Coochbehar Government Engineering College
Department: Electronics And Communication
Semester:7th
Name: Aritra Sharma
Roll no:34900321011
Subject: Neural Network and Fuzzy Logic Control
Subject Code:PE-EC702C

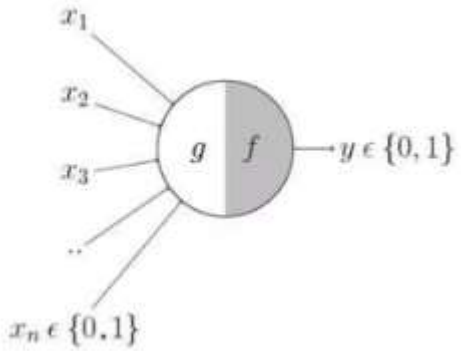
TOPIC:: MCCULLOCH PITS NEURON, HEBB NET WITH EXAMPLE

Discussion Topics

- ❑ McCulloch-Pitts Neuron
- ❑ Linear Separability
- ❑ Hebb Network

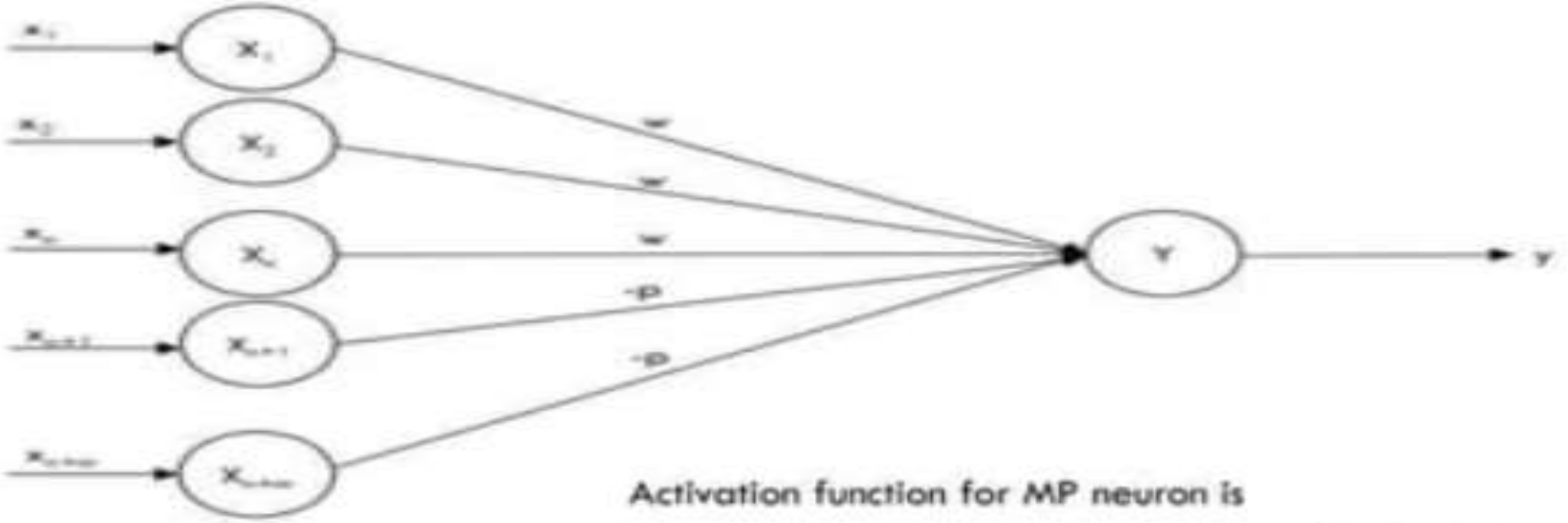
McCulloch-Pitts Neuron

- Mankind's First Mathematical Model Of A Biological Neuron.
- The very first step towards the perceptron used today was taken in 1943 by McCulloch and Pitts.
- The first computational model of a neuron was proposed by Warren McCulloch (neuroscientist) and Walter Pitts (logician) in 1943.



- It may be divided into 2 parts - the first part, g takes an input, performs an aggregation and based on the aggregated value the second part, f makes a decision.

