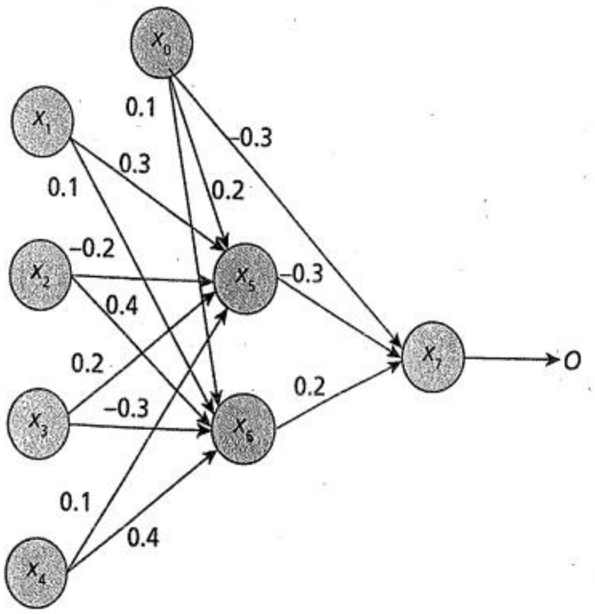
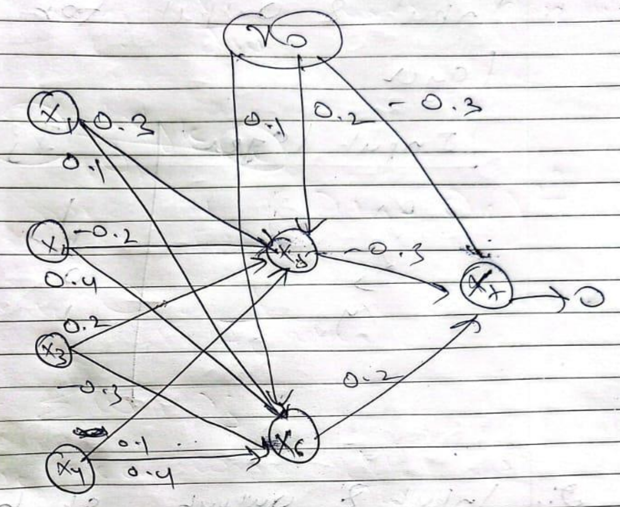
USN | | | | | | | | | |

Internal Assessment Test 2 – May 2025

	Sub:	Artificial Intelligence and Machine Learning
	Date:	27/05/2025
		Answer any 5 Questions

1	Consider learning in a Multi-Layer Perceptron. The given MLP consists of an Input layer, one Hidden layer and an Output layer. The input layer has 4 neurons, the hidden layer has 2 neurons and the output layer has a single neuron. **Train the MLP upto second iteration by updating the weights and biases in the network**. X1 = 1, X2 = 1, X3 = 0, X4 = 1, $O_{desired}$ = 1 , Learning rate = 0.8 	10

MLP Numerical



x_1	x_2	x_3	x_4	w_{15}	w_{16}	w_{25}	w_{26}	w_{35}	w_{36}	w_{45}	w_{46}
1	1	0	1	0.3	0.1	-0.2	0.4	0.2	-0.3	0.1	0.4

w_{57}	w_{67}	θ_5	θ_6	θ_7
-0.3	0.2	0.2	0.1	-0.3

Date / /

Step-1 Forward Propogation

1. Input & Output for Input layer

Input layer	I_j	O_j
x_1	1	1
x_2	1	1
x_3	0	0
x_4	1	1

2. Input & output at hidden layer and Output layer

Unit	Net input	Output
x_5	$I_5 = x_1 \times w_{15} + x_2 \times w_{25} + x_3 \times w_{35} + x_4 \times w_{45} + \theta_5$ $= 1 \times 0.3 + 1 \times -0.2 + 0 \times 0.2 + 1 \times 0.1 + 1 \times 0.2 = 0.4$	$O_5 = \frac{1}{1+e^{-0.4}}$ $= 0.599$

