begin{bmatrix} 1 & -1 & 0 & 0.5 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \\ 0 \\ -1 \end{bmatrix} = 2.5.$$

$$\text{Sign}(net_1) = \text{Sign}(2.5) = 1$$

②

$$\Delta W_1 = c(d-o_1)\cancel{x} = 0.1(-1-1)\begin{bmatrix}1\\-2\\0\\1\end{bmatrix} = \begin{bmatrix}-0.2\\0.4\\0\\0.2\end{bmatrix}$$

$$W_2 = W_1 + \Delta W_1 = \begin{bmatrix}1\\-1\\0\\0.5\end{bmatrix} + \begin{bmatrix}-0.2\\0.4\\0\\0.2\end{bmatrix} = \begin{bmatrix}0.8\\-0.6\\0\\0.7\end{bmatrix}$$

Similarly:

Step 2: Taking Second training pair.

Set $X = X_2$ $d = d_2$.

$$X = \begin{bmatrix}0\\1.5\\-0.5\\-1\end{bmatrix} \qquad d = 1$$

$$net_2 = W_2^t X = \begin{bmatrix}0.8 & -0.6 & 0 & 0.7\end{bmatrix}\begin{bmatrix}0\\1.5\\-0.5\\-1\end{bmatrix} = -1.6$$

$$O_2 = Sign(net_2) = Sign(-1.6)$$
$$= -1$$

$d = -1$ $O_2 = -1$

So $(d - O_2) = 0$

Hence $\Delta W_2 = 0$. Thus there is no change in weight

$$\therefore W_3 = W_2 = \begin{bmatrix}0.8\\-0.6\\0\\0.7\end{bmatrix}$$

Step 3: Take the third training pair

Set $X = X_3$ $d = d_3$.

$$X = \begin{bmatrix}-1\\1\\0.5\\-1\end{bmatrix} \qquad d = 1$$

$$net_3 = W_3^t X = \begin{bmatrix} 0.8 & -0.6 & 0 & 0.7 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 0.5 \\ -1 \end{bmatrix} = -2.1$$

$$O_3 = \text{Sign}(net_3) = \text{Sign}(-2.1) = -1$$

$$d = 1 \qquad O_3 = -1$$

$$\therefore \Delta W_3 = c(d - O_3) = 0.1\,(1-(-1)) \begin{bmatrix} -1 \\ 1 \\ 0.5 \\ -1 \end{bmatrix}$$

$$= \begin{bmatrix} -0.2 \\ 0.2 \\ 0.1 \\ -0.2 \end{bmatrix}$$

$$W_4 = W_3 + \Delta W_3 = \begin{bmatrix} 0.8 \\ -0.6 \\ 0 \\ 0.7 \end{bmatrix} + \begin{bmatrix} -0.2 \\ 0.2 \\ 0.1 \\ -0.2 \end{bmatrix} = \begin{bmatrix} 0.6 \\ -0.4 \\ 0.1 \\ 0.5 \end{bmatrix}$$

(2) Perform two training steps of the neural network using delta learning rule $\lambda = 1$ $c = 0.25$. Train the network using following data pairs.

$$X_1 = \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix} \quad d_1 = -1 \qquad X_2 = \begin{bmatrix} 1 \\ -2 \\ -1 \end{bmatrix} \quad d_2 = 1.$$

The initial weights are $w_1 = [1\ 0\ 1]^t$.

Soln:- for delta learning rule.

$$net_i = w_i^t x.$$

$$O_i = f(net_i) = \frac{2}{1 + e^{-\lambda net_i}} - 1$$

$$f'(net_i) = \frac{1}{2}(1 - O_i^2)$$

$$\Delta w_i = c(d - O_i)\left[\frac{1}{2}(1 - O_i)^2\right.$$

Step 1: 1st training pair $x = x_1 \quad d = d_1$

$$x = \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix} \quad d = -1$$

$$net_1 = w_1^t x = [1 \ 0 \ 1] \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix} = 2 - 1 = 1.$$

$$o_1 = \frac{2}{1 + e^{-1}} - 1 = 0.463$$

$$f'(net_1) = \frac{1}{2}(1 - (0.463)^2) = 0.393$$

$$\Delta w_1 = 0.25(-1 - 0.463)\, 0.393 \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix}$$

$$= -0.1437 \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} -0.287 \\ 0 \\ 0.1437 \end{bmatrix}$$

$$w_2 = w_1 + \Delta w_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -0.287 \\ 0 \\ 0.1437 \end{bmatrix} = \begin{bmatrix} 0.713 \\ 0 \\ 1.1437 \end{bmatrix}$$

