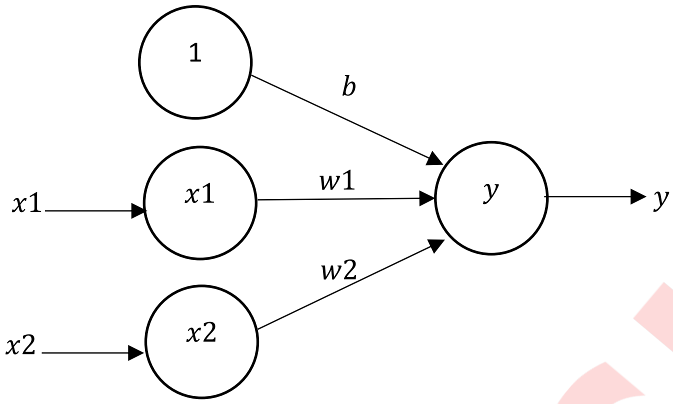
Artificial Intelligence

1.)Implement AND function using perceptron networks for bipolar inputs and targets. (Initial values are w1=w2=b=0, learning rate=1, threshold=0)
Using the linear separability concept, obtain the positive and negative response.
If threshold value is not 0, how to match the output responses? Show graphically.
Answer:



Truth Table

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

$$f(yin) = \begin{cases} 1 \text{ if } Yin > \theta \ (\theta=0) \\ 0 \text{ if } Yin = \theta \ (\theta=0) \\ -1 \text{ if } Yin \leq \theta \ (\theta=0) \end{cases}$$

EPOCH 1

Input			Target	Net input	Calculated output	Weight Changes			Weights		
x1	x2	1	(t)	(Yin)	(y)	Δw1	Δw2	Δb	w1 0	w2 0	b 0
1	1	1	1	0	0	1	1	1	1	1	1
1	-1	1	-1	1	1	-1	1	-1	0	2	0
-1	1	1	-1	2	2	1	-1	-1	1	1	-1
-1	-1	1	-1	-3	-3	0	0	0	1	1	-1

EPOCH 2

Input			Target	Net input	Calculated output	Weight Changes			Weights		
x1	x2	1	(t)	(Yin)	(y)	Δw1	Δw2	Δb	w1 1	w2 1	b -1
1	1	1	1	1	1	0	0	0	1	1	-1

1	-1	1	-1	-1	-1	0	0	0	1	1	-1
-1	1	1	-1	-1	-1	0	0	0	1	1	-1
-1	-1	1	-1	-3	-1	0	0	0	1	1	-1

θ=0

The equation of separating line is

$$x2 = \left(\frac{-w1}{w2}\right)x1 - \left(\frac{b}{w2}\right)$$

Here,

$$w1x1 + w2x2 + b > \theta$$

$$w1x1 + w2x2 + b > 0$$

Thus, using final weights we obtain

$$x2 = \left(-\frac{1}{1}\right)x1 - \left(-\frac{1}{1}\right)$$

$$\therefore x1 + x2 = 1$$

Graphical representation:

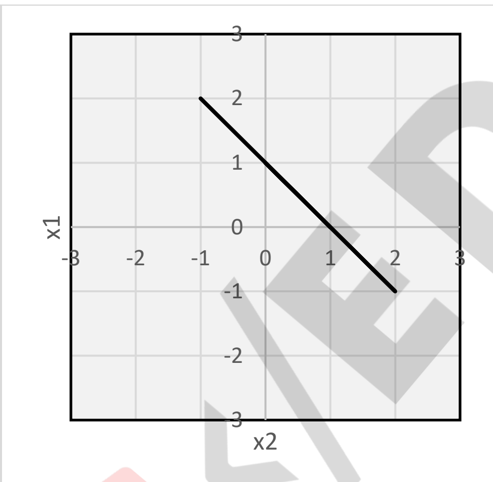
i. Put x1=0 in x1+x2=0

$\therefore$x2=1

ii. Put x2=0 in x1+x2=0

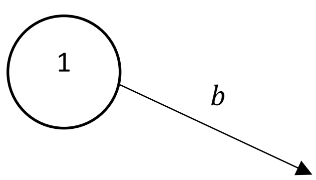
$\therefore$x1=1

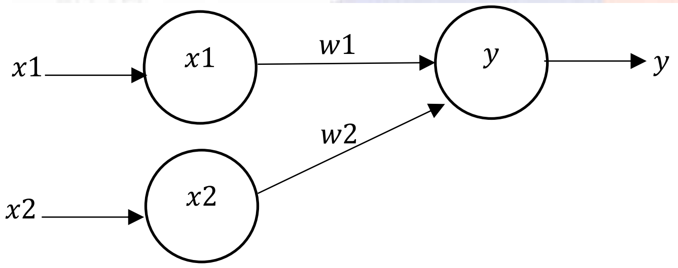
Hence the points are (0,1) & (1,0)