Architecture



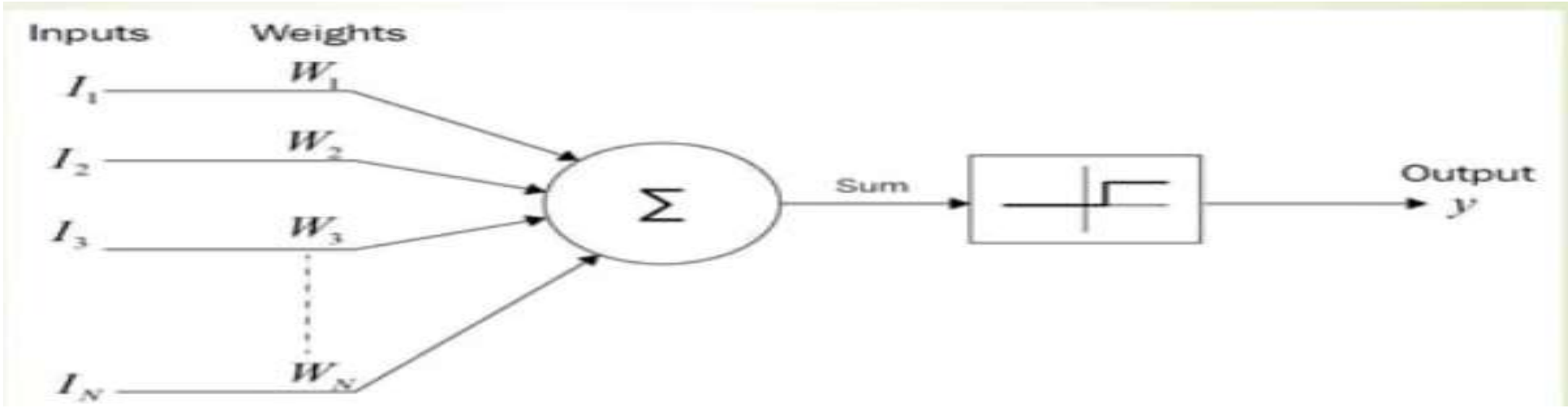
Activation function for MP neuron is

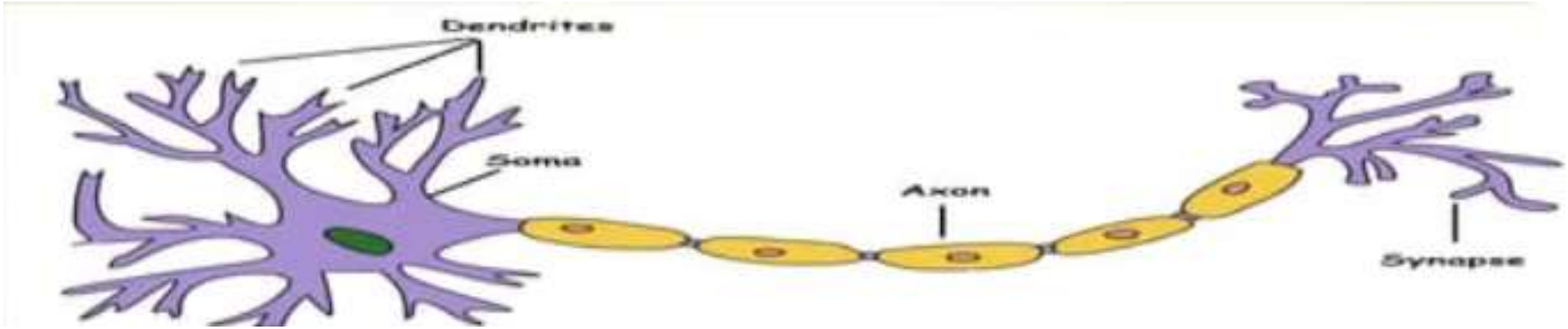
$$f(y) = \begin{cases} 1 & if\ y \geq \theta \\ 0 & if\ y < \theta \end{cases}$$

