Page No. | Date

Que → four steps of Hebbian learning of a single-neuron network as in fig. 2.21 have been implemented starting with $w' = [1, -1]^T$ for learning constant $c = 1$ using i/ps as follows.

$$x_1 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad x_4 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

soln → Given

starting weight $W = [1, -1]^T$

learning constant $C = 1$

i/ps:

$x_1 = [1, -2]^T$, $x_2 = [0, 1]^T$, $x_3 = [2, 3]^T$

$x_4 = [1 -1]^T$

Activation funcn → Bipolar binary sigmoidal funcn with $\lambda = 1$ which is given by

$$y = \tanh(\lambda \cdot net) = \tanh(net)$$

where $net = W^T \cdot x$

Step by step Hebbain learnin process

(1) step I : Learning from $x_1 = [1, -2]^T$

- Net i/p calculation

$$net_1 = W^T x_1 = [1, -1]^T \times [1, -2]^T$$

$$= (1 \times 1) + (-1 \times -2) = 1 + 2 = 3.$$

- Activation y_1

$$y_1 = \tanh(3) \approx 0.995$$

Page No.
Date

- Weight update using the Hebbian learning rule.

$\Delta W_1 = C \cdot Y_1 \cdot X_1$

$= 1 \times 0.995 \cdot [1, -2]^T$

$\Delta W_1 = 0.995 [1, -2]^T = [0.995, 1.99]^T$

- New weight after step 1:

$W_1 = W + \Delta W_1 = [1, -1] + [0.995, 1.99]^T$

$= [1.995, -2.99]$

Step 2 → learning for $X_2 = [0, 1]^T$

- Net i/p calculation.

$net_2 = W_1^T X_2 = [1.995, -2.99] \cdot [0, 1]^T$

$= -2.99$

- Activation Y_2 -

$Y_2 = \tanh(-2.99) \approx -0.994$

- Weight update.

$\Delta W_2 = C \cdot Y_2 \cdot X_2 = 1 \cdot (-0.994) \cdot [0, 1]^T$

$= [0, -0.994]^T$

- New weight after step 2.

$W_2 = W_1 + \Delta W_2 = [1.995, -2.99] + [0, -0.994]$

$= [1.995, -3.984]$

Step 3 : learning from $x_3 = [2,3]^T$

- Net i/p calculation

$$net_3 = W^T x_3 = [1.995, -3.984] \cdot [2,3]^T$$
$$= (1.995 \times 2) + (-3.984 \times 3)$$
$$= 3.99 - 11.952$$
$$= -7.962.$$

- Activation y_3.

$$y_3 = \tanh(-7.962) \approx -0.999$$

- weight updates

$$\Delta W_3 = c \cdot y_3 \cdot x_3 = 1 \times (-0.999) \cdot [2,3]^T$$
$$= [-1.998, -2.997]^T$$

- New weight after step 3.

$$W_3 = W_2 + \Delta W_3 = [1.995, -3.984] + [-1.998, -2.997]$$
$$= [-0.003, -6.981]$$

Step 4 → learning from $x_4 = [1,-1]^T$

- Net i/p calculation.

$$net_4 = W^T x_4 = [-0.003, -6.981] \cdot [1,-1]^T$$
$$= (-0.003 \times 1) + (-6.981 \times -1)$$
$$= -0.003 + 6.981$$
$$= 6.978$$

- Activation y_4.

$$y_4 = \tanh(6.978) \approx 0.999$$

• weight update

$$\Delta W_4 = c \cdot y_4 \cdot x_4 = 1 \times 0.999 \times [1, -1]^T$$

$$= [0.999, -0.999]^T$$

• New weight after step 4.

$$W_4 = W_3 + \Delta W_4$$

$$= [-0.003, -6.981] + [0.999, -0.999]$$

$$= [0.996, -7.98]$$

Final weights:

After two steps of Hebbian learning, the final weights are approximately

$$W = [0.996, -7.98]$$