SOFT COMPUTING

B.Tech(CSE), Winter 2015
Worked Out Examples (SET-1)

B.K.Tripathy
Senior Professor
School of Computer Science and Engineering
VIT University, Vellore-632014
Tamil Nadu, India

EXAMPLE-1

Q. Implement ANDNOT function using McCullah-Pitts neuron (Using binary data representation)

A. An ANDNOT function, the response is true (1) if the first input is true(1) and the second input is false (0)

Table:

x1	x2	y
0	0	0
0	1	0
1	0	1
1	1	0

EXAMPLE-1 CONTD…

- **Case-1:** Assume that both weights w1 and w2 are excitatory
- That is we take w1 = w2 = 1
- Using this the formula for the input is
- yin = x1w1+x2w2
- So,

Input	Input Computation	Input Value
(1,1)	Yin = 1x1+1x1	2
(1,0)	Yin = 1x1+0x1	1
(0,1)	Yin=0x1+1x1	1
(0,0)	Yin=0x1+0x1	0

EXAMPLE-1 CONTD…

- From the net inputs it is not possible to fire the neuron for input (1,0) only
- **Case-2:** Assume that one weight is excitatory and the other one is inhibitory, i.e. w1 =1 and w2 = -1.
- Using the same formula

Input	Input Computation	Input Value
(1,1)	Yin=1x1+1x-1	0
(1,0)	Yin=1x1+0x-1	1
(0,1)	Yin=0x1+1x-1	-1
(0,0)	Yin=0x1+0x-1	0

EXAMPLE-1 CONTD…

- From the computed net input it is now possible to fire (1,0) alone. We have to fix the threshold value t=1
- So, w1=1, w2 = -1 and t = 1
- The value of ‘t’ can be computed from t >= n.w – p
- i.e. t>= 2x1-1 or t>= 1
- Note that for inhibitory ‘p’ only the magnitude only is considered
- **The output of neuron ‘y’ can be written as**
- y = f(yin) = 1 if yin >= 1 and
- = 0 if yin <1

EXAMPLE-2

- **Implement XOR function using M-P neuron**
- The truth table for the XOR is given by

x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	0

- The output is "ON" only for odd number of 1's

EXAMPLE-2 CONTD…

- The XOR function can be represented as

- y = x1. x2+x1.x2 or y = z1 + z2; where
- z1 = x1.x2 (**function 1**)
- z2 = x1.x2 (**function 2**)
- y = z1 (OR) z2 (**function 3**)

- **NOTE: In this case we have to use an intermediate layer to compute (function 1) and (function 2).**
- **The final output is (function 3)**

FIRST FUNCTION

- **First Function**($z_1 = x_1\overline{x_2}$)
- The truth table for this function is

x1	x2	z1
0	0	0
0	1	0
1	0	1
1	1	0

COMPUTATIONS FOR Z1

- **Case 1:** Assume both weights are excitatory($w_{11} = w_{21} = 1$)
- Computation of net inputs

inputs	Zin computations	Zin values
(0, 0)	**Z1in = 0x1 + 0x1**	**0**
(0, 1)	**Z1in = 0x1 + 1x1**	**1**
(1, 0)	**Z1in = 1x1 + 0x1**	**1**
(1, 1)	**Z1in = 1x1 + 1x1**	**2**

- It is not possible to obtain function z1 using these weights

COMPUTATIONS FOR Z1

- **Case 2:** Assume that one of the weights is excitatory and the other one is inhibitory $(w_{11} = 1; w_{21} = -1)$
- The computation of inputs is

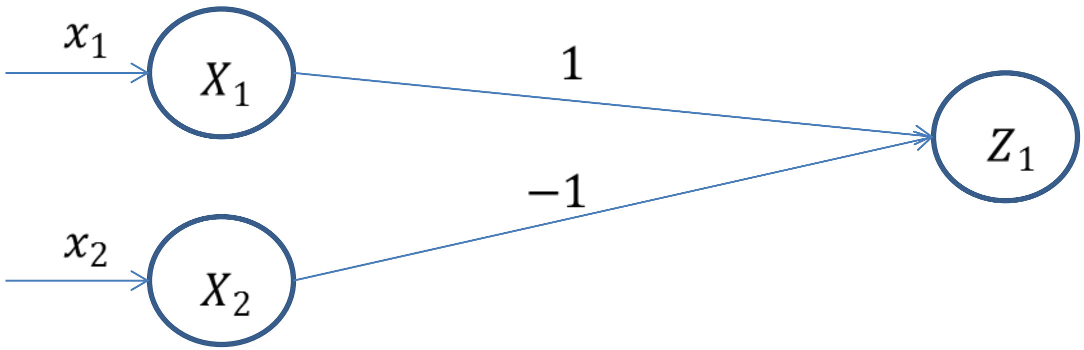
inputs	Zin computations	Zin values
(0, 0)	**Z1in = 0x1 + 0x-1**	**0**
(0, 1)	**Z1in = 0x1 + 1x-1**	**-1**
(1, 0)	**Z1in = 1x1 + 0x-1**	**1**
(1, 1)	**Z1in = 1x1 + 1x-1**	**0**