And

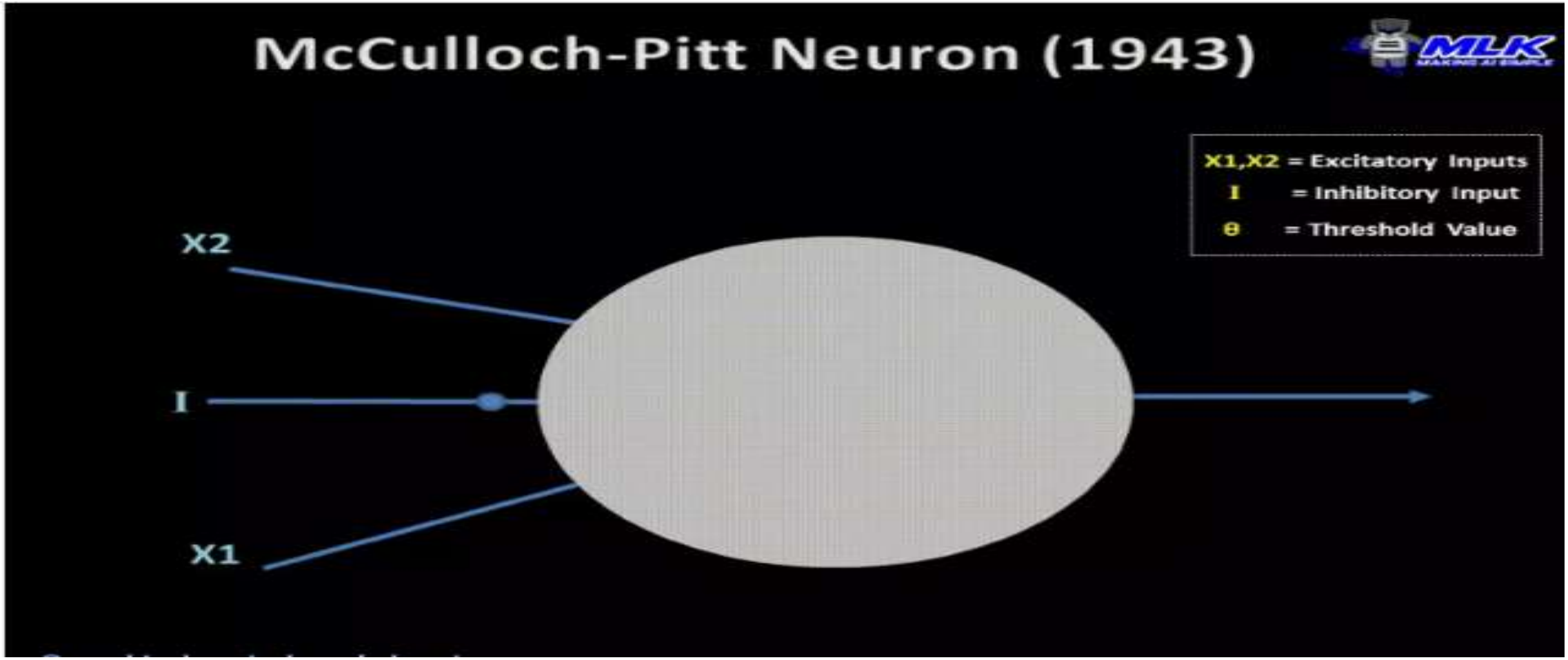
$$\theta > nw - p$$

Schematic


Inputs
Weights
I_1
I_2
I_3
I_N
W_1
W_2
W_3
W_N
Σ
Sum
Output
y


Dendrites
Soma
Axon
Synapse

McCulloch-Pitts Neuron



McCulloch-Pitts Neuron - Example

Example:

Lets suppose in order to predict one's own decision, whether to watch a random football game or not on TV. The inputs are all boolean i.e., {0,1} and output variable is also boolean {0: Will watch it, 1: Won't watch it}.