Page No.
Date / /

n_6	$I_6 = x_1 \times w_{16} + x_2 * w_{26}$ $+ x_3 \times w_{36} + x_4 \times w_{46} + \theta_6$ $= 1 \times 0.1 + 1 \times 0.4 + 0 \times 0.3$ $+ 1 \times 0.4 \times + 1 \times 0.1 = 1$	O_6 $= \frac{1}{1+e^{-1}}$ $= 0.73$
n_7	$I_7 = O_5 \times w_{57} + O_6 \times w_{67}$ $+ \theta_7$ $= 0.599 \times -0.3 + 0.73 \times$ $0.2 + 1 \times -0.3 = -0.33$	O_7 $= \frac{1}{1+e^{0.33}}$ $= 0.419$

3. Calculate error $O_{desired} - O_{estimated}$

$= O_{des} - O_7 = 1 - 0.419$
$= 0.581$

Date / /

Step - 2 Backward Prop

1. Calculate error at each node

Formulas :-

Output layer :-

$Error_k = O_k(1-O_k)(O_{desired} - O_k)$

For hidden layer :-

$Error_j = O_j(1-O_j)\sum_k Error_k w$

O/P layer	Error
x_7	$Error_7 = O_7(1-O_7)(O_{des} - O_7)$ $\Rightarrow 0.419 \times (1-0.419) \times (1-0.419)$ $= 0.141$

DOMS Page No.
Date / /

For hidden layer Error:

x_6

$Error_6 =$

$O_6(1-O_6)\sum_k Error_k\, w_{jk}$

$= O_6(1-O_6)\, Error_7\, w_{67}$

$= 0.73(1-0.73) \times 0.2 \times 0.141$

$= 0.005$

x_5

$Error_5$

$= O_5(1-O_5)\sum_k Error_k\, w_{jk}$

$= O_5(1-O_5)\, Error_7\, w_{57}$

$= 0.579(1-0.579) \times 0.1412 \times -0.3 = -0.0101$

Date / /

2. Update weights.

$$\Delta W_{ij} = \alpha \times Error_j \times O_i$$

$$W_{ij} = W_{ij} + \Delta W_{ij}$$

$$\alpha = 0.8$$

W_{ij}	$W_{ij} = W_{ij} + \alpha \times Error_j \times O_i$	Update weight
W_{15}	$W_{15} = W_{15} + 0.8 \times Error_5 \times O_1$ $= 0.3 + 0.8 \times -0.0101 \times 1$	0.29
W_{16}	$W_{16} = W_{16} + 0.8 \times Error_6 \times O_1$ $= 0.1 + 0.8 \times 0.005 \times 1$	0.104
W_{25}	$W_{25} = W_{25} + 0.8 \times Error_5 \times O_2$ $= -0.2 + 0.8 \times -0.0101 \times 1$	−0.208
W_{26}	$W_{26} = W_{26} + 0.8 \times Error_6 \times O_2$ $= 0.4 + 0.8 \times 0.005 \times 1$	0.404

Date / /

W_{35}	$W_{35} = W_{35} + 0.8 \times Error_5 \times O_3$	0.2
	$= 0.2 + 0.8 \times -0.0101 \times 0$	
W_{36}	$W_{36} = W_{36} + 0.8 \times Error_6 \times O_3$	-0.3
	$= -0.3 + 0.8 \times 0.005 \times 0$	
W_{45}	$W_{45} = W_{45} + 0.8 \times Error_5 \times O_4$	0.092
	$= 0.1 + 0.8 \times -0.0101 \times 1$	
W_{46}	$W_{46} = W_{46} + 0.8 \times Error_6 \times O_4$	0.404
	$= 0.4 + 0.8 \times 0.005 \times 1$	
W_{57}	$W_{57} = W_{57} + 0.8 \times Error_7 \times O_5$	-0.232
	$= -0.3 + 0.8 \times 0.141 \times 0.597$	
W_{67}	$W_{67} = W_{67} + 0.8 \times Error_7 \times O_6$	0.282
	$= 0.2 + 0.8 \times 0.141 \times 0.73$	