2.) Implement OR function using Perceptron networks for binary inputs and bipolar targets. (Initial values are w1=w2=b=0 , learning rate=1, threshold=0)

Answer:





Truth Table

x1	x2	t
1	1	1
1	0	1
0	1	1
0	0	-1

f(yin)= { 1 if Yin > θ (θ=0)
0 if Yin ≤ θ (θ=0)

EPOCH 1

Input			Target	Net	Calculated	Weight Changes			Weights		
x1	x2	1	(t)	input (Yin)	output (y)	Δw1	Δw2	Δb	w1 0	w2 0	b 0
1	1	1	1	0	0	1	1	1	1	1	1
1	0	1	1	2	1	0	0	0	1	1	1
0	1	1	1	2	1	0	0	0	1	1	1
0	0	1	-1	1	1	0	0	-1	1	1	0

EPOCH 2

Input			Target	Net	Calculated	Weight Changes			Weights		
x1	x2	1	(t)	input (Yin)	output (y)	Δw1	Δw2	Δb	w1 1	w2 1	B 0
1	1	1	1	2	1	0	0	0	1	1	0
1	0	1	1	1	1	0	0	0	1	1	0
0	1	1	1	1	1	0	0	0	1	1	0
0	0	1	-1	0	0	0	0	-1	1	1	-1

EPOCH 3

Input			Target	Net	Calculated	Weight Changes			Weights		
x1	x2	1		input	output	Δw1	Δw2	Δb	w1	w2	B

			(t)	(Yin)	(y)				1	1	-1
1	1	1	1	2	1	0	0	0	1	1	0
1	0	1	1	1	1	0	0	0	1	1	0
0	1	1	1	1	1	0	0	0	1	1	0
0	0	1	-1	0	-1	0	0	-1	1	1	0

3.) Construct a Maxnet with four neurons and inhibitory weight €=0.2 given the initial activations as follows:
a1 (0)=0.3, a2(0)=0.5, a3(0)=0.7, a4(0)=0.9
Answer:
Update the activation for each node i.e.
aj (new) = f [aj(old) – € ∑ ak(old)]
k≠j
The activation function is given by,
f(x) = x ,if x>0
= 0 , if x<=0

<u>First Iteration-</u>
a1(0)=0.3
a2(0)=0.5
a3(0)=0.7
a4(0)=0.9
a1 (1) = f [a1(0) – € ∑ ak(0)]
k≠j
=f [0.3-0.2*(0.5+0.7+0.9)]
=f [0.3-0.42]
=f [-0.21]
a1 (1) =0

a2 (1) = f [a2(0) – € ∑ ak(0)]
k≠j
=f [0.5-0.2*(0.3+0.7+0.9)]
=f [0.5-0.38]
=f [0.12]
a2 (1) =0.12

$\varepsilon = 0.2,$

$a_1(0) = 0.3,\ a_2(0) = 0.5,\ a_3(0) = 0.7,\ a_4(0) = 0.9$

a3 (1) = f [a3(0) – € ∑ ak(0)]
k≠j
=f [0.7-0.2*(0.5+0.3+0.9)]
=f [0.7-0.34]
=f [0.36]
a3 (1) =0.36

a4 (1) = f [a4(0) – € ∑ ak(0)]
k≠j
=f [0.9-0.2*(0.5+0.7+0.3)]
=f [0.9-0.3]
=f [0.6]
a4 (1) =0.6

Second Iteration-

a1(1)=0
a2(1)=0.12
a3(1)=0.36
a4(1)=0.6

a1 (2) = f [a1(1) – € ∑ ak(1)]
k≠j
=f [0 -0.2*(0.12+0.36+0.6)]
=f [0-0.216]
=f [-0.216]
a1 (2) =0

a2 (2) = f [a2(1) – € ∑ ak(1)]
k≠j
=f [0.12-0.2*(0+0.36+0.6)]
=f [0.12-0.192]
=f [-0.072]
a2 (2) =0

a3 (2) = f [a3(1) – € ∑ ak(1)]
k≠j
=f [0.36-0.2*(0.12+0+0.6)]
=f [0.36-0.144]
=f [0.216]
a3 (2) =0.216

a4 (2) = f [a4(1) – € ∑ ak(1)]
k≠j
=f [0.6-0.2*(0.12+0.36+0)]
=f [0.6-0.096]
=f [0.504]
a4 (2) =0.504