- So, x_1 could be is PremierLeagueOn (I like Premier League more)
- x_2 could be isItAFriendlyGame (I tend to care less about the friendlies)
- x_3 could be isNotHome (Can't watch it when I'm running errands. Can I?)
- x_4 could be isMan UnitedPlaying (I am a big Man United fan. GGMU!) and so on.
- Inputs can either be excitatory or inhibitory.
- Inhibitory inputs are those that have maximum effect on the decision making irrespective of other inputs i.e., if x_3 is 1 (not home) then the output will always be 0 i.e., the neuron will never fire, so x_3 is an inhibitory input.
- Excitatory inputs are NOT the ones that will make the neuron fire on their own but they might fire it when combined together.

McCulloch-Pitts Neuron – Example

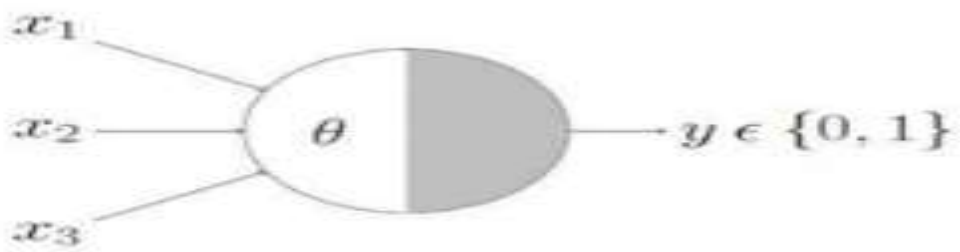
$$g(x_1, x_2, x_3, ..., x_n) = g(\mathbf{x}) = \sum_{i=1}^{n} x_i$$

$$y = f(g(\mathbf{x})) = 1 \quad if \quad g(\mathbf{x}) \geq \theta$$
$$= 0 \quad if \quad g(\mathbf{x}) < \theta$$

- It can be seen that g(x) is just doing a sum of the inputs a simple aggregation.
- And theta here is called thresholding parameter.
- For example, if the person always watch the game when the sum turns out to be 2 or more, the theta is 2 here. This is called the thresholding logic.

Boolean Function Using M-P Neuron

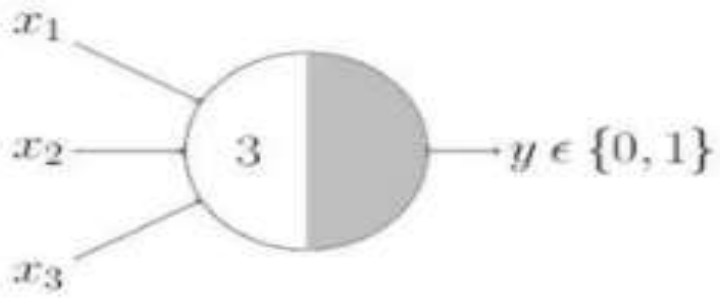
- Let's see how this neuron can be used to represent a few boolean functions.
- Inputs are all boolean and the output is also boolean so essentially, the neuron is just trying to learn a boolean function.
- M-P Neuron: A Concise Representation



- This representation just denotes that, for the boolean inputs x_1, x_2 and x_3 if the g(x) i.e., sum ≥ theta, the neuron will fire otherwise, it won't.

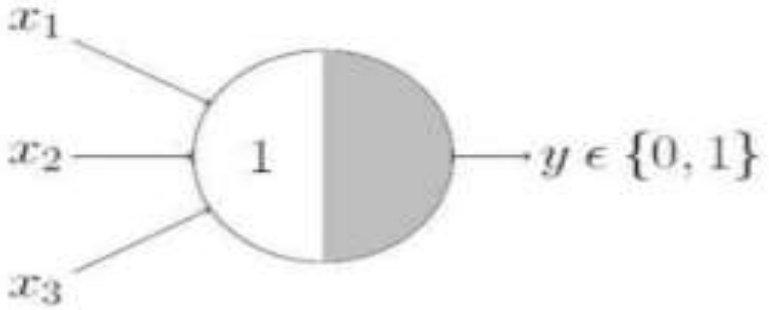
Boolean Function Using M-P Neuron

AND Function:



- An AND function neuron would only fire when ALL the inputs are ON i.e., $g(x) \geq 3$ here.

