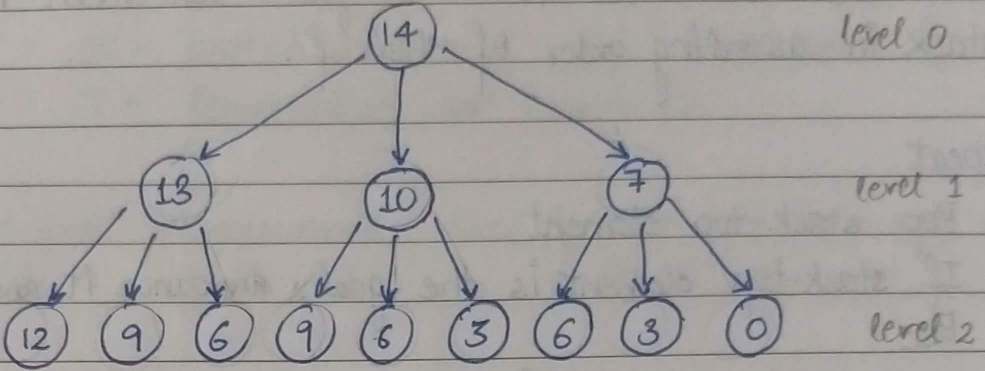
Artificial Intelligence (CSE241) - Mid-Term Assignment.

Name: Shruti T. Avhad Roll no.:- 20074030 Branch:- CSE (IDD)

Q1.

Ans-

a) According to the given information, the state space up to the node with label '0' will be:



b)

Breadth-First-Search is implemented using queue data structure.
It follows a FIFO (First-in-first-out) principle.
We explore a node and push its children into the queue, and similarly, explore the children node until the goal is reached.

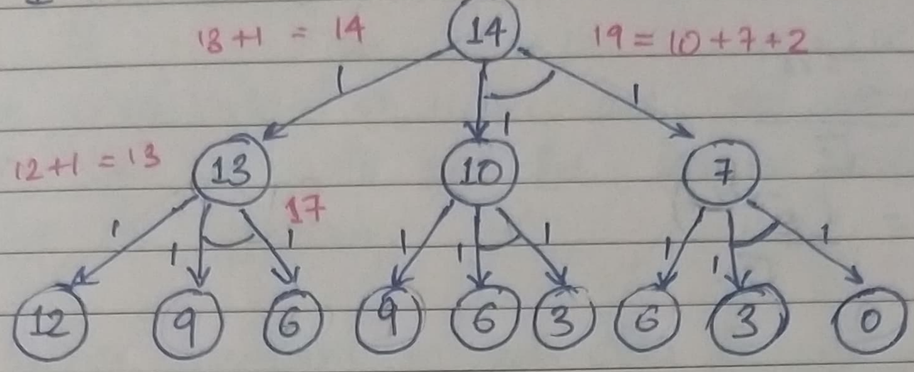
So, state-space traversal in the following manner:

Queue	Current node	Children node
14	14	13, 10, 7
13, 10, 7	13	12, 9, 6
10, 7, 12, 9, 6	10	9, 6, 3
7, 12, 9, 6, 9, 6, 3	7	6, 3, 0
12, 9, 6, 9, 6, 3	12	–

So, as we have reached the goal node '12', traversal stops.

∴ BFS traversal is : 14, 13, 10, 7, 12.

c) According to the ques, each non-leaf node has ~~hyper~~ 2^{nd} and 3^{rd} child has an AND hyperarc connection and each ~~node~~ edge cost is 1 :



Starting from 14, we will move to the left node as, $14 < 19$, then we will reach node 13.
We will move left node again as $13 < 17$, reaching 12 which is the last node.
Again moving up, the revised cost is again 13 and further again revised cost is 14 of the root node.

Hence,
The final label order of nodes explored is :

14, 13, 12.

Q2

Ans - **Algorithm for simulated Annealing**

1. Identify the possible starting states and measure the distance (f) of their closseness with the goal. Push them into a stack in ascending order of their f.

