Q2) Using the Hebb rule, find the weights required to perform the following classifications:
Given that the vectors (1,1,1,1) and (-1,1,-1,-1) are members of the same class (target 1) and vectors (1,1,1,-1) and (1,-1,-1,1) are not members of the class (target -1).

⇒

Input Vector (x)	Target (t)
(1,1,1,1)	1
(-1,1,-1,-1)	1
(1,1,1,-1)	-1
(1,-1,-1,1)	-1

Initial weights:

$$w = (0,0,0,0)$$

Bias:

$$b = 0.$$

Hebb Rule formula

weight:-

$$w_{new} = w_{old} + x \times t$$

Bias:

$$b_{new} = b_{old} + t$$

i) $x_1 = (1,1,1,1)$ $t = 1$

weight:-

$w = (0,0,0,0) + (1,1,1,1)$

$w = (1,1,1,1).$

Bias:

$b = 0 + 1 = 1.$

ii) $x_2 = (-1,1,-1,-1)$ $t = 1$

$w = (1,1,1,1) + (-1,1,-1,-1)$

$w = (0,2,0,0)$

Bias. = $1 + (1 \times 1)$

$= 2.$

iii) $t = -1$, $x = (1,1,1,-1)$

$w = (0,2,0,0) + (-1,-1,-1,1)$

$w = (-1,1,-1,1)$

bias $= 2 - 1 = 1$ //

iv) $t = -1$ $x = (1, -1, -1, 1)$

$w = (-1, 1, -1, 1) + (-1, 1, 1, -1)$

$= (-2, 2, 0, 0)$

$b = 1 - 1 = 0$

Final weight vector :

$$w = (-2, 2, 0, 0)$$

Final bias :

$$b = 0.$$

2b) Implement the following logical function using the M-P neuron. Use binary data representation.

1. AND
2. OR
3. XOR.

⇒ The McCulloch-Pitts (M-P) neuron is a binary neuron model. It gives output based on the weighted sum of inputs & a threshold.

Let : inputs : x_1, x_2

weights : w_1, w_2

Threshold : θ

Output :

$$y = \begin{cases} 1 & \text{if } w_1 x_1 + w_2 x_2 \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

Binary values : 0 and 1.

1. AND Function

x_1	x_2	Output
0	0	0
0	1	0
1	0	0
1	1	1

$w_1 = 1$ $w_2 = 1$ $\theta = 2$

Condition:-

$Y = 1$ if $x_1 + x_2 \geq 2$.

Only when both inputs are 1, output is 1.

Hence AND function is implemented.

2. OR Function.

x_1	x_2	Output
0	0	0.
0	1	1
1	0	1
1	1	1

$w_1 = 1$ $w_2 = 1$ $\theta = 1$

condition:

$Y = 1$ if $x_1 + x_2 \geq 1$

If any input is 1, output is 1

Hence OR function is implemented.

3. XOR function.

x_1	x_2	output
0	0	0
0	1	1
1	0	1
1	1	0

XOR cannot be implemented using a single M-P neuron because it is not linearly seperable. It requires multiple neurons.

Implementation using Multiple Neurons.

$$XOR = (x_1 \text{ AND } \overline{x_2}) + (\overline{x_1} \text{ AND } x_2)$$

Using:-

Neuron 1 → AND

Neuron 2 → AND

Neron 3 → OR

Find output gives XOR.