DOMS Page No.
Date / /

Update Bias

$\Delta\theta_j = \alpha \times Error_j$

$\theta_j = \theta_j + \Delta\theta_j$

θ_j	$\theta_j = \theta_j + \alpha \times Error_j$	Updated Bias
θ_5	$\theta_5 = \theta_5 + \alpha \times Error_5$ $= 0.2 + 0.8 \times -0.0101$	0.192
θ_6	$\theta_6 = \theta_6 + \alpha \times Error_6$ $= 0.1 + 0.8 \times 0.005$	0.104
θ_7	$\theta_7 = \theta_7 + \alpha \times Error_7$ $= -0.3 + 0.8 \times 0.1311$	-0.192

DOMS Page No.
Date / /

Iteration - 2

Calculate net Input & Output at hidden and Output layer

Unit	Net input I_j	Output
x_5	$I_5 = x_1 \times w_{15} + x_2 \times w_{25} + x_3 \times w_{35} + x_4 \times w_{45} + x_0 \times \theta_5$ $= 1 \times 0.0292 + 1 \times -0.208 + 0 \times 0.2 + 1 \times 0.092 + 1 \times 0.192 = 0.368$	$O_5 = \frac{1}{1+e^{-0.368}}$ $= 0.591$
x_6	$I_6 = x_1 \times w_{16} + x_2 \times w_{26} + x_3 \times w_{36} + x_4 \times w_{46} + x_0 \times \theta_6$ $= 1 \times 0.104 + 1 \times 0.404 + 0 \times -0.3 + 1 \times 0.404 + 1 \times 0.104 = 1.016$	$O_6 = \frac{1}{1+e^{-1.016}}$ $= 0.734$

x_7 | $I_7 = O_5 \times W_{57} + O_6 \times W_{67} + x_0 \times \theta_7$

$= 0.591 \times -0.232 + 0.234 \times 0.287 + 1 \times -0.187$

$= -0.114$

$O_7 = \frac{1}{1+e^{0.114}} = 0.471$

Output received at ~~unit~~ node X_7 is 0.471

Error $= 1 - 0.471 = 0.529$

Error reduction

$\Rightarrow 0.581 - 0.529$

0.052

2 A

Give the PDF for Exponential and Binomial Distribution along with parameters.

Exponential Distribution Formula

cuemath
THE MATH EXPERT

$$f(x) = m e^{-mx} \quad \text{or} \quad f(x) = \frac{1}{\mu} e^{-\frac{1}{\mu}x}$$

m = The Rate Parameter,
μ = Average time between occurences.

B

Binomial Distribution Formula

$$P(x) = \binom{n}{x} p^x q^{n-x} = \frac{n!}{(n-x)!\,x!} p^x q^{n-x}$$

where
n = the number of trials (or the number being sampled)
x = the number of successes desired
p = probability of getting a success in one trial
$q = 1 - p$ = the probability of getting a failure in one trial

B

Explain Parzen Window Algorithm

Parzen Window Let there be 'n' samples, $X = \{x_1, x_2, \cdots, x_n\}$

The samples are drawn independently, called as identically independent distribution. Let R be the region that covers 'k' samples of total 'n' samples. Then, the probability density function is given as:

$$p = k/n \tag{2.38}$$

The estimate is given as:

$$p(x) = \frac{k/n}{V} \tag{2.39}$$

where, V is the volume of the region R. If R is the hypercube centered at x and h is the length of the hypercube, the volume V is h^2 for 2D square cube and h^3 for 3D cube.

The Parzen window is given as follows:

$$\varphi\left(\frac{x_i - x}{h}\right) = \begin{cases} 1 & \text{if } \frac{|x_{ik} - x_k|}{h} < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{2.40}$$

The window indicates if the sample is inside the region or not. The Parzen probability density function estimate using Eq. (2.40) is given as:

$$p(x) = \frac{k/n}{V} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi\left(\frac{x_i - x}{h}\right) \tag{2.41}$$

This window can be replaced by any other function too. If Gaussian function is used, then it is called Gaussian density function.