2. Repeat
   Pop stack-top element
   If stack-top element is the goal, announce it and exit.
   Else
   - Generate children of the stack-top element N and compute f for each of them.
   - If measure of f for at least one child is improving, push those children onto the stack in ascending order of their f.
   - If none of the children of N has a better f, then
   → Select any of them randomly and compute its p',
     p' is the probability of transition to the next state i.e lower to higher, where
     $p' = \exp\left(\frac{-\Delta E}{T}\right)$

     ΔE denotes a positive change in energy,
     T is the temperature at current thermal state.

   Then test whether p' exceeds a randomly generated number in the interval [0,1]. If yes, select the next state. If no, generate another alternative legal state and test in this way until one move can be selected.
   Replace stack top element by the selected move.
   → Reduce T slightly. If the reduced value is negative, set it to zero.

Repeat above steps until the stack is empty.
End.

For calculating probability of transition in simulate annealing.

$p' = e^{-\Delta E/T}$.

where ΔE = current state cost - next state cost = $f_c - f_n$

T = Temperature ~~use~~ parameter.

current Eval^n (f_c)	Neighbourhood Eval^n (f_n)	$\Delta E = f_c - f_n$	current Temp. (T)	$p' = \exp\left(\frac{-\Delta E}{T}\right)$
-2	-4	2	2	0.367
-2	-3	1	2	0.606
-3	-4	1	1.6	0.535
-3	-1	-2	1.6	($p'=1$) 3.490
-1	-2	1	1.2	0.434
-1	-3	2	1.2	0.188
-3	-1	-2	0.8	($p'=1$) 12.182
-1	-3	2	0.4	0.006
-1	-2	1	0.4	0.082
-1	0	-1	0.4	($p'=1$) 12.182

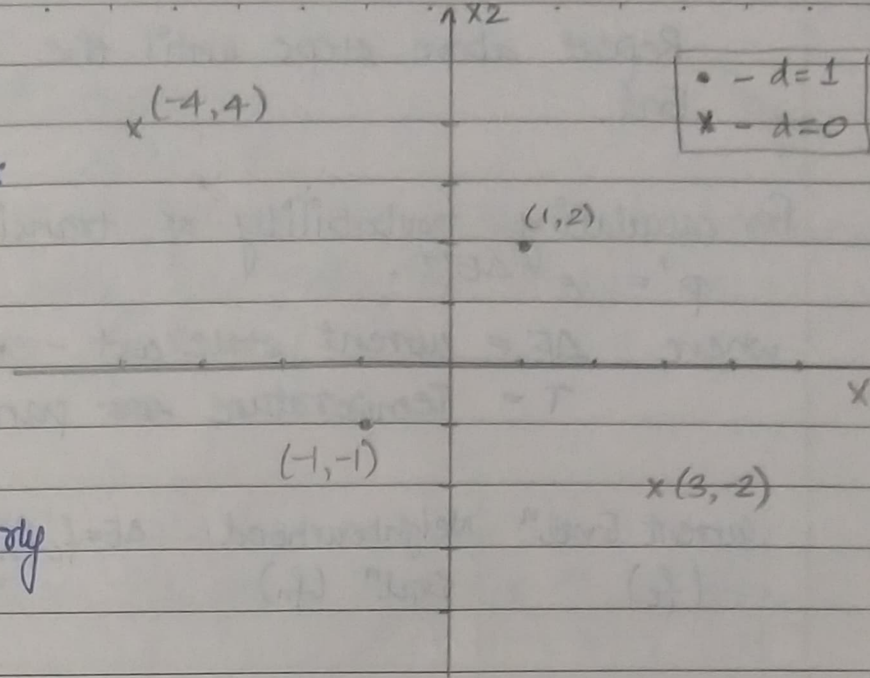
when $p' > 1$, it means its neighbour or next state is better than current state (with greater evaluation of objective function.)