OR Function:

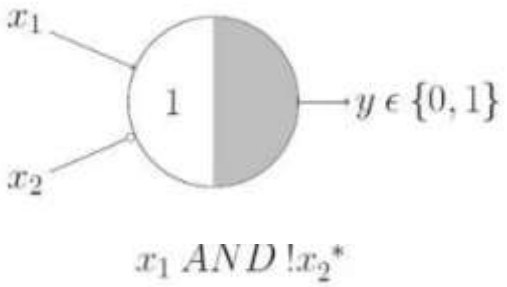


- An OR function neuron would fire if ANY of the inputs is ON i.e., $\boldsymbol{g(x) \geq 1}$ here.

Boolean Function Using M-P Neuron

A Function With An Inhibitory Input:

- Given an inhibitory input i.e., x_2 so whenever x_2 is 1, the output will be 0.
- Keeping that in mind, we know that x_1 AND !x_2 would output 1 only when x_1 is 1 and x_2 is 0 so it is obvious that the threshold parameter should be 1.

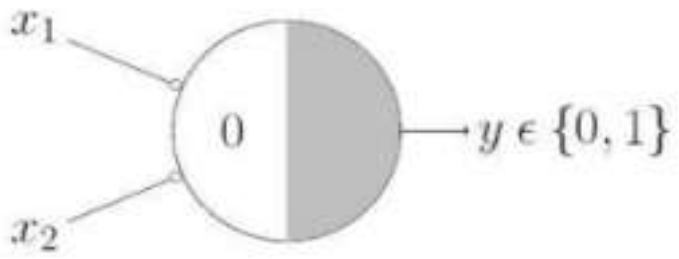


$x_1\ AND\ !x_2^*$

- Lets verify that, the g(x) i.e., x_1 + x_2 would be ≥ 1 in only 3 c
  - Case 1: when x_1 is 1 and x_2 is 0
  - Case 2: when x_1 is 1 and x_2 is 1
  - Case 3: when x_1 is 0 and x_2 is 1
- But in both Case 2 and Case 3, we know that the output will be 0 because x_2 is 1 in both of them and also x_1 _1 AND !x_2 will output 1 for Case 1 so the thresholding parameter holds good for the given function.

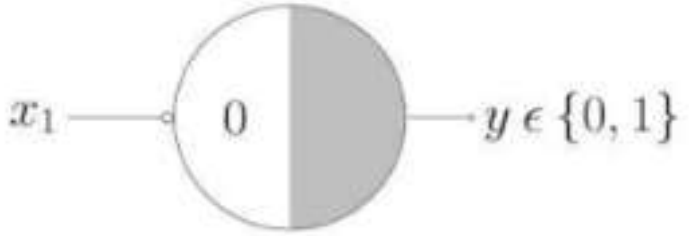
Boolean Function Using M-P Neuron

NOR Function:



- For a NOR neuron to fire, we want ALL the inputs to be 0 so the thresholding parameter should also be 0 and we take them all as inhibitory input.

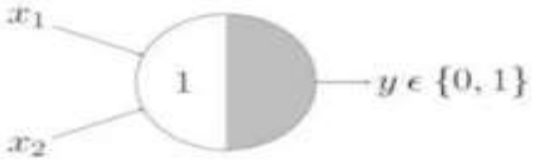
NOT Function:



- For a NOT neuron, 1 outputs 0 and 0 outputs 1. So we take the input as an inhibitory input and set the thresholding parameter to 0. It works!

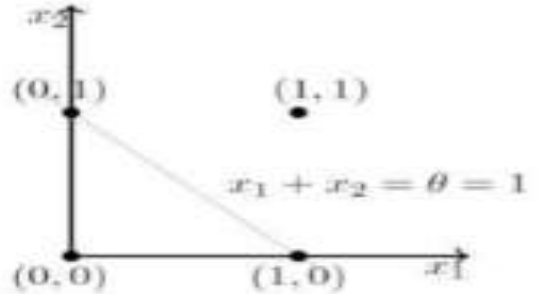
Geometric Interpretation Of M-P Neuron

OR Function:

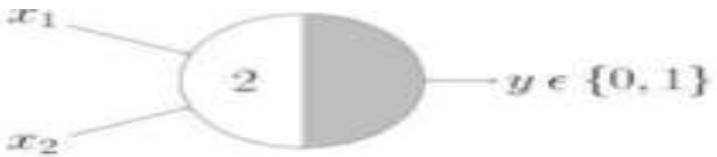


OR function

$$x_1 + x_2 = \sum_{i=1}^{2} x_i \geq 1$$

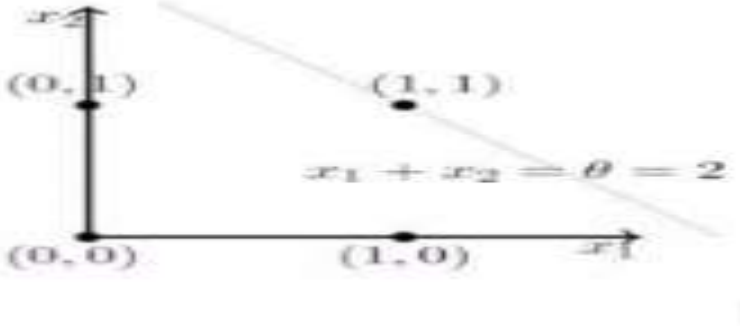


AND Function:



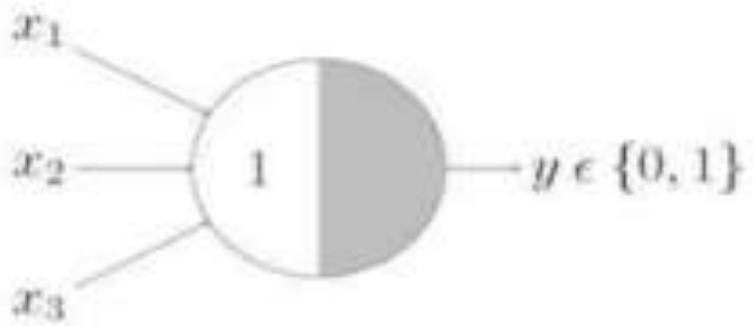
AND function

$$x_1 + x_2 = \sum_{i=1}^{2} x_i \geq 2$$



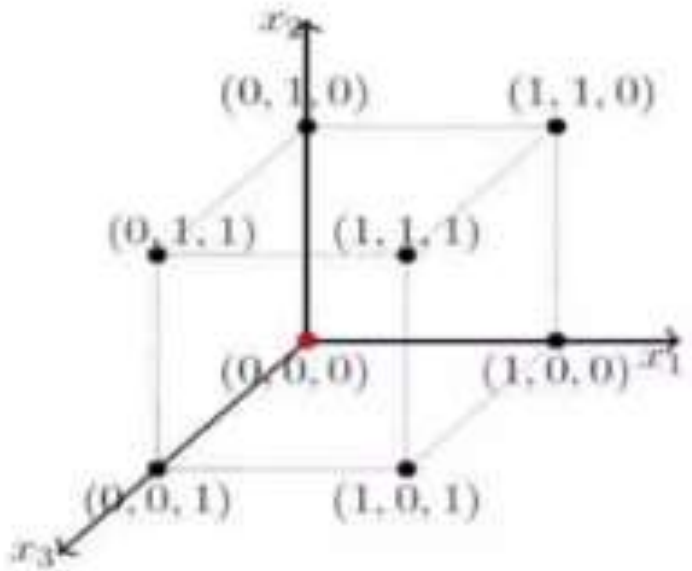
Geometric Interpretation Of M-P Neuron

OR Function With 3 Inputs:



OR function

$$x_1 + x_2 + x_3 = \sum_{i=1}^{3} x_i \geq 1$$



Linear Separability

- **It is a concept wherein the separation of the input space into regions is based on whether the network response is positive or negative. A decision line is drawn to separate positive or negative response. The decision line is also called as decision-making line or decision-support line or linear-separable line. The net input calculation to the output unit is given as**

