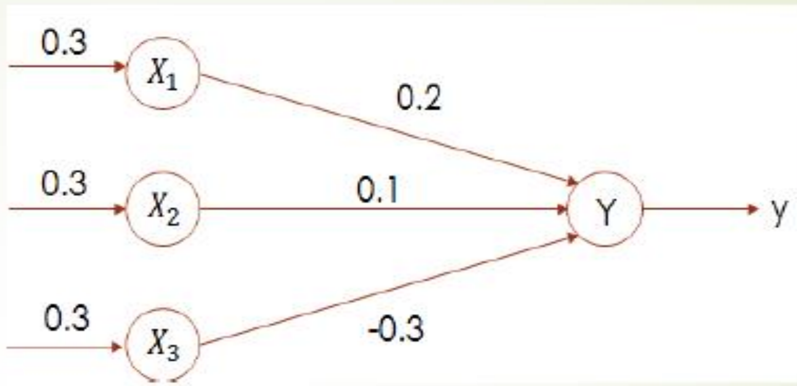
Module 1.2 (part 2) MP Neuron

Example of NN to calculate net output

Q1. For the network shown in Figure 1, calculate the weights are net input to the output neuron.



Solution: The given neural net consists of three input neurons and one output neuron. The inputs and weights are

$$[x_1, x_2, x_3] = [0.3, 0.5, 0.6]$$

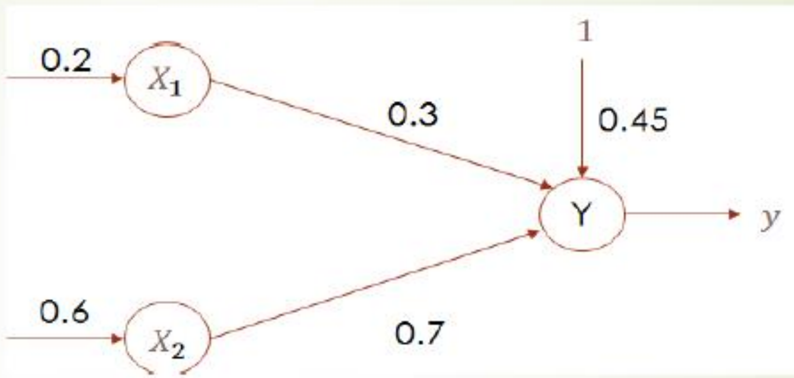
$$[w_1, w_2, w_3] = [0.2, 0.1, -0.3]$$

The net input can be calculated as

$$\begin{aligned} y_{in} &= x_1 w_1 + x_2 w_2 + x_3 w_3 \\ &= 0.3 \times 0.2 + 0.5 \times 0.1 + 0.6 \times (-0.3) \\ &= 0.06 + 0.05 - 0.18 = -0.07 \end{aligned}$$

Example of NN to calculate net output

Q2. Calculate the net input for the network shown in Figure 2 with bias included in the network.



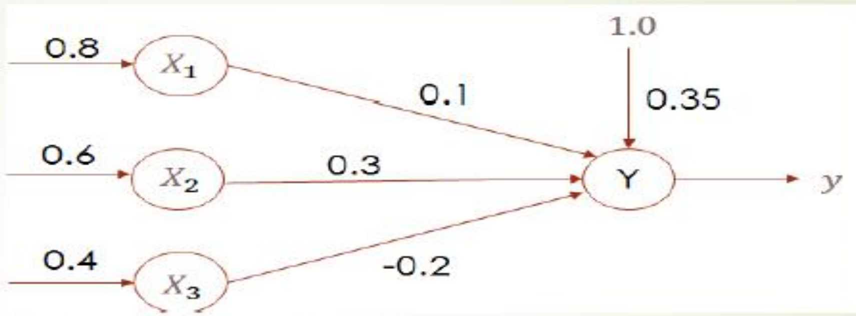
Solution: The given net consists of two input neurons a bias and an output neuron. The inputs are $[x_1, x_2] = [0.2, 0.6]$ and the weights are $[w_1, w_2] = [0.3, 0.7]$. Since the bias is included b=0.45 and bias input x_0 is equal to 1. The net input is calculated as

$$
\begin{aligned}
y_{in} &= b + x_1 w_1 + x_2 w_2 \\
&= 0.45 + 0.2 \times 0.3 + 0.6 \times 0.7 \\
&= 0.45 + 0.06 + 0.42 = 0.93
\end{aligned}
$$

Therefore, $y_{in} = 0.93$ is the net input.