- On the basis of this calculated net input it is possible to get the required output is

$$w_{11} = 1; w_{21} = -1; \theta \geq 1$$

NET FOR Z1

- The nets corresponding to z_1 is



SECOND FUNCTION

- **Second Function:** ($z_2 = \overline{x_1} x_2$)
- The truth table for z2 is

x1	x2	z2
0	0	0
0	1	1
1	0	0
1	1	0

COMPUTATIONS FOR Z2

- **Case 1:** Assume both weights are excitatory $(w_{12} = w_{22} = 1)$
- Computation of net inputs

inputs	Zin computations	Zin values
(0, 0)	Z2in = 0x1 + 0x1	0
(0, 1)	Z2in = 0x1 + 1x1	1
(1, 0)	Z2in = 1x1 + 0x1	1
(1, 1)	Z2in = 1x1 + 1x1	2

- Hence it is not possible to obtain function z2 using these weights

COMPUTATIONS FOR Z2

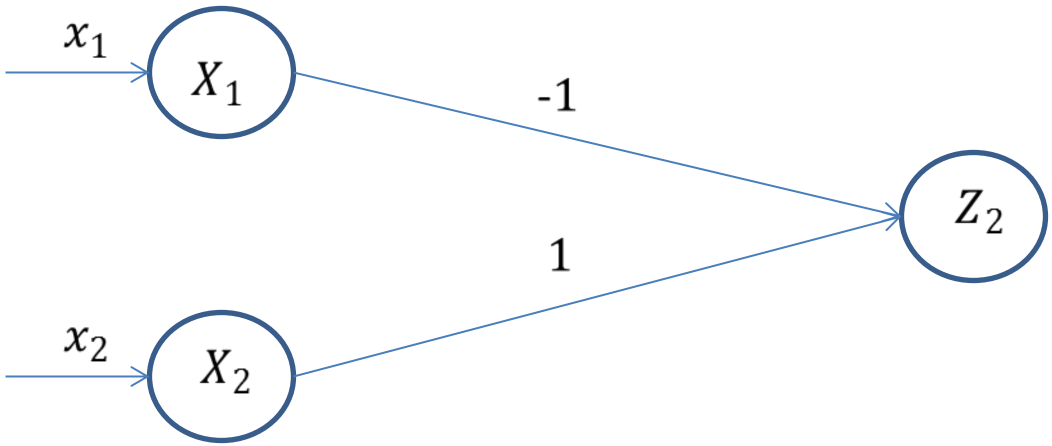
- **Case 2:** Assume that one of the weights is excitatory and the other one is inhibitory $(w_{12} = -1; w_{22} = 1)$
- Computation of net inputs

inputs	Zin computations	Zin values
(0, 0)	Z2in = 0x-1 + 0x1	0
(0, 1)	Z2in = 0x-1 + 1x1	1
(1, 0)	Z2in = 1x-1 + 0x1	-1
(1, 1)	Z2in = 1x-1 + 1x1	0

- Thus with these weights it is possible to get the output
- Hence
- $w_{12} = -1; w_{22} = 1 \& \theta \geq 1$

NET FOR Z2

- The 2 nets corresponding to z_2 is



THIRD FUNCTION

- The third function is $y = z_1\ OR\ z_2$
- The truth table for this function is

x1	x2	z1	z2	y
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0

- The net input can be computed by using

$$y_{in} = z_1 v_1 + z_2 v_2, v_1 \& v_2\ are\ weights$$

COMPUTATIONS FOR Y

- **Case 1:** Assume that both the functions are excitatory

$$v_1 = v_2 = 1$$

- Computations for net input are

inputs	z1	z2	yin computations	yin values
(0, 0)	0	0	yin = 0x1 + 0x1	0
(0, 1)	0	1	yin = 0x1 + 1x1	1
(1, 0)	1	0	yin = 1x1 + 0x1	1
(1, 1)	0	0	yin = 0x1 + 0x1	0