$$y_{in} = b \; y_{in} - b_j + \sum_{i=1}^{n} xiwi \; wi$$

The region which is called as decision boundary is determined by the relation

$$b + \sum_{i=1}^{n} xiwi = 0$$

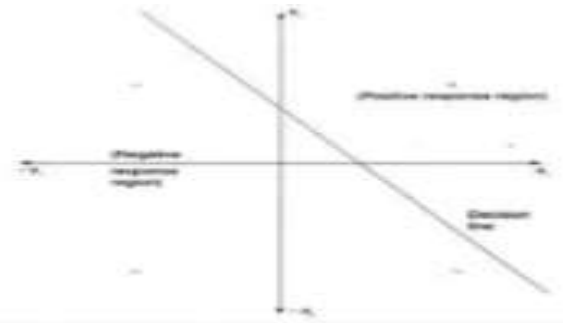
Linear Separability

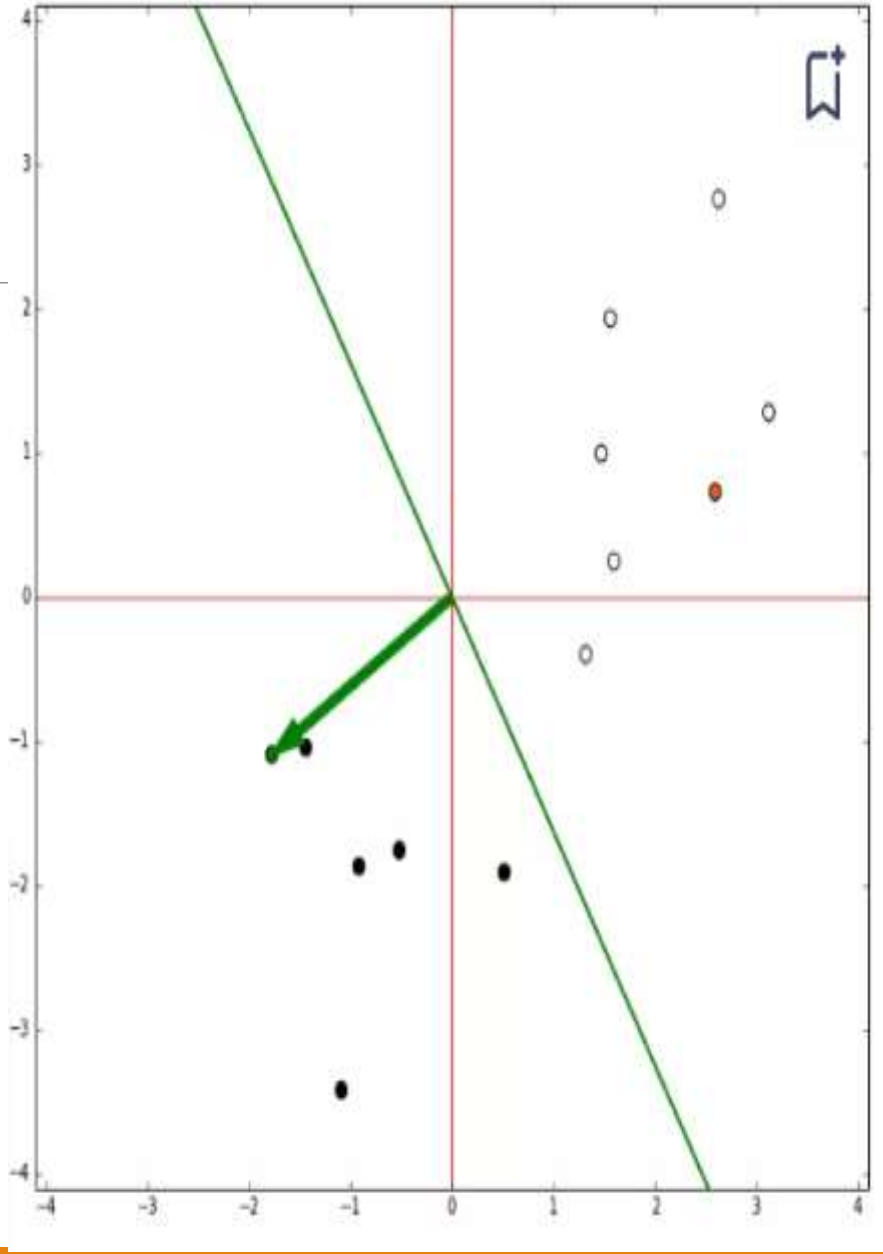
Consider a network having positive response in the first quadrant and negative response in all other quadrants with either binary or bipolar data.