Third Iteration-

a1(2)=0
a2(2)=0
a3(2)=0.216

a4(2)=0.504

a1 (3) = f [a1(2) – € ∑ ak(2)]
k≠j
=f [0 -0.2*(0+0.216+0.504)]
=f [0-0.144]
=f [-0.144]

a1 (3) =0

a2 (3) = f [a2(2) – € ∑ ak(2)]
k≠j
=f [0-0.2*(0+0.216+0.504)]
=f [0-0.144]
=f [-0.144]

a2 (3) =0

a3 (3) = f [a3(2) – € ∑ ak(2)]
k≠j
=f [0.216-0.2*(0+0+0.504)]
=f [0.216-0.1008]
=f [0.1152]

a3 (3) =0.1152

a4 (3) = f [a4(2) – € ∑ ak(2)]
k≠j
=f [0.504-0.2*(0+0.216+0)]
=f [0.504-0.0432]
=f [0.4608]

a4 (3) =0.4608

<u>Fourth Iteration-</u>

a1(3)=0
a2(3)=0
a3(3)=0.1152
a4(3)=0.4608

a1 (4) = f [a1(3) – € ∑ ak(3)]
k≠j
=f [0 -0.2*(0+0.1152+0.4608)]
=f [0-0.1152]
=f [-0.1152]

a1 (4) =0

a2 (4) = f [a2(3) – € ∑ ak(3)]
k≠j
= f [0 -0.2*(0+0.1152+0.4608)]

=f [0-0.1152]
=f [-0.1152]

a2 (4) =0

a3 (4) = f [a3(3) – € ∑ ak(3)]
k≠j
=f [0.1152-0.2*(0+0+0.4608)]
=f [0.36-0.09216]
=f [0.02304]

a3 (4) =0.02304

a4 (4) = f [a4(3) – € ∑ ak(3)]
k≠j
=f [0.4608-0.2*(0+0.1152+0)]
=f [0.4608-0.02304]
=f [0.43776]

a4 (4) =0.43776

<u>Fifth Iteration-</u>

a1(4)=0
a2(4)=0
a3(4)=0.02304
a4(4)=0.43776

a1 (5) = f [a1(4) – € ∑ ak(4)]
k≠j
=f [0 -0.2*(0+0.02304+0.43776)]
=f [0-0.09216]
=f [-0.09216]

a1 (5) =0

a2 (5) = f [a2(4) – € ∑ ak(4)]
k≠j
=f [0 -0.2*(0+0.02304+0.43776)]
=f [0-0.09216]
=f [-0.09216]

a2 (5) =0

a3 (5) = f [a3(4) – € ∑ ak(4)]
k≠j
=f [0.02304 -0.2*(0+0+0.43776)]
=f [0.02304-0.087552]
=f [-0.064512]

a3 (5) =0

$$a4(5) = f\left[a4(4) - € \sum_{k \neq j} ak(4)\right]$$

=f [0.43776-0.2*(0+0+0.02304)]

=f [0.43776-0.004608]

=f [0.433152]

a4 (5) =0.433152

4.) Comparison between Biological Neuron and Artificial Neuron

i) Size and Complexity

- Biological Neural Network has 10 billion neuron
- Artificial Neural Network has 100 – 10000 neurons

ii) Style of Computation

- Biological Neural Networks communicate through pulses
- Artificial Neural Network are based on computational model involving the propagation of continuous variable from one processing unit to the next

iii) Processing Speed

- Biological Neural Network is slow as neurons needs several milliseconds to react to stimulus
- Artificial Neural Network is fast this can be achieved by switching between them

iv) Information Storage

- Biological Neural Network stores information at the synapses
- Artificial Neural Network stores information at the weight matrix

v) Fault Tolerance

- Biological Neural Network possesses fault tolerance capability this mean that partial recovery from damage is possible if health units take over the functions previously carried out by the damaged areas
- Artificial Neural Network has no fault tolerance. The information gets corrupted if the new interconnection are disconnected

5.) Soft Computing VS Hard Computing

Hard Computing

1. Require a precisely stated analytical model and often a lot of computation time
2. Based on Binary logic, crisp system, numerical analysis and crisp s/w
3. Require exact in data to solve a particular problem
4. Allows strictly sequential computation
5. Produces precise answer
6. Required program to be Written

Soft Computing

1. It is tolerant of imprecision, uncertainty, partial truth and approximation In effect the role mode: human mind
2. Based on Fuzzy logic, neural network and probabilities reasoning
3. Can deal with ambiguous and noisy data
4. Allow parallel computations
5. Can yield approx. answers
6. They are model free, they can evolve their own models and program

6.a) Water Jug Problem

In the water jug problem, we are provided with two jugs: one having the capacity to hold 3 liter of water and the other has the capacity to hold 4 liter of water.