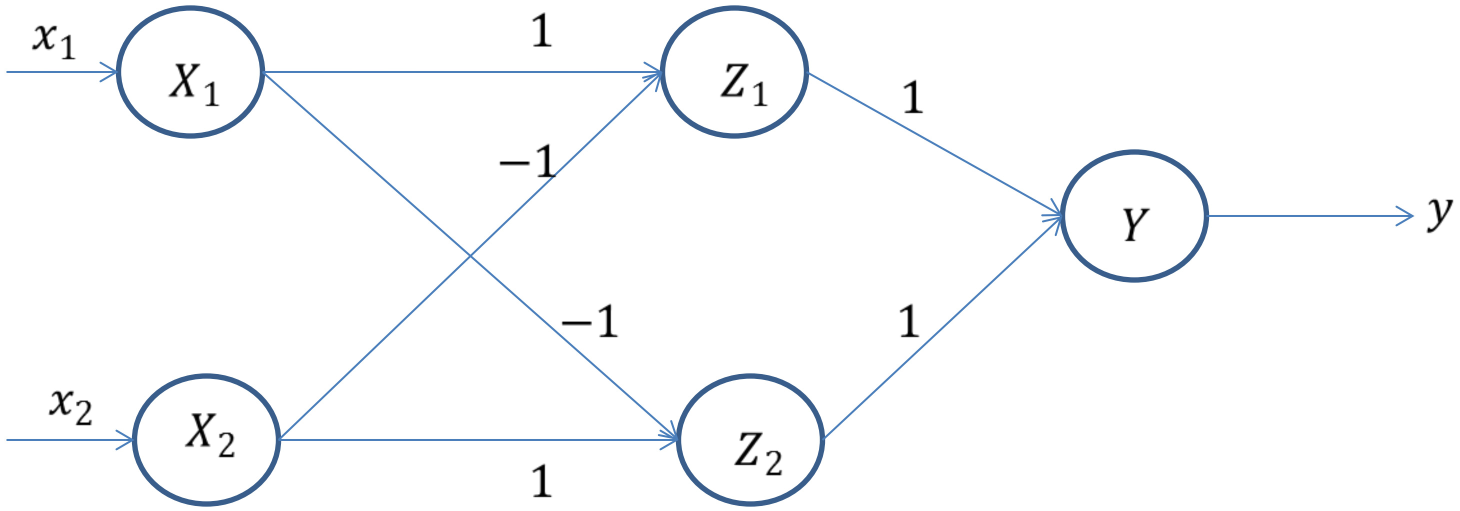
FINAL ANALYSIS

- Setting a threshold value $\theta \geq 1$ and $v_1 = v_2 = 1$, the net is recognized
- So, the weights for the XOR function using M-P neuron are

$$\begin{aligned} &w_{11} = 1; w_{12} = -1; \\ &w_{21} = -1; w_{22} = 1; \\ &v_1 = 1; \qquad v_2 = 1. \end{aligned}$$