Example of NN to calculate net output

Q3. Obtain the output of the neuron Y for the network shown in Figure 3 using activation function as: (1)binary sigmoidal and (ii) bipolar sigmoidal.



Solution: The given network has three neurons with bias and output neuron. These form a single layer network. The inputs ere given as $[x_1, x_2, x_3] = [0.8, 0.6, 0.4]$ and the weights are $[w_1, w_2, w_3]=[0.1, 0.3, -0.2]$ with bias b=0.35 (its input is always 1). The net input to the output neuron is

$$y_{in} = b + \sum_{i=1}^{n} x_i w_i$$

[n=3, because only 3 input neurons are given]

$$= b + x_1\, w_1 + x_2 w_2 + x_3 w_3$$
$$= 0.35 + 0.8 \times 0.1 + 0.6 \times 0.3 + 0.4 \times (-0.2)$$
$$= 0.35 + 0.08 + 0.18 - 0.08 = 0.53$$

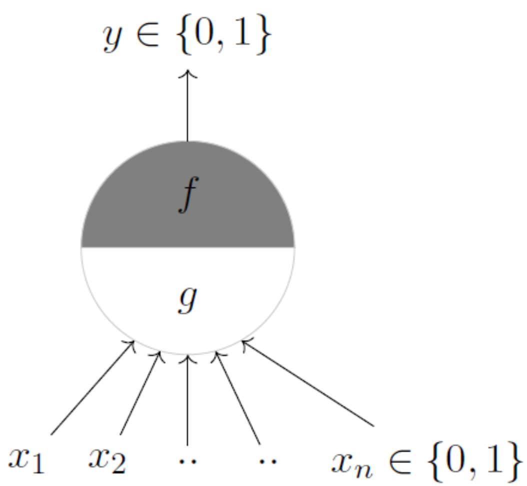
(i)For binary sigmoidal activation function,

$$y = f(y_{in}) = \frac{1}{1+e^{-y_{in}}} = \frac{1}{1+e^{-0.53}} = 0.625$$

(ii) For bipolar sigmoidal activation function,

$$y = f(y_{in}) = \frac{2}{1+e^{-y_{in}}} - 1 = \frac{2}{1+e^{-0.53}} - 1 = 0.259$$

McCulloch Pitts Neuron (MP neuron)



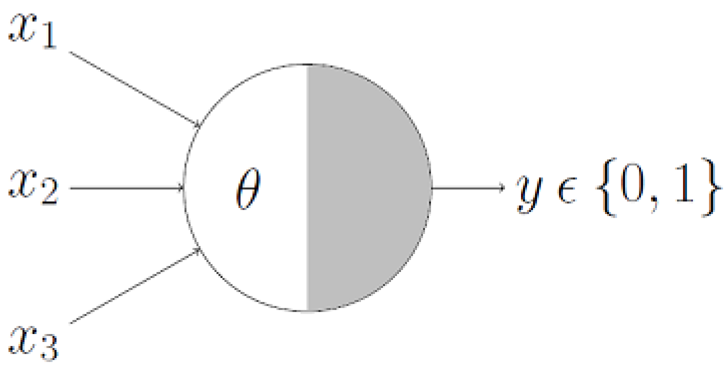
- McCulloch (neuroscientist) and Pitts (logician) proposed a highly simplified computational model of the neuron (1943)
- g aggregates the inputs and the function f takes a decision based on this aggregation
- The inputs can be excitatory or inhibitory
- $y = 0$ if any x_i is inhibitory, else