There is no other measuring equipment available and the jugs also do not have any kind of marking on them.

So, the agent's task here is to fill the 4-liter jug with 2 liter of water by using only these two jugs and no other material. Initially, both our jugs are empty.

So, to solve this problem, following set of rules were proposed:

If x<4.

(x,y)->(x,3) fill the 3 liter jug

If x<3

(x,y)->(x-d,y) pour some water out of the 4-liter jug.

If x>0

(x,y)->(x-d,y) pour some water out of the 3-liter jug.

If y>0

(x,y)->(0,y) empty the 4-liter jug on the ground

If x>0

(x,y)->(x,0) empty the 3-liter jug on the ground

If y>0

(x,y)->(4,y-(4-x)) pour water from the 3-liter jug into the 4-liter

If x+y>=4 and y>0 jug until the 4-liter jug is full

(x,y)->(x-(3-y),3)) pour water from the 4-liter jug into the 3-liter

If x+y>=3 and x>0 jug until the 3-liter jug is full.

(x,y)->(x+y,0) pour all the water from the 3-liter jug into

If x+y<=4 and y>0 the 3-liter jug.

(x,y)->(0,x+y) pour all the water from the 4-liter jug into

If x+y<=3 and x>0 the 3-liter jug.

1. (0,2)->(2,0) pour the 2-liter from the 3-liter jug into the 4-liter jug.
2. (2,y)->(0,x) empty the 2 liter in the 4 liter on the ground.

Solution of Water Jug Problem

(0,0) (0,3) (3,0) (3,3) (4,2) (0,2) (2,0)

6.b) Chess Problem

i. In a chess game problem, the start state is the initial configuration of chessboard.
ii. The final or goal state is any board configuration, which is a winning position for any player (clearly, there may be multiple final positions and each board configuration can be thought of as representing a state of the game).
iii. Whenever any player moves any piece, it leads to different state of game.
iv. It is estimated that the chess game has more than 10^120 possible states.
v. The game playing would mean finding (or searching) a sequence of valid moves which bring the board from start state to any of the possible final states.

6.c) 8 – Puzzle Problem

i. The 8-puzzle problem belongs to the category of "sliding-block puzzle" types of problems.
ii. It is described as follows: "It has set of a 3x3 board having 9 block spaces out of which, 8 blocks are having tiles bearing number from 1 to 8. One space is left blank. The tile adjacent to blank space can move into it. We have to arrange the tiles in a sequence."

1		4
6	5	8
2	3	7

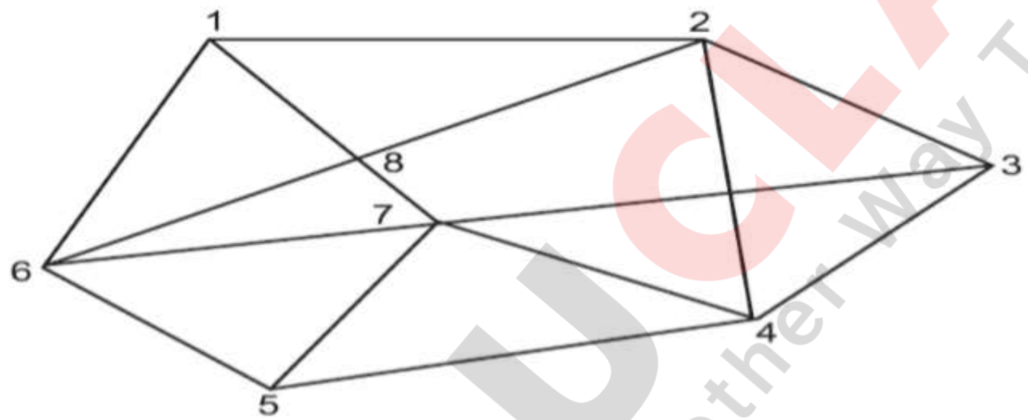
1	2	3
4	5	6
7	8	

iii. Here the data structure to represent the states can be 9-element vector indicating the tiles in each board position.
iv. Hence, a starting state corresponding to above configuration will be {1, blank, 4, 6, 5, 8, 2, 3, 7} (there can be various different start positions).
v. The goal state is {1,2,3,4,5,6,7,8, blank}.
vi. Here, the possible movement outcomes after applying a move can be many.
vii. They are represented as tree.

viii. This tree is called states pace tree.
ix. The depth of the tree will depend upon the number of steps in the solution.