NET REPRESENTATION FOR EXAMPLE-2

- e



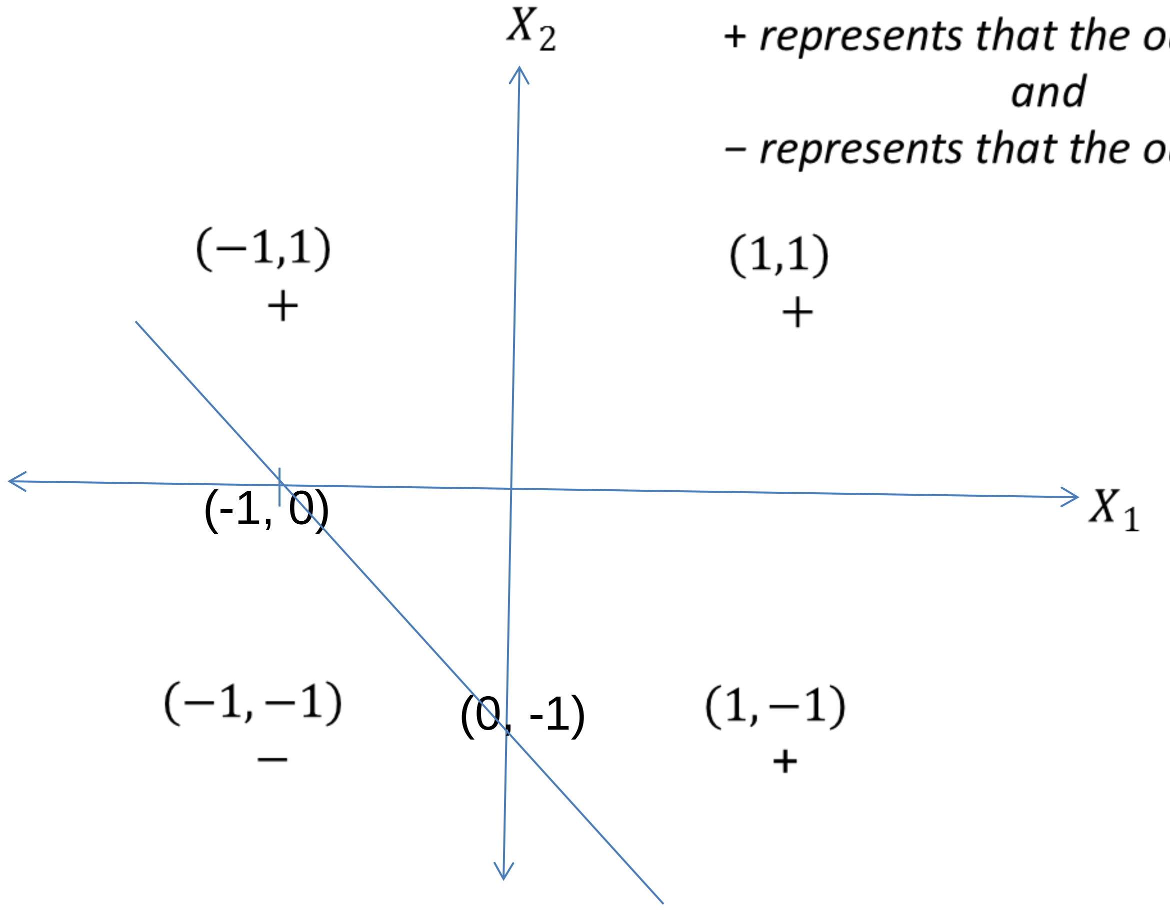
EXAMPLE-3

- **Using the linear separability concept obtain the response for OR function (Take bipolar inputs and bipolar targets)**
- The truth table for 'OR' function with bipolar inputs and targets is

x1	x2	y
1	1	1
1	-1	1
-1	1	1
-1	-1	-1

GRAPHICAL REPRESENTATION

- a



+ represents that the output is +1
and
− represents that the output is −1

COMPUTATIONS

- Let us take the line of separation as the line joining the points (-1, 0) and (0, -1)
- We get the slope of the line of separation as

$$m = \frac{y_2 - y_1}{x_2 - x_1} = -1$$

- The value of ‘c’ is obtained as c = y – m.x = 0 – (-1) (-1) = -1
- So, the equation of the line is y = -x -1
- Since the axes used are X_1 *and* X_2 the equation is written as
- $$x_2 = -x_1 - 1$$

COMPUTATIONS CONTD…

- The above equation can be written as $x_2 = \frac{-w_1}{w_2} x_1 - \frac{b}{w_2}$
- Comparing the two equations $\frac{w_1}{w_2} = 1 \; and \; \frac{b}{w_2} = 1$
- So, $w_1 = w_2 = b = 1$
- Computing the net input and output of OR function on the basis of these weights and bias we get

x1	x2	b	Yin=b+x1w1+x2w2	y
1	1	1	3	1
1	-1	1	1	1
-1	1	1	1	1
-1	-1	1	-1	-1

FINAL ANALYSIS

- From the table the output of Y can be written as

$$y = f(y_{in}) = \begin{cases} 1, & if\ y_{in} \geq 1; \\ 0, & if\ y_{in} < 1. \end{cases}$$

- Here, the threshold is taken to be ‘1’.