$\therefore$ It will be selected as first priority (as in step 4) without calculating its p' we put it as $p' = 1$.

Q3.

Ans - From the given data distribution, there exists no such straight line which can accurately separate the two classes.

i.e. the data is not linearly separable.



(∵ for $X1 * X2 > 0$, $d=1$
⇒ data points for class $d=1$ will be in I^{st} and III^{rd} Quad.
and no straight line can classify them as a single class.)

Since, the data is not linearly separable, we can introduce two new features $X1' = (X1)^2$ and $X2' = X1 * X2$.

These two features will cause the equation to be a quadratic polynomial (a Hyperbolic equation in specific) which will effectively classify the given data in their respective classes.

∴ In SLP layer,

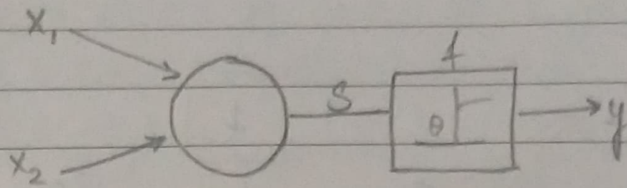
$$v = w_0 \cdot 1 + w_1 * X1 + w_2 * X2 + w_3 * X1' + w_4 X_2'$$

$$\boxed{v = w_0 + w_1 X1 + w_2 X2 + w_3 X1^2 + w_4 X1 * X2}$$

$$\boxed{\begin{aligned} X1' &= X1^2 \\ X2' &= X1 * X2 \end{aligned}}$$

Q4.

Ans-



McCulloch - Pitt's Neuron accepts binary input and gives binary output.
AND function also deals with binary inputs and outputs.

From this table, the threshold value θ in the step function should be 2 ie $\theta = 2$.

$d = X_1 \text{ AND } X_2$.

X_1	X_2	S	d
0	0	0	0
0	1	1	0
1	0	1	0
1	1	2	1

$$\therefore \quad y = f(s) = \begin{cases} 0, & s < 2 \\ 1, & s \geq 2. \end{cases}$$

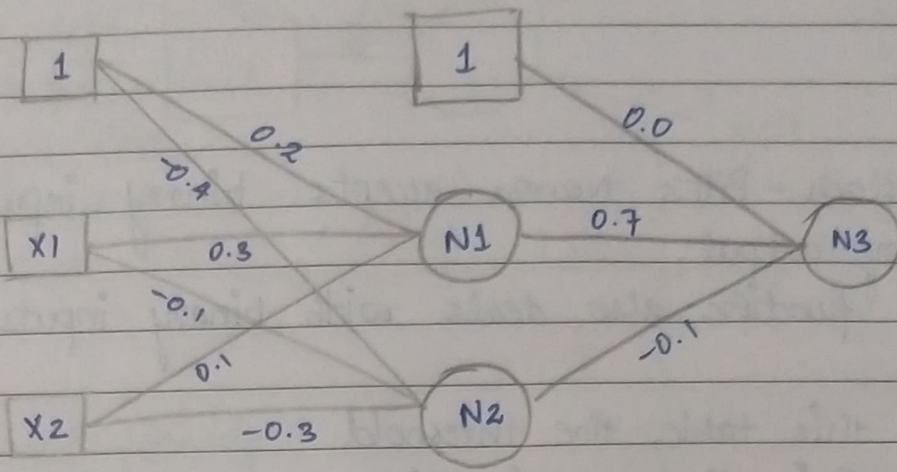
Main Differences between McCulloch-Pitt's Neuron and Perceptron model are:

1. MP Neuron accepts only boolean inputs whereas Perceptron model can process any real input.

2. In MP Neuron model, the input signals are not weighed while in perceptron model, the input signals are multiplied with their respective weights and then added.
These weights are also real numbers.

Q5.