Step 2: 2nd training pair $x = x_2 \quad d = d_2$

$$x = \begin{bmatrix} 1 \\ -2 \\ -1 \end{bmatrix} \quad d = 1$$

$$net_2 = w_2^t x = [0.713 \ 0 \ 1.1437] \begin{bmatrix} 1 \\ -2 \\ -1 \end{bmatrix}$$

$$= -0.4307$$

$$o_2 = f(net_2) = \frac{2}{1 + e^{0.4307}} - 1 = -0.2119$$

$$f'(net_2) = \frac{1}{2}(1 - (-0.2119)^2)$$

$$= \frac{1}{2}(1 - 0.0449) = 0.477$$

$$\Delta W_2 = (0.25)\ (1-(-0.2119)) \times (0.477) \times X_2$$

$$= (0.145) \begin{bmatrix} 1 \\ -2 \\ -1 \end{bmatrix} = \begin{bmatrix} 0.145 \\ -0.29 \\ -0.145 \end{bmatrix}$$

$$W_3 = W_2 + \Delta W_2 = \begin{bmatrix} 0.713 \\ 0 \\ 1.143 \end{bmatrix} + \begin{bmatrix} 0.145 \\ -0.29 \\ -0.145 \end{bmatrix}$$

$$= \begin{bmatrix} 0.858 \\ -0.29 \\ 0.998 \end{bmatrix}$$

3) Perform two training steps of widrow - Hoff learning rule

$$X_1 = \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix} \quad d_1 = -1 \qquad X_2 = \begin{bmatrix} 1 \\ -2 \\ -1 \end{bmatrix} \quad d_2 = 1$$

$$W_1 = [1\ 0\ 1]^t \qquad \lambda = 1 \qquad c = 0.25.$$

Soln:- <u>widrow Hoff learning</u>

$$O_i = W_i^t X \qquad \Delta W_i = c(d_i - o_i)x$$

$$O_1 = W_i^t X_1 = [1\ 0\ 1] \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix} = 1$$

$$\Delta W_1 = c\,(d_1 - o_1)\,X_1 = 0.25\ (-1-1) \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix}$$

$$= \begin{bmatrix} -1 \\ 0 \\ 0.5 \end{bmatrix}$$

$$W_2 = W_1 + \Delta W_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 \\ 0 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1.5 \end{bmatrix}$$

$$O_2 = W_2^t X_2 = \begin{bmatrix} 0 & 0 & 1.5 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \\ -1 \end{bmatrix} = -1.5$$ (6)

$$\Delta W_2 = c\,(d_2 - O_2)\,X_2 = 0.25 \times 2.5 \begin{bmatrix} 1 \\ -2 \\ -1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.625 \\ -1.25 \\ -0.625 \end{bmatrix}$$

$$\Delta W_3 = W_2 + \Delta W_2 = \begin{bmatrix} 0 \\ 0 \\ 1.5 \end{bmatrix} + \begin{bmatrix} 0.625 \\ -1.25 \\ -0.625 \end{bmatrix}$$

$$= \begin{bmatrix} 0.625 \\ -1.25 \\ 0.875 \end{bmatrix}$$

(4) Consider the given input pattern's P_1, P_2, P_3, P_4, P_5 & P_6 as well as initial weights W_1, W_2 & W_3.

$$P_1 = \begin{bmatrix} -0.1961 \\ 0.9806 \end{bmatrix} \quad P_2 = \begin{bmatrix} 0.1961 \\ 0.9806 \end{bmatrix} \quad P_3 = \begin{bmatrix} 0.9806 \\ 0.1966 \end{bmatrix}$$

$$P_4 = \begin{bmatrix} 0.9806 \\ -0.1961 \end{bmatrix} \quad P_5 = \begin{bmatrix} -0.5812 \\ -0.8173 \end{bmatrix} \quad P_6 = \begin{bmatrix} -0.8137 \\ -0.5812 \end{bmatrix}$$

$$W_1 = \begin{bmatrix} 0.7071 \\ -0.7071 \end{bmatrix} \quad W_2 = \begin{bmatrix} 0.7071 \\ 0.7071 \end{bmatrix} \quad W_3 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