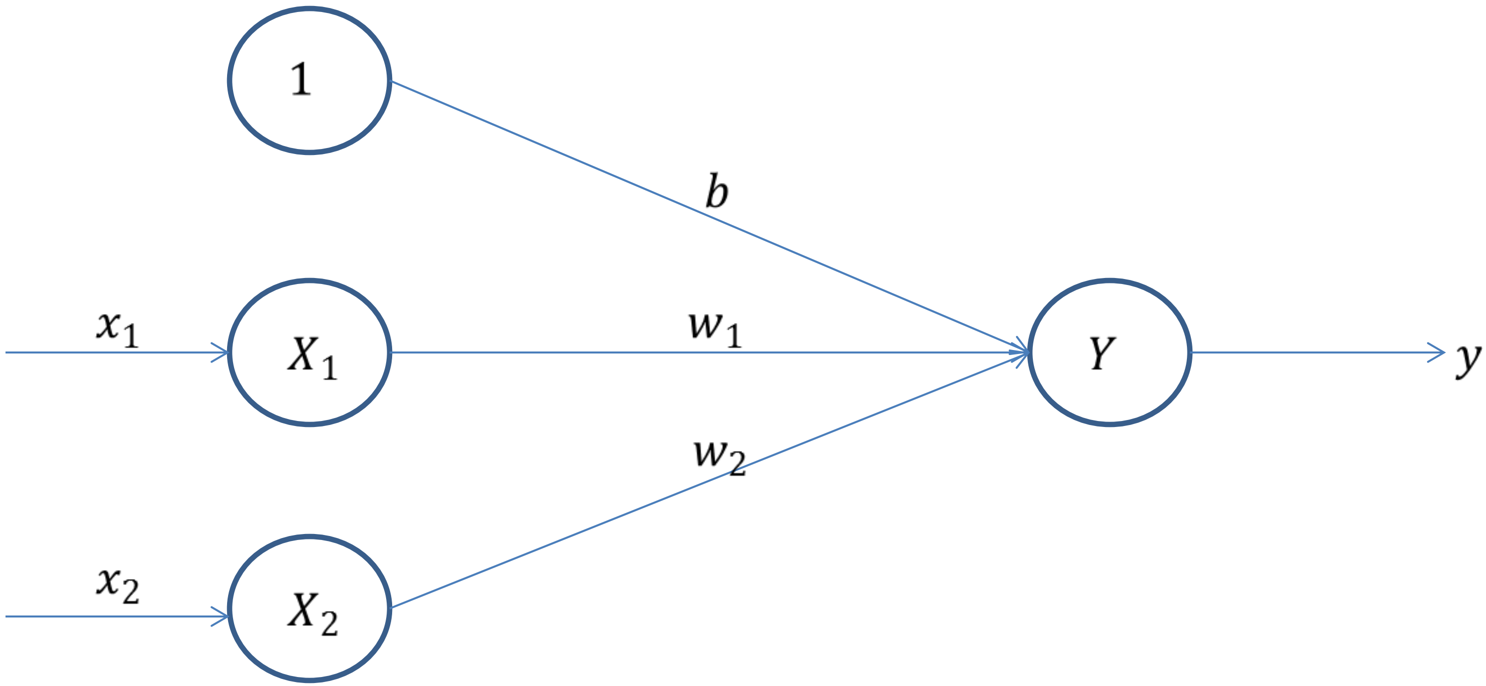
EXAMPLE-4

- **Implement AND function using perceptron networks for bipolar inputs and targets**
- The truth table for the AND function using bipolar inputs and targets is

x1	x2	y
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1

PERCEPTRON NETWORK FOR AND FUNCTION

- s



COMPUTATIONS

- **All the inputs are to be presented to the network one after another to complete one epoch**
- Let us set the initial weights and bias as $w_1 = w_2 = 0 = b$
- The net input for the first input is $y_{in} = b + w_1 x_1 + w_2 x_2 = 0$
- The output 'y' is computed by applying activation function over the net input. So, taking the threshold $\theta = 0,$

$$y = \begin{cases} 1, & if\ y_{in} > 0; \\ 0, & if\ y_{in} = 0; \\ -1, & if\ y_{in} < 0. \end{cases}$$

- As t = 1 and y = 0, we have $t \neq y$. So, weight updation takes place

COMPUTATIONS

- Using

$$w_i(new) = w_i(old) + \alpha.t.x_i$$

$$w_1(new) = w_1(old) + 1.1.1 = 1$$
$$w_2(new) = w_2(old) + 1.1.1 = 1$$
$$b(new) = b(old) + 1.1 = 1$$