KNN Estimation The KNN estimation is another non-parametric density estimation method. Here, the initial parameter k is determined and based on that k-neighbours are determined. The probability density function estimate is the average of the values that are returned by the neighbours.

A

Find LU Decomposition of the following 3*3 Matrix :

1 2 4
3 3 2

3

3 4 2

Example 2.9: Find LU decomposition of the given matrix:

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 3 & 3 & 2 \\ 3 & 4 & 2 \end{pmatrix}$$

Solution: First, augment an identity matrix and apply Gaussian elimination. The steps are as shown in:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 4 \\ 3 & 3 & 2 \\ 3 & 4 & 2 \end{bmatrix} \quad \boxed{\text{Initial Matrix}}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 4 \\ 0 & -3 & -10 \\ 3 & 4 & 2 \end{bmatrix} \quad \boxed{R_2 = R_2 - 3R_1}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 4 \\ 0 & -3 & -10 \\ 0 & -2 & -10 \end{bmatrix} \quad \boxed{R_3 = R_3 - 3R_1}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 3 & \frac{2}{3} & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 4 \\ 0 & -3 & -10 \\ 0 & 0 & \frac{-10}{3} \end{bmatrix} \quad \boxed{R_3 = R_3 - \frac{2}{3}R_2}$$

Now, it can be observed that the first matrix is L as it is the lower triangular matrix whose values are the determiners used in the reduction of equations above such as 3, 3 and 2/3. The second matrix is U, the upper triangular matrix whose values are the values of the reduced matrix because of Gaussian elimination.

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 3 & \frac{2}{3} & 1 \end{pmatrix} \text{ and } U = \begin{pmatrix} 1 & 2 & 4 \\ 0 & -3 & -10 \\ 0 & 0 & -\frac{10}{3} \end{pmatrix}.$$

AIS

B

Find the Correlation Coefficient of Data :

X = {1, 2, 3, 4, 5 }, y = {1, 4, 9, 16, 25}

Solution: Mean(X) = $E(X) = \frac{15}{5} = 3$, Mean(Y) = $E(Y) = \frac{55}{5} = 11$. The covariance is computed using Eq. (2.17) as:

$$\frac{(1-3)(1-11)+(2-3)(4-11)+(3-30)(9-11)+(4-3)(16-11)+(5-3)(25-11)}{5} = 12$$

Solution: The mean values of X and Y are $\frac{15}{5} = 3$ and $\frac{55}{5} = 11$. The standard deviations of X and Y are 1.41 and 8.6486, respectively. Therefore, the correlation coefficient is given as ratio of covariance (12 from the previous problem 2.5) and standard deviation of x and y as per Eq. (2.18) as:

$$r = \frac{12}{1.41 \times 8.6486} \approx 0.984$$

4

A

Explain the following terms : Hypothesis Testing, p-value and Confidence Interval

1. Hypothesis Testing

Definition:
A **statistical method** used to make decisions or inferences about a population based on sample data.

Key Idea:
You begin with a **null hypothesis (H_0)** — usually a statement of "no effect" or "no difference" — an
alternative hypothesis (H_1 or H_a).

Example:
H_0: A new drug has no effect.
H_1: The new drug has a significant effect.

---

2. p-value

Definition:
The **p-value** is the probability of obtaining a result **as extreme or more extreme** than the observed r
null hypothesis is true.

Interpretation:

- **Low p-value (typically < 0.05)** → strong evidence against H_0 → reject H_0.
- **High p-value** → weak evidence against H_0 → fail to reject H_0.

Example:
A p-value of 0.02 means there is a 2% chance the result occurred under H_0 → statistically significant

---

3. Confidence Interval (CI)

Definition:
A **range of values** within which we believe the **true population parameter** lies, with a certain level
(e.g., 95%).

Example:
If the 95% confidence interval for the mean weight of students is (58 kg, 62 kg), we are 95% confide
average lies within this range.

<table>
<tr><td></td><td>B</td><td>Find the 5-point summary of the list : {13, 11, 2, 3, 4, 8, 9 }

Example 2.5: Find the 5-point summary of the list {13, 11, 2, 3, 4, 8, 9}.