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

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

- θ is called the thresholding parameter
- This is called Thresholding Logic

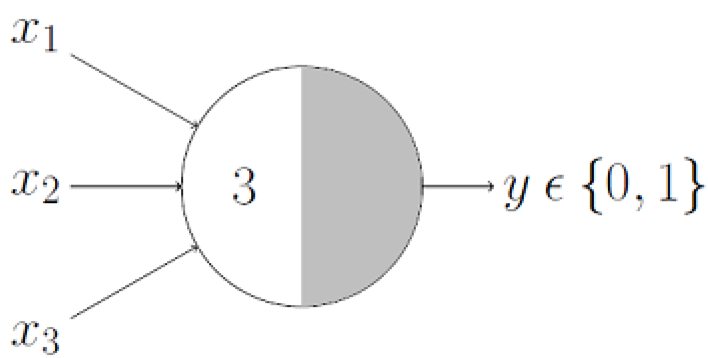
Boolean Functions Using M-P Neuron



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

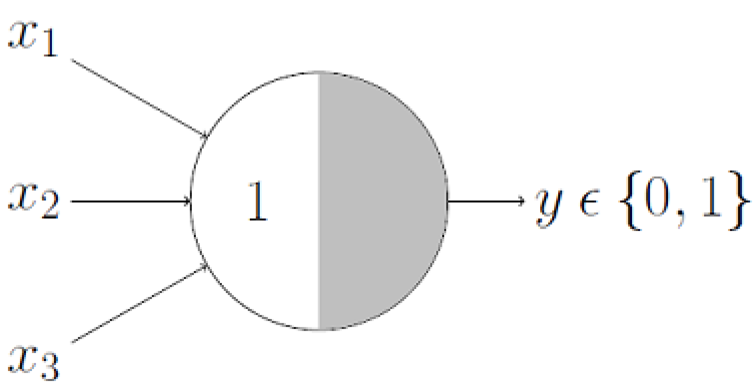
Concise representation of M-P neuron

AND Function



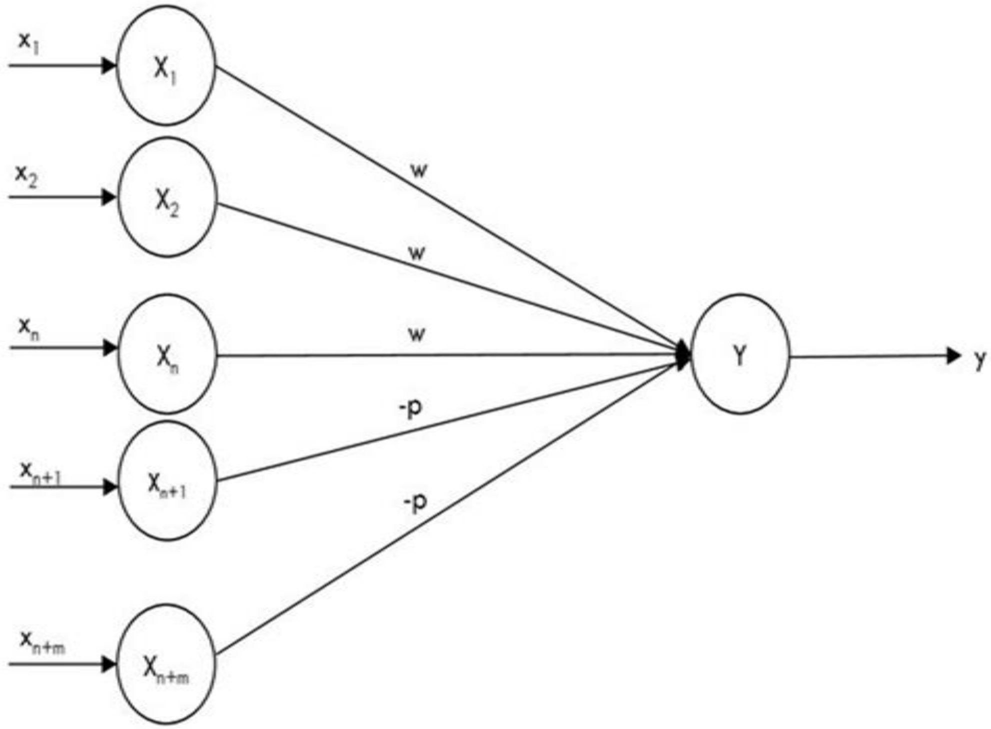
AND function neuron would only fire when ALL the inputs are ON i.e., ***g*(x)** ≥ 3 .