- These values are used for the next input

COMPUTATIONS

In	put	b	Target	input	Output	Updtd	Weig	hts
x1	x2	1	t	yin	y	w1	w2	b
EPOCH 1								
1	1	1	1	0	0	1	1	1
1	-1	1	-1	1	1	0	2	0
-1	1	1	-1	1	1	1	1	-1
-1	-1	1	-1	-3	-1	1	1	-1
EPOCH 2								
1	1	1	1	1	1	1	1	-1
1	-1	1	-1	-1	-1	1	1	-1
-1	1	1	-1	-1	-1	1	1	-1
-1	-1	1	-1	-1	-1	1	1	-1

COMPUTATION

- **The stopping conditions are:**
- **All the calculated outputs are equal to the target outputs**
- When a separating line is obtained using the final weights for separating the positive responses from the negative responses
- **After the second epoch the target outputs are equal to the computed outputs**
- The weights after the second epoch are $w_1 = w_2 = 1 \; and \; b = -1$
- Since the threshold value is '0' the separating line is

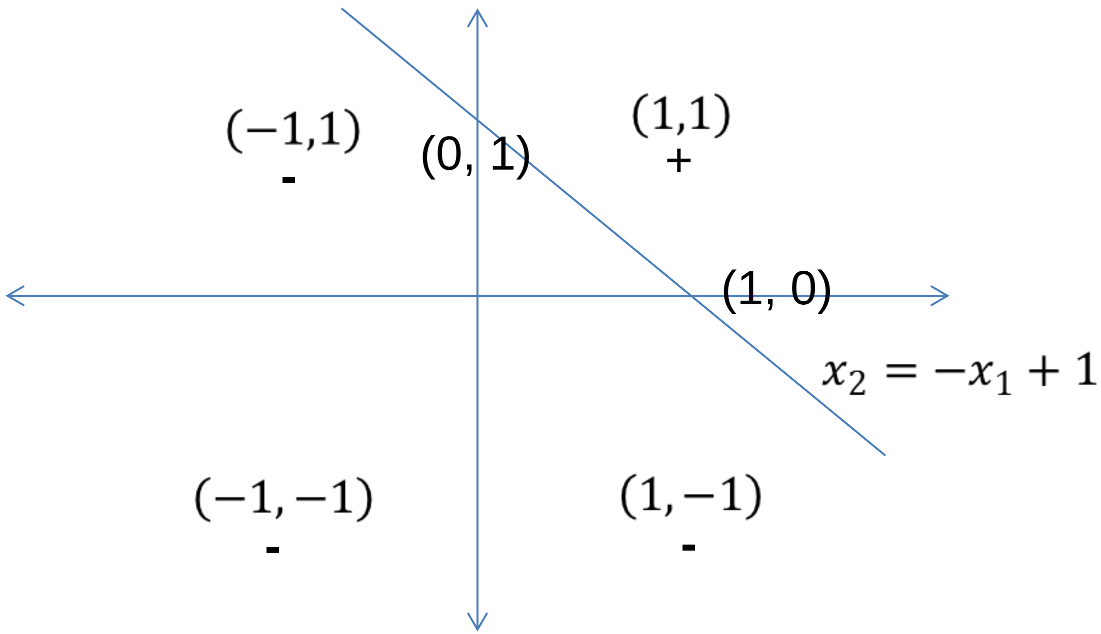
COMPUTATION

- s

$$x_2 = \frac{-w_1}{w_2} x_1 - \frac{b}{w_2}$$

- So,