Solution: The minimum is 2 and the maximum is 13. The Q_1, Q_2 and Q_3 are 3, 8 and 11, respectively. Hence, 5-point summary is {2, 3, 8, 11, 13}, that is, {minimum, Q_1, median, Q_3, maximum}.

Box plots are useful for describing 5-point summary. The Box plot for the set is given in Figure 2.7.

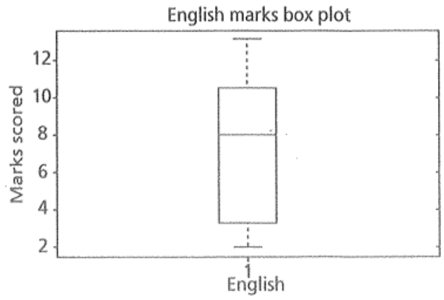



Figure 2.7: Box Plot for English Marks</td></tr>
<tr><td></td><td>A</td><td>Write and Explain the complete algorithm of Perceptron.</td></tr>
</table>

5

The first neural network model 'Perceptron', designed by Frank Rosenblatt in 1958, is a linear binary classifier used for supervised learning. He modified the McCulloch & Pitts Neuron model by combining two concepts, McCulloch-Pitts model of an artificial neuron and Hebbian learning rule of adjusting weights. He introduced variable weight values and an extra input that represents *bias* to this model. He proposed that artificial neurons could actually learn weights and thresholds from data and came up with a supervised learning algorithm that enabled the artificial neurons to learn the correct weights from training data by itself. The perceptron model (shown in Figure 10.5) consists of 4 steps:

1. Inputs from other neurons
2. Weights and bias
3. Net sum
4. Activation function

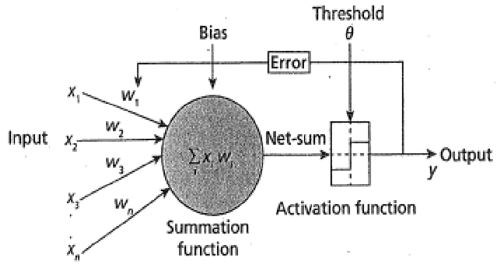


Figure 10.5: Perceptron Model

Thus, the modified neuron model receives a set of inputs $x_1, x_2, \ldots, x_n$, their associated weights $w_1, w_2, \ldots, w_n$ and a bias. The summation function 'Net-sum' Eq. (10.13) computes the weighted sum of the inputs received by the neuron.

Algorithm 10.1: Perceptron Algorithm

Set initial weights $w_1, w_2, \ldots, w_n$ and bias θ to a random value in the range [−0.5, 0.5].

For each Epoch,

1. Compute the weighted sum by multiplying the inputs with the weights and add the products.
2. Apply the activation function on the weighted sum:

$$Y = \text{Step}\ ((x_1 w_1 + x_2 w_2) - \theta)$$

3. If the sum is above the threshold value, output the value as positive else output the value as negative.
4. Calculate the error by subtracting the estimated output $Y_{estimated}$ from the desired output $Y_{desired}$:

$$\text{error } e(t) = Y_{desired} - Y_{estimated}$$

[If error $e(t)$ is positive, increase the perceptron output Y and if it is negative, decrease the perceptron output Y.]

5. Update the weights if there is an error:

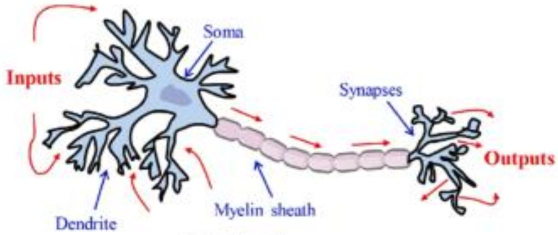
$$\Delta w_i = \propto \times e(t) \times x_i$$

$$w_i = w_i + \Delta w_i$$

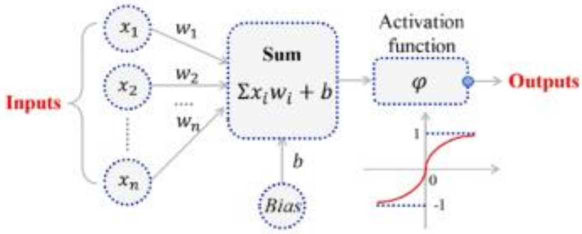
where, x_i is the input value, $e(t)$ is the error at step t, $\propto$ is the learning rate and Δw_i is the difference in weight that has to be added to w_i.