Ans-



$X=1$

$X=1$

desired output $d=0$

w_{ij} - weight from input layer neuron j to hidden layer neuron i

Initial weights,	$w_{10} = 0.2$	$w_{11} = 0.3$	$w_{12} = 0.1$
	$w_{20} = -0.4$	$w_{21} = -0.1$	$w_{22} = -0.3$
	$w^{o}_{10} = 0.0$	$w^{o}_{11} = 0.7$	$w^{o}_{12} = -0.1$

w^{o} = hidden layer to output layer weights.

Activation function, $\phi(v) = \frac{1}{1+e^{-v}}$ Learning rate $n = 0.5$

Forward Propagation

First hidden layer,

$v_1 = w_{10} \cdot 1 + w_{11} \cdot x1 + w_{12} \cdot x2 = \sum_i w_{ij} \cdot x_j$

$v_1 = (0.2) \cdot 1 + (0.3) \cdot 1 + (0.1) \cdot 1$

$v_1 = 0.6$

$N_1 = \phi(v_1) = \frac{1}{1+e^{-0.6}} = \underline{\underline{0.6456}}$

$$v_2 = w_{20} \cdot 1 + w_{21} \cdot x_1 + w_{22} \cdot x_2 = \sum_j w_{2j} \cdot x_j$$

$$v_2 = (-0.4) \cdot 1 + (-0.1) \cdot 1 + (-0.3) \cdot 1$$

$$v_2 = \underline{-0.8}$$

$$\therefore N_2 = \phi(v_2) = \frac{1}{1 + e^{-(-0.8)}} \Rightarrow \underline{N_2 = 0.3100}$$

Output layer

$$v_3 = w^{o}_{10} \cdot 1 + w^{o}_{11} \cdot N_1 + w^{o}_{12} \cdot N_2$$

$$v_3 = (0.0) \cdot 1 + (0.7) \cdot (0.6456) + (-0.1) \cdot (0.3100).$$

$$v_3 = \underline{0.4209}$$

$$N_3 = \phi(v_3) = \frac{1}{1 + e^{-(0.4209)}} \Rightarrow \underline{N_3 = 0.6037}$$

$$\therefore y = N_3 = 0.6037 \quad ; \quad d = 0$$

error / cost $\quad J = \frac{1}{2}(d-y)^2$

$$J = \frac{1}{2}(0 - 0.6037)^2$$

$$\underline{J = 0.1822}$$

* Backpropagation

$$w(n+1) = w(n) + \Delta w(n)$$

$$\Delta w = \eta \cdot \delta \cdot x$$

Now, for input layer unit i to hidden layer unit j.

$\Delta w_{ji} = \eta \cdot \delta_j \cdot x_i$ where $\delta_j = \phi(v_j) \cdot (1 - \phi(v_j)) \cdot \sum_k w_{kj} \cdot \delta_k$

the weight change from hidden layer unit j to output layer unit k $\Delta w_{kj} = \eta \cdot \delta^o_k \cdot N_j$ where $\delta^o_k = (d_k - y)_k \cdot y_k \cdot (1 - y_k)$

∴ for output layer,

$$\delta^o_1 = (d - y) \cdot y \cdot (1 - y)$$

$\left[\delta^o_1\right.$ ← output layer; $\left. k=1 \because \text{only one output unit}\right]$

$$\delta^o_1 = (0 - 0.6037) \cdot 0.6037 \cdot (1 - 0.6037)$$

$$\boxed{\delta^o_1 = -0.1444}$$

∴ $\Delta w^o_{10} = \eta \cdot \delta^o_1 \cdot N_0 = 0.5 \cdot (-0.1444) \cdot 1 = -0.0722$

$\Delta w^o_{11} = \eta \cdot \delta^o_1 \cdot N_1 = 0.5 \cdot (-0.1444) \cdot 0.6456 = -0.0466$