$$x_2 = -\frac{1}{1} x_1 - \frac{-1}{1} \; or \; x_2 = -x_1 + 1$$



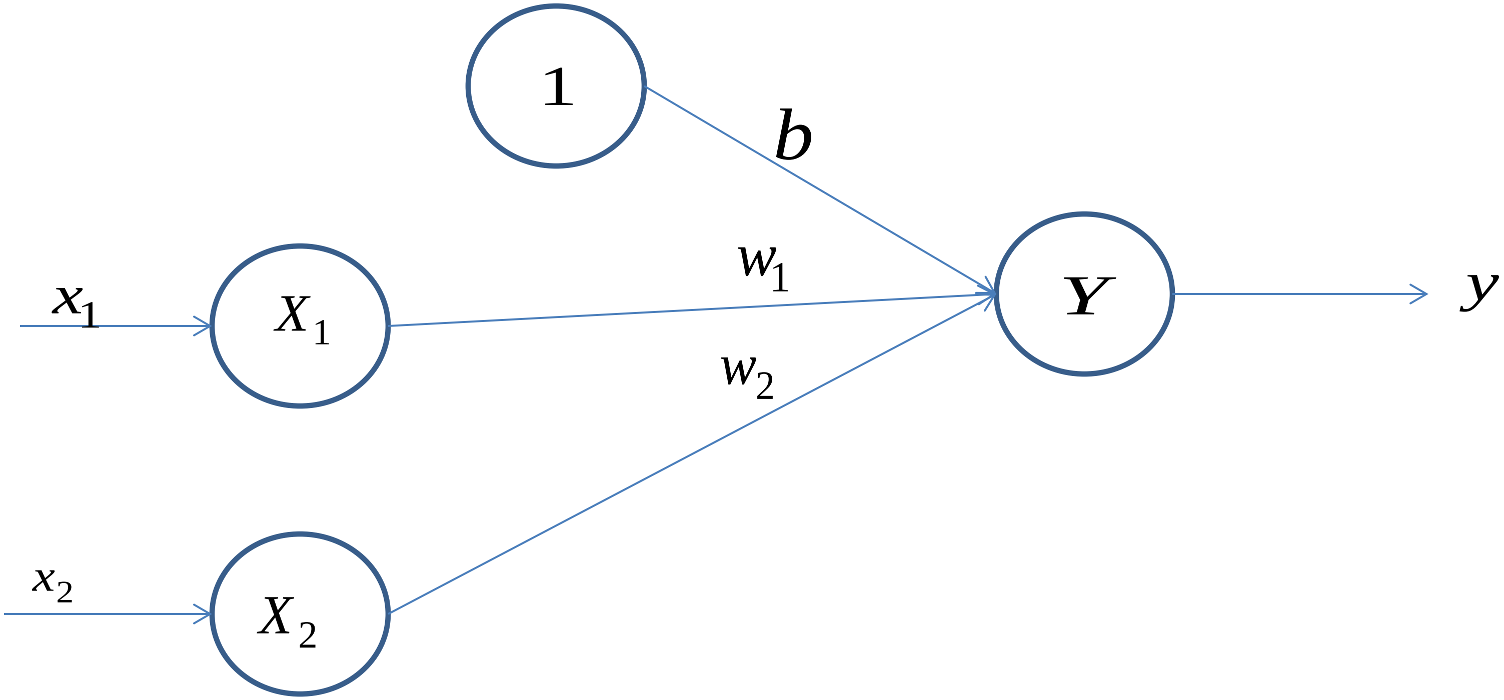
EXAMPLE-5

- **Find the weights using perceptron network for ANDNOT function when all the inputs are presented only one time. Use bipolar inputs and targets**
- The truth table for the ANDNOT function using bipolar inputs and targets is

x1	x2	t
1	1	-1
1	-1	1
-1	1	-1
-1	-1	-1

NETWORK STRUCTURE

- s



COMPUTATIONS

- Let us take $w_1 = w_2 = 0; \alpha = 1, \theta = 0$
- **First input:** (1, 1, -1)

$$y_{in} = b + x_1 w_1 + x_2 w_2 = 0 + 1 \times 0 + 1 \times 0 = 0$$

- The output using activation function is

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

COMPUTATIONS

- So, output (y = 0) $\neq$ (t = -1)
- So **weight updation is necessary**

$$w_1(new) = w_1(old) + \alpha.t.x_1$$
$$= 0 + 1 \times (-1) \times 1$$
$$= -1$$
$$w_2(new) = w_2(old) + \alpha.t.x_1$$
$$= 0 + 1 \times (-1) \times 1$$
$$= -1$$
$$b(new) = -1$$