B

Compare with diagram, Biological Neural Network and Artificial Neural Network



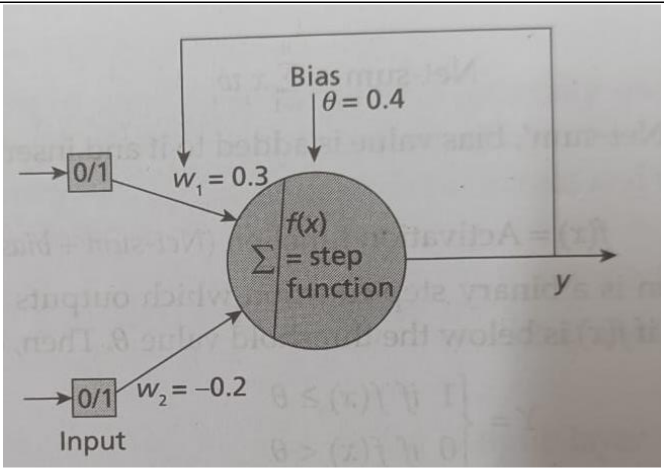
(a) Biological neuron



(b) Artificial neuron

Aspect	Artificial Neural Network (ANN)	Biological Neural Network (BNN)
Structure	Made of artificial neurons arranged in layers	Composed of real neurons connected via synapses
Signal Transmission	Uses numerical values and weighted sums	Uses electrochemical signals and neurotransmitters
Learning Mechanism	Trained using algorithms like backpropagation	Learns through synaptic plasticity (e.g., Hebbian learning)
Speed & Complexity	Fast, but limited in adaptability and energy efficiency	Slower, but highly adaptive and energy efficient

6	Consider a perceptron to represent the Boolean function AND with the initial weights w1=0.3, w2 = -0.2, learning rate α =0.2 and bias ϴ = 0.4 as shown in Figure. The activation function used here is the Step function f(x) which gives the output value as binary, i.e., 0 or 1. If value of f(x) is greater than or equal to 0, it outputs 1 or else it outputs 0. Design a perceptron that performs the Boolean function AND and update the weights until the Boolean function gives the desired output.	10



Solution: Desired output for Boolean function AND is shown in Table 10.1.

Table 10.1: AND Truth Table

x_1	x_2	Y_{des}
0	0	0
0	1	0
1	0	0
1	1	1

For each Epoch, weighted sum is calculated and the activation function is applied to compute the estimated output Y_{est}. Then, Y_{est} is compared with Y_{des} to find the error. If there is an error, the weights are updated.

Tables 10.2 to 10.5 show how the weights are updated in the four Epochs.

Table 10.2: Epoch 1

Epoch	x_1	x_2	Y_{des}	Y_{est}	Error	w_1	w_2	Status
1	0	0	0	Step $((0 \times 0.3 + 0 \times -0.2) - 0.4) = 0$	0	0.3	−0.2	No change
	0	1	0	Step $((0 \times 0.3 + 1 \times -0.2) - 0.4) = 0$	0	0.3	−0.2	No change
	1	0	0	Step $((1 \times 0.3 + 0 \times -0.2) - 0.4) = 0$	0	0.3	−0.2	No change
	1	1	1	Step $((1 \times 0.3 + 1 \times -0.2) - 0.4) = 0$	1	0.5	0	Change

For input (1, 1) the weights are updated as follows:

$\Delta w_1 = \propto \times e(t) \times x_1 = 0.2 \times 1 \times 1 = 0.2$

$w_1 = w_1 + \Delta w_1 = 0.3 + \Delta w_1 = 0.3 + 0.2 = 0.5$

$\Delta w_2 = \propto \times e(t) \times x_2 = 0.2 \times 1 \times 1 = 0.2$

$w_2 = w_2 + \Delta w_1 = -0.2 + \Delta w_2 = -0.2 + 0.2 = 0$

Table 10.3: Epoch 2

Epoch	x_1	x_2	Y_{des}	Y_{est}	Error	w_1	w_2	Status
2	0	0	0	Step $((0 \times 0.5 + 0 \times 0) - 0.4) = 0$	0	0.5	0	No change
	0	1	0	Step $((0 \times 0.5 + 1 \times 0) - 0.4) = 0$	0	0.5	0	No change
	1	0	0	Step $((1 \times 0.5 + 0 \times 0) - 0.4) = 1$	–1	0.3	0	Change
	1	1	1	Step $((1 \times 0.3 + 1 \times 0) - 0.4) = 0$	1	0.5	0.2	Change

For input (1, 0) the weights are updated as follows:

$\Delta w_1 = \propto \times e(t) \times x_1 = 0.2 \times -1 \times 1 = -0.2$

$w_1 = w_1 + \Delta w_1 = 0.5 + \Delta w_1 = 0.5 - 0.2 = 0.3$

$\Delta w_2 = \propto \times e(t) \times x_2 = 0.2 \times -1 \times 0 = 0$

$w_2 = w_2 + \Delta w_2 = 0 + \Delta w_2 = 0 + 0 = 0$

For input (1, 1), the weights are updated as follows:

$\Delta w_1 = \propto \times e(t) \times x_1 = 0.2 \times 1 \times 1 = 0.2$

$w_1 = w_1 + \Delta w_1 = 0.3 + \Delta w_1 = 0.3 + 0.2 = 0.5$

$\Delta w_2 = \propto \times e(t) \times x_2 = 0.2 \times 1 \times 1 = 0.2$

$w_2 = w_2 + \Delta w_2 = 0 + \Delta w_2 = 0 + 0.2 = 0.2$

Table 10.4: Epoch 3

Epoch	x_1	x_2	Y_{des}	Y_{est}	Error	w_1	w_2	Status
3	0	0	0	Step $((0 \times 0.5 + 0 \times 0.2) - 0.4) = 0$	0	0.5	0.2	No change
	0	1	0	Step $((0 \times 0.5 + 1 \times 0.2) - 0.4) = 0$	0	0.5	0.2	No change
	1	0	0	Step $((1 \times 0.5 + 0 \times 0.2) - 0.4) = 1$	–1	0.3	0.2	Change
	1	1	1	Step $((1 \times 0.3 + 1 \times 0.2) - 0.4) = 1$	0	0.3	0.2	No change

For input (1, 0) the weights are updated as follows:

$$\Delta w_1 = \propto \times e(t) \times x_1 = 0.2 \times -1 \times 1 = -0.2$$

$$w_1 = w_1 + \Delta w_1 = 0.5 + \Delta w_1 = 0.5 - 0.2 = 0.3$$

$$\Delta w_2 = \propto \times e(t) \times x_2 = 0.2 \times -1 \times 0 = 0$$

$$w_2 = w_2 + \Delta w_2 = 0 + \Delta w_2 = 0.2 + 0 = 0.2$$

Table 10.5: Epoch 4

Epoch	x_1	x_2	Y_{des}	Y_{est}	Error	w_1	w_2	Status
4	0	0	0	Step $((0 \times 0.3 + 0 \times 0.2) - 0.4) = 0$	0	0.3	0.2	No change
	0	1	0	Step $((0 \times 0.3 + 1 \times 0.2) - 0.4) = 0$	0	0.3	0.2	No change
	1	0	0	Step $((1 \times 0.3 + 0 \times 0.2) - 0.4) = 0$	0	0.3	0.2	No change
	1	1	1	Step $((1 \times 0.3 + 1 \times 0.2) - 0.4) = 1$	0	0.3	0.2	No change

It is observed that with 4 Epochs, the perceptron learns and the weights are updated to 0.3 and 0.2 with which the perceptron gives the desired output of a Boolean AND function.

CCI CI HOD