Consider $\alpha = 0.5$, perform clustering using winner take all algorithm (Competitive learning) Inputs patterns presented in following sequence $P_4, P_3, P_2, P_1, P_6, P_5$

Soln:-

weight vector

$$w^t = \begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix} = \begin{bmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \\ -1 & 0 \end{bmatrix}$$

7

Step 1- Compete $w^t P_4$

$$= \begin{bmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0.9806 \\ -0.1961 \end{bmatrix}$$

$$= \begin{bmatrix} 0.832 \\ 0.5547 \\ -0.9806 \end{bmatrix}$$

W_1 win's So update W_1

$$W_1 \text{ new} = W_1 \text{ old} + \alpha (P_4 - W_1 \text{ old})$$

$$= \begin{bmatrix} 0.7071 \\ -0.7071 \end{bmatrix} + 0.5 \left(\begin{bmatrix} 0.9806 \\ -0.1961 \end{bmatrix} - \begin{bmatrix} 0.7071 \\ -0.7071 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 0.844 \\ -0.452 \end{bmatrix}$$

updated weights

$$w^t = \begin{bmatrix} 0.844 & -0.452 \\ 0.7071 & 0.7071 \\ -1 & 0 \end{bmatrix}$$

Step 2:- Compete $w^t P_3$

$$\cancel{W_2} \; w^t P_3 = \begin{bmatrix} 0.739 \\ 0.832 \\ -0.9806 \end{bmatrix}$$

W_2 win's So update only W_2

$$W_{2new} = W_{2old} + \alpha (P_3 - W_{2old})$$

$$= \begin{bmatrix} 0.844 \\ 0.452 \end{bmatrix}$$

Hence

$$wt = \begin{bmatrix} 0.844 & -0.452 \\ 0.844 & 0.452 \\ -1 & 0 \end{bmatrix}$$

Step 3: Compete $w^t P_2$

$$= \begin{bmatrix} -0.2777 \\ 0.6087 \\ -0.1961 \end{bmatrix}$$

$$W_{2new} = W_{2new} + \alpha (P_2 - W_{2old})$$

$$= \begin{bmatrix} 0.52 \\ 0.7163 \end{bmatrix}$$

$$wt = \begin{bmatrix} 0.844 & -0.452 \\ 0.52 & 0.7163 \\ -1 & 0 \end{bmatrix}$$

Step 4: Compete $\begin{bmatrix} -0.6087 \\ 0.6 \\ 0.1961 \end{bmatrix}$

again W_2 wins:

$$W_{2new} = \begin{bmatrix} 0.1619 \\ 0.8484 \end{bmatrix}$$

W_1 (0.844 −0.452)

(−0.740, ...) P5

⑨

updated weights are

$$w^t = \begin{bmatrix} 0.844 & -0.452 \\ 0.1619 & 0.8484 \\ -1 & 0 \end{bmatrix}$$

Step 5: Compete $w^t P_6$:

$$\text{compete} \begin{bmatrix} -0.424 \\ -0.624 \\ 0.8137 \end{bmatrix}$$

W_3 wins

$$W_{3new} = \begin{bmatrix} -0.9 \\ -0.3 \end{bmatrix}$$

$$w^t = \begin{bmatrix} 0.844 & -0.452 \\ 0.1619 & 0.844 \\ -0.9 & -0.3 \end{bmatrix}$$