$\Delta w^o_{12} = \eta \cdot \delta^o_1 \cdot N_2 = 0.5 \cdot (-0.1444) \cdot 0.3100 = -0.0223$

for hidden layer,

$$\delta_1 = \phi(v_1) \cdot (1 - \phi(v_1)) \cdot \sum_k w_k \cdot \delta_k$$

$$= N_1 \cdot (1 - N_1) \cdot w^o_{11} \cdot \delta^o_1 \qquad (\phi(v_1) = N_1)$$

$$= (0.6456)(1 - 0.6456)(0.7)(-0.1444) \qquad (k=1 \because \text{only 1 output unit})$$

$$= -0.023$$

$$\delta_2 = \phi(v_2)\cdot(1-\phi(v_2))\cdot\sum_K w_K\cdot\delta_K$$
$$= N_2\cdot(1-N_2)\cdot w^{o}_{12}\cdot\delta^{o}_1$$
$$= (0.31)(1-0.31)\cdot(0.1)\cdot(-0.1444)$$
$$= \underline{0.0031}$$

$$\therefore\ \Delta w_{10} = \eta\cdot\delta_1\cdot X0 = 0.5(-0.0231)\cdot 1 = \underline{-0.0115}$$
$$\Delta w_{11} = \eta\cdot\delta_1\cdot X1 = 0.5\cdot(-0.0231)\cdot 1 = \underline{-0.0115}$$
$$\Delta w_{12} = \eta\cdot\delta_1\cdot X2 = 0.5\cdot(-0.0231)\cdot 1 = \underline{-0.0115}$$

$$\Delta w_{20} = \eta\cdot\delta_2\cdot X0 = 0.5\cdot(0.0031)\cdot 1 = \underline{0.0015}$$
$$\Delta w_{21} = \eta\cdot\delta_2\cdot X1 = 0.5\cdot(0.0031)\cdot 1 = \underline{0.0015}$$
$$\Delta w_{22} = \eta\cdot\delta_2\cdot X2 = 0.5\cdot(0.0031)\cdot 1 = \underline{0.0015}$$

$\therefore$ weight update

$$w_{10} = w_{10(old)} + \Delta w_{10} = 0.2 + (-0.0115)$$
$$= 0.1885 \approx \underline{0.18}$$

$$w_{11} = w_{11(old)} + \Delta w_{11} = 0.3 - 0.0115 = \underline{0.2885} \approx \underline{0.28}$$
$$w_{12} = w_{12(old)} + \Delta w_{12} = 0.1 - 0.0115 = 0.0885 \approx \underline{0.08}$$
$$w_{20} = w_{20(old)} + w_{20} = -0.4 + 0.0015 = -0.3985 \approx \underline{-0.39}$$

$$w_{21} = w_{21(old)} + \Delta w_{21} = -0.1 + 0.0015 = -0.0985 \approx \underline{-0.09}$$
$$w_{22} = w_{22(old)} + \Delta w_{22} = -0.3 + 0.0015 = -0.2985 \approx \underline{-0.29}$$

$$w^{o}_{10} = w^{o}_{10(old)} + \Delta w^{o}_{10} = 0.0 - 0.0722 = -0.0722 = \underline{-0.07}$$
$$w^{o}_{11} = w^{o}_{11(old)} + \Delta w^{o}_{11} = 0.7 - 0.0466 = 0.6534 = \underline{0.65}$$
$$w^{o}_{12} = w^{o}_{12(old)} + \Delta w^{o}_{12} = -0.1 - 0.0223 = -0.1223 = \underline{-0.12}$$

Smriti Ahad

Ans :

$$\begin{array}{cc} & \begin{array}{ccc} w_{j0} & w_{j1} & w_{j2} \end{array} \\ W^{l} = & \begin{bmatrix} 0.18 & 0.28 & 0.08 \\ -0.39 & -0.09 & -0.029 \end{bmatrix} \end{array}$$

Input → Hidden layer weight

$$\begin{array}{cc} & \begin{array}{ccc} w^{o}_{10} & w^{o}_{11} & w^{o}_{12} \end{array} \\ W^{o} = & \begin{bmatrix} -0.07 & 0.65 & -0.12 \end{bmatrix} \end{array}$$

Hidden → Output layer weight.