- **The new weights are (-1, -1, -1)**

COMPUTATIONS

- **Second input: (1,-1,1)**

$$y_{in} = b + x_1 w_1 + x_2 w_2 = -1 + 1 \times (-1) + (-1 \times -1) = -1$$

$$(y = f(y_{in}) = -1) \neq (t = 1)$$

- So, **new weights are to be computed**

$$w_1(new) = w_1(old) + \alpha.t.x_1 \qquad w_2(new) = w_2(old) + \alpha.t.x_1$$

$$= (-1) + 1 \times 1 \times 1 \qquad = -1 + 1 \times 1 \times (-1)$$

$$= 0 \qquad = -2$$

$$b(new) = 0$$

- **The new weights are (0, -2, 0)**

COMPUTATIONS

- **Third input:** (-1,1,-1)

$$y_{in} = b + x_1 w_1 + x_2 w_2 = 0 + (-1) \times 0 + (1 \times -2) = -2$$

$$(y = f(y_{in}) = -1) = (t = -1)$$

- **Weight updation is not necessary**
- So, new weights are not to be computed
- **Fourth input:** (-1, -1, -1)

$$y_{in} = b + x_1 w_1 + x_2 w_2 = 0 + (-1) \times 0 + (-1 \times -2) = 2$$

$$(y = f(y_{in}) = 1) \neq (t = -1)$$

- So, **new weights are to be computed**

COMPUTATIONS

- s

$$w_1(new) = w_1(old) + \alpha.t.x_1$$
$$= 0 + 1 \times (-1) \times (-1)$$
$$= 1$$

$$w_2(new) = w_2(old) + \alpha.t.x_1$$
$$= -2 + 1 \times (-1) \times (-1)$$
$$= -1$$

$$b(new) = -1$$

- **The new weights are (-1, -1, -1)**

FINAL ANALYSIS

	Input		Target	Net Input	Calculated output	Upda ted	weights	
x1	x2	b	t	yin	y	w1	w2	b
1	1	1	-1	0	0	-1	-1	-1
1	-1	1	1	-1	-1	0	-2	0
-1	1	1	-1	-2	-1	0	-2	0
-1	-1	1	-1	2	1	1	-1	-1

EXAMPLE-6

- Find the weights required to perform classification using perceptron network.
- The vectors (1, 1, 1, 1) and (-1, 1, -1, -1) are belonging to the class (so have target value 1),
- vectors (1, 1, 1,-1) and (1, -1, -1, 1) are not belonging to the class (so we have target value -1).
- Assume learning rate as 1 and initial weight as 0.

INITIAL TABLE

- The truth table is given by

x1	x2	x3	x4	b	t
1	1	1	1	1	1
-1	1	-1	-1	1	1
1	1	1	-1	1	-1
1	-1	-1	1	1	-1

COMPUTATIONS

- Here we take $w_1 = w_2 = w_3 = w_4 = 1, \theta = 0$. Also, $\alpha = 1$
- The activation function is given by
- $$y = \begin{cases} 1, & if \;\; y_{in} > 0; \\ 0, & if \;\; y_{in} = 0; \\ -1, & if \;\; y_{in} < 0. \end{cases}$$
- The net input is given by

$$y_{in} = b + x_1 w_1 + x_2 w_2 + x_3 w_3 + x_4 w_4$$

- The next table reflects the training performed with weights computed

COMPUTATIONS

	Input				Target	Net Input	Out put	Up	da	ted	Weig	hts
x1	EPOCH-1 x2	x 3	x4	b	t	Y in	y	w1	w2	w3	w4	b
1	1	1	1	1	1	0	0	1	1	1	1	1
-1	1	-1	1	1	1	-1	-1	0	2	0	0	2
1	1	1	-1	1	-1	4	1	-1	1	-1	1	1
1	-1	-1	1	1	-1	1	1	-2	2	0	0	0

COMPUTATIONS

	Input				Target	Net Input	Out put	Up	da	ted	Weig	hts
x1	EPOCH-2 x2	x 3	x4	b	t	Y in	y	w1	w2	w3	w4	b
1	1	1	1	1	1	0	0	-1	3	1	1	1
-1	1	-1	1	1	1	3	1	-1	3	1	1	1
1	1	1	-1	1	-1	4	1	-2	2	0	2	0
1	-1	-1	1	1	-1	-2	-1	-2	2	0	2	0

COMPUTATIONS

	Input				Target	Net Input	Out put	Up	da	ted	Weig	hts
x1	EPOCH-3 x2	x 3	x4	b	t	Y in	y	w1	w2	w3	w4	b
1	1	1	1	1	1	2	1	-2	2	0	2	0
-1	1	-1	1	1	1	2	1	-2	2	0	2	0
1	1	1	-1	1	-1	-2	-1	-2	2	0	2	0
1	-1	-1	1	1	-1	-2	-1	-2	2	0	2	0

Here the target outputs are equal to the actual outputs. So, we stop.

THE FINAL NET

- s