6.d) Traveling Salesperson Problem

i. This problem falls in the category of path finding problems.
ii. The problem is defined as follows: “Given ‘n’ cities connected by roads, and distances between each pair of cities. A sales person is required to travel each of the cities exactly once. We are required to find the route of salesperson so that by covering minimum distance, he can travel all the cities and come back to the city from where the journey was started”.



6.e) Missionaries and Cannibals

The problem is stated as follows:
“Three missionaries and three cannibals are present at one side of a river and need to cross the river. There is only one boat available. At any point of time, the number of cannibals should not outnumber the number of missionaries at that bank. It is also known that only two persons can occupy the boat available at a time.”

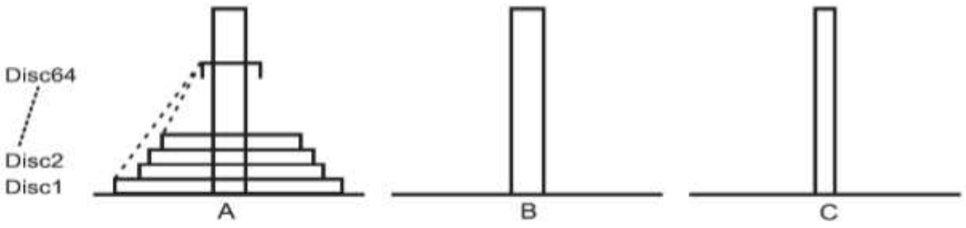
i. The objective of the solution is to find the sequence of their transfer from one bank of river to other using the boat sailing through the river satisfying these bank of river to other using the boat sailing through the river satisfying these constraints.
ii. We can form various production rules as presented in water-jug problem.
iii. Let Missionary is denoted by ‘M’ and Cannibal, by ‘C’.
iv. These rules are described below

Rule 1: (0, M): One missionary sailing the boat from bank-1 to bank-2
Rule 2: (M, 0): One missionary sailing the boat from bank-2 to bank-1
Rule 3: (M, M): Two missionaries sailing the boat from bank-1 to bank-2
Rule 4: (M, M): Two missionaries sailing the boat from bank-2 to bank-1
Rule 5: (M, C): One missionary and one Cannibal sailing the boat from bank-1 to bank-2
Rule 6: (C, M): One missionary and one Cannibal sailing the boat from bank-2 to bank-1
Rule 7: (C, C): Two Cannibals sailing the boat from bank-1 to bank-2
Rule 8: (C, C): Two Cannibals sailing the boat from bank-2 to bank-1
Rule 9: (0, C): One Cannibal sailing the boat from bank-1 to bank-2
Rule 10: (C, 0): One Cannibal sailing the boat from bank-2 to bank-1

After application of rule	persons in the river bank-1	persons in the river bank-2	boat position
Start state	M, M, M, C, C, C	0	bank-1
5	M, M, C, C	M, C	bank-2
2	M, M, C, C, M	C	bank-1
7	M, M, M	C, C, C	bank-2
10	M, M, M, C	C, C	bank-1
3	M, C	C, C, M, M	bank-2
6	M, C, C, M	C, M	bank-1
3	C, C	C, M, M, M	bank-2
10	C, C, C	M, M, M	bank-1
7	C	M, M, M, C, C	bank-2
10	C, C	M, M, M, C	bank-1
7	0	M, M, M, C, C, C	bank-2

6.f) Tower of Hanoi Problem

Near the city of ‘Hanoi’, there is a monastery. There are three tall posts in the courtyard of the monastery.
One of these posts is having sixty-four disks, all having a hole in the center and are of different diameters, placed one over the other in such a way that always a smaller disk is placed over the bigger disk.
The monks of the monastery are busy in the task of shifting the disks from one post to some other, in such a way that at no point of time, a bigger disk is placed above smaller disk. Only one disk can be removed at a time. Moreover, at every point of time during the process, all other disks than the one removed, should be on one of the should be on one of the posts.
The third post can be used as a temporary resting place for the disks.
We have to help the monks in finding the easiest and quickest way to do so.



It consists of three pegs, and a number of disks (usually 60) of different sizes which can slide onto any peg.

The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.

The objective of the puzzle is to move the entire stack to another rod obeying the following rules

i. Only one disk must be moved at a time.
ii. Each move consists of taking the upper disk from one of the rod sand sliding it onto another rod, on top of the other disks that may already be present on that rod.
iii. No disk may be placed on top of a smaller disk.