Decision line is drawn separating two regions as shown in Fig.

Using bipolar data representation, missing data can be distinguished from mistaken data. Hence bipolar data is better than binary data.

Missing values are represented by 0 and mistakes by reversing the input values from +1 to-1 or vice versa.





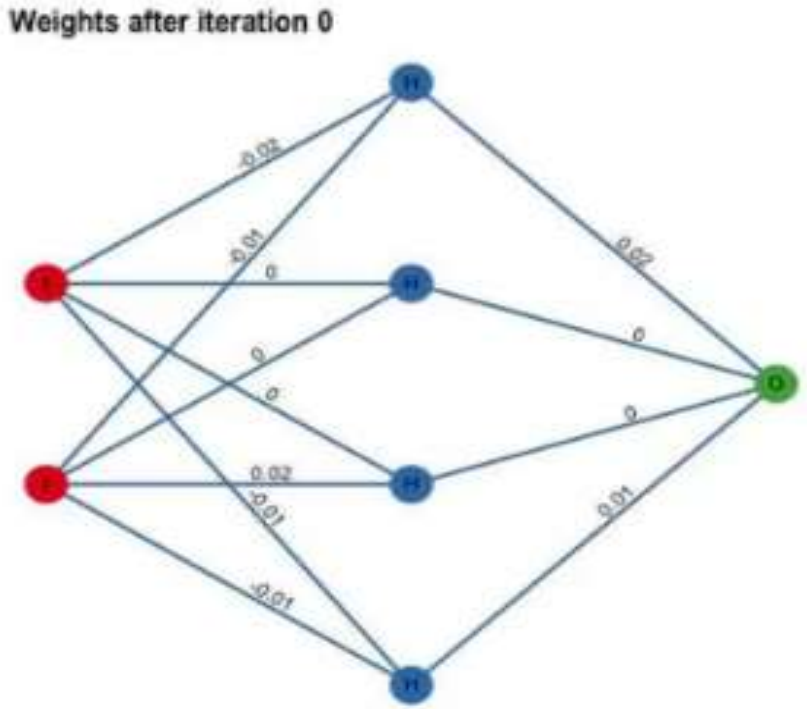
Hebb Network

- Donald Hebb stated in 1949 that " In brain, the learning is performed by the change in the synaptic gap".
- When an axon of cell A is near enough to excite cell B, and repeatedly or permanently takes place in firing it, some growth process or metabolic change takes place in one or both the cells such than A's efficiency, as one of the cells firing B, is increased.

Hebb Network

- According to Hebb rule, the weight vector is found to increase proportionately to the product of the input and the learning signal.
- In Hebb learning, two interconnected neurons are 'on' simultaneously.
- The weight update in Hebb rule is given by,

$$W_i(new) = w_i\,(old) + x_i\,y$$



Hebb Network

- The Hebb rule is more suited for bipolar data.
- If binary data is used, the weight updation formula cannot distinguish two conditions namely,
  - A training pair in which an input unit is "on" and the target value is "off"
  - A training pair in which both the input unit and the target value is "off".

  Thus there are limitations in Hebb rule application over binary data.
- Hence representation using bipolar data is advantageous.
- Hebb rule can be used for pattern association, pattern categorization, pattern classification and over a range of other areas.

Hebb Network-Training Algorithm

Steps:

Step 0: First initialize the weights.

Step 1: Steps 2-4 have to be performed for each input training vector and target output pair, s:t

Step 2: Input activations are set. The activation function for input layer is identity function.

X_1 =S; for i=1 to n

Step 3: Output activations are set.

Step 4: Weight adjustment and bias adjustments are performed.

W(new) = w;(old)+xiy

b(new)=b(old)+y

In step 4, the weight updation formula can be written in vector form as w(new)=w(old)+y

Change in weight is expressed as

Δw = xy

Hence,

w(new)=w(old) + Δw

Hebb Network-Flow Chart