OR Function



- OR function neuron would fire if ANY of the inputs is ON i.e., g(x) ≥ 1

McCULLOCH-PITTS NEURON: ARCHITECTURE (MP NEURON MODEL)

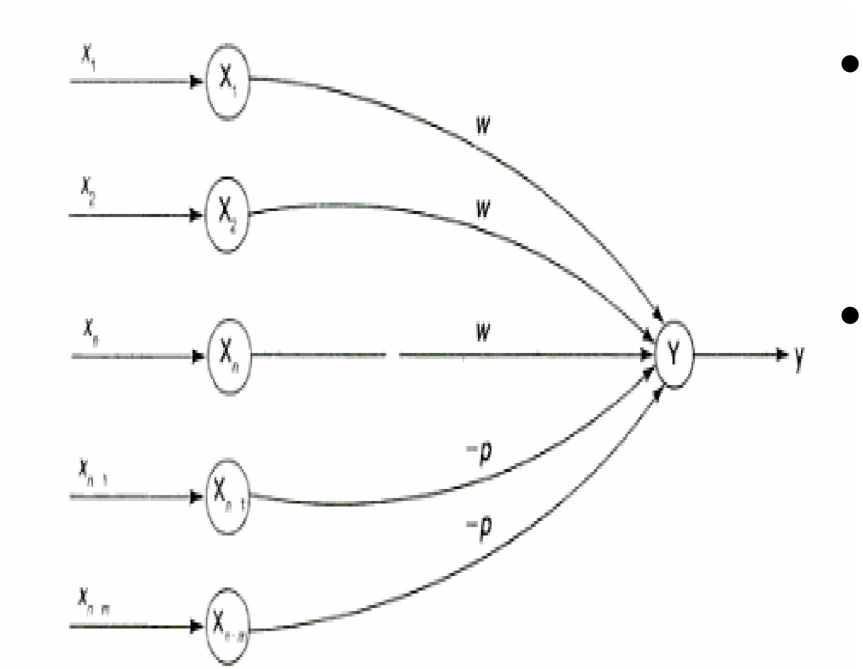


McCULLOCH-PITTS NEURON

Activation function for MP neuron is

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

And $$\theta > nw - p$$



- A simple M-P neuron is shown in the figure.
- It is excitatory with weight (*w>0*)/inhibitory with weight *–p (p<0)*.
- In the Fig., inputs from x_1 to x_n possess excitatory weighted connection and X_{n+1} to x_{n+m} has inhibitory weighted interconnections.
- Since the firing of neuron is based on threshold, activation function is defined as

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

- For inhibition to be absolute, the threshold with the activation function should satisfy the following condition:

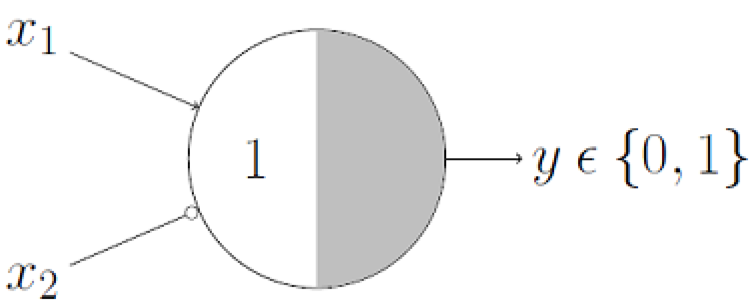
$$\theta > nw - p$$

- Output will fire if it receives "*k*" or more excitatory inputs but no inhibitory inputs