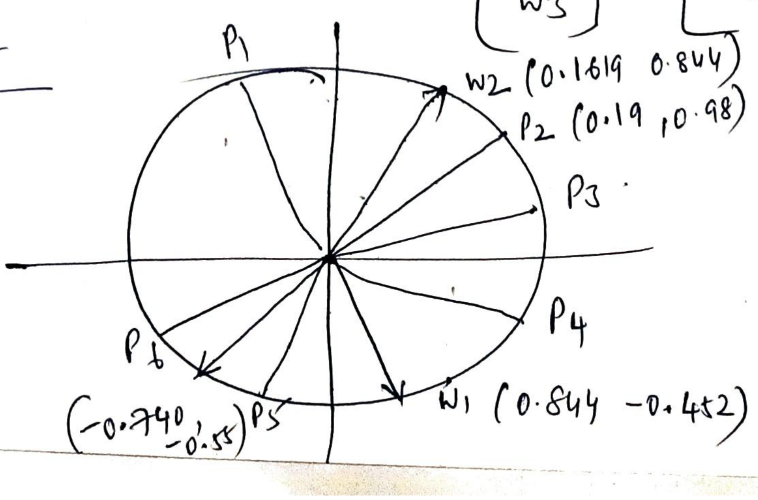
Step 6: - Complete $w^t P_5$

$$\text{compete} \begin{bmatrix} -0.1227 \\ -0.7808 \\ 0.7671 \end{bmatrix}$$ W3 wins

$$W_{3new} = \begin{bmatrix} -0.7406 \\ -0.5569 \end{bmatrix}$$

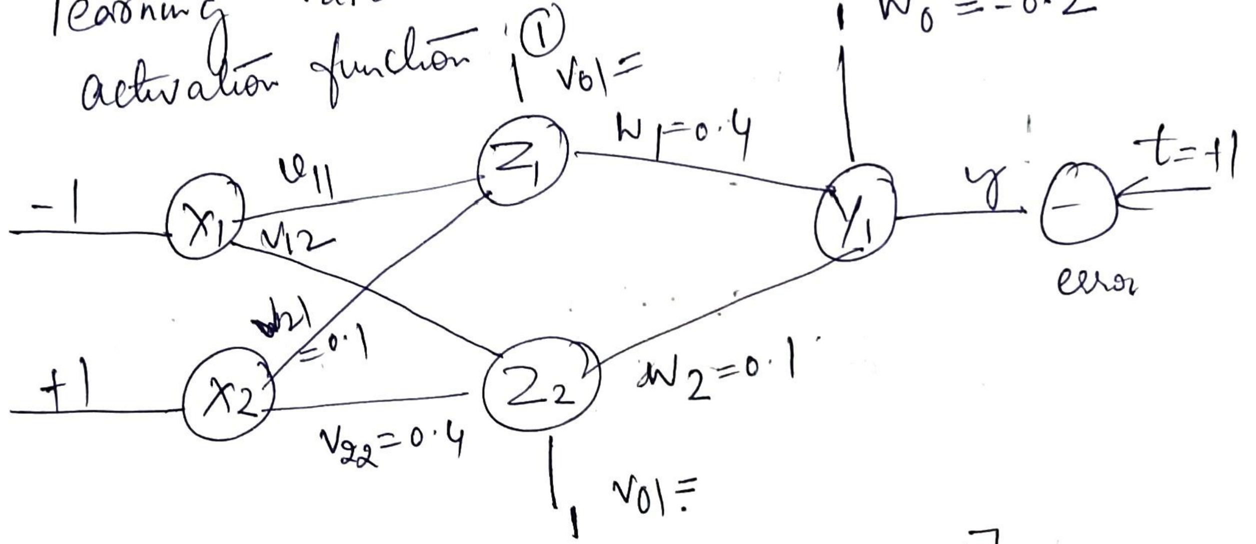
Hence final weights $$w^t = \begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix} = \begin{bmatrix} 0.844 & -0.452 \\ 0.1619 & 0.844 \\ -0.7406 & -0.5569 \end{bmatrix}$$

final wts



W_2 is rotated towards pattern P_1 P_2 ...
Similarly W_3 is rotated towards P_5 & P_6 forming Cluster Centers for respective clusters.

(5) Error Back propogation algorithm (EBPTA)
find the new weights using EBPTA ~~to find~~ ~~the new weights~~ for the following network. The network is presented using input pattern [-1 1] & target output 1. Use a learning rate $\alpha = 0.25$ & bipolar sigmoidal activation function



Soln:
$$[v_{01} \; v_{11} \; v_{12}] = [0.3 \;\; 0.6 \;\; -0.1]$$
$$[v_{02} \; v_{12} \; v_{21}] = [0.5, -0.3, 0.4]$$
$$[W_0 \; W_1 \; W_2] = [-0.2, 0.4, 0.1]$$

input $[x_1 \; x_2] = [-1 \;\; +1]$
target o/p $t = +1$ $\quad \alpha = 0.25$

Step 6 – Change in weight b/w hidden &

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

first derivative $f'(x) = \frac{1}{2}\left(1-f(x)^2\right)$

Step 1 – Calculate the hidden layer z_1 z_2

$Z_{in1} = 0.3\times1 + (-1\times0.6) + (1\times-0.1)$

$= 0.3 - 0.6 - 0.1 = -0.4$

$Z_{in2} = 0.5\times1 + (-1\times0.3) + (1\times0.4)$

$= 1.2$

$z_1 = f(z_{in1}) = \frac{1-e^{+0.4}}{1+e^{0.4}} = -0.1974$

$z_2 = f(z_{in2}) = \frac{1-e^{-1.2}}{1+e^{-1.2}} = 0.537$

Step 2 Calculate at output layer (y)

$y_{in} = z_1 w_1 + z_2 w_2 + 1\times w_0$

$= -0.1974\times0.4 + 0.537\times0.1 + 1\times-0.2$

$= -0.22526$

$y = f(y_{in}) = \frac{1-e^{+0.22526}}{1+e^{0.22526}} = -0.1122$

Step 3 - Compute the error b/w output & hidden layer (12)

$$\text{error}\ (\delta_{oh}) = (t-y)\, f'(y_{in})$$

$$\delta_{oh} = (1+0.1122)\ \frac{1}{2}\left(1-0.1122^2\right)$$

$$= \underline{0.5491}$$

Step 4: Change the weight b/w output layer & hidden layer

$$\Delta w_1 = \alpha\, \delta_{oh}\, z_1 = 0.25 \times 0.5491 \times (-0.1974)$$

$$\Delta w_1 = -0.0271$$

$$\Delta w_2 = \alpha\, \delta_{oh}\, z_2 = 0.25 \times 0.5491 \times (0.537)$$

$$\Delta w_2 = 0.0737$$

$$\Delta w_0 = \alpha\, \delta_{oh}\, 1 = 0.25 \times 0.5491$$

$$= 0.1373$$

Step 4: Compute error b/w hidden & i/p layer

$$\text{error}\ (\delta_{inz_1}) = \delta_{oh} \times w_1 \times f'(z_1)$$

$$= 0.5491 \times 0.4 \times \frac{1}{2}\left(1-z_1^2\right)$$

$$= \underline{0.1056}$$

$$\text{error}\ (\delta_{inz_2}) = \delta_{oh} \times w_2 \times f'(z_2)$$

$$= 0.0195$$

Step 6 — Change in weight b/w hidden & i/p layer

$\Delta v_{01} = \alpha \, \delta_{in_{z1}} \times 1 = 0.25 \times 0.1056 \times 1 = 0.0264$

$\Delta v_{11} = -0.0264$ $(\alpha \, \delta_{in z1} x_1)$

$\Delta_{12} = 0.0264$ $(\alpha \, \delta_{in z2} x_2)$

$\Delta v_{02} = 0.0049$ $(\alpha \, \delta_{in z2} \times 1)$

$\Delta v_{21} = -0.0049$ $(\alpha \, \delta_{in z2} \times x_1)$

$\Delta v_{22} = +0.0049$ $(\alpha \, \delta_{in z2} x_2)$

Step 7 : Calculation of new weights.

$v_{01}(new) = v_{01}(old) + \Delta v_{01} = 0.3 + 0.0264 = \underline{0.3264}$

$v_{11}(new) = v_{11}(old) + \Delta v_{11} = 0.6 + (-0.0264) = \underline{0.5736}$

$v_{12}(new) = -0.1 + 0.0264 = \underline{-0.0736}$

$v_{02}(new) = 0.5 + 0.0049 = \underline{0.5049}$

$v_{21}(new) = -0.3 + 0.0049 = -0.2951$

$v_{22}(new) = 0.4 + 0.0049 = \underline{0.4049}$

$w_0(new) = -0.2 + \cancel{0.0271}\ 0.1373 = \underline{-0.0627}$

$w_1(new) = 0.4 + 0.0271 = \underline{0.3729}$

$w_2(new) = 0.1 + 0.0737 = \underline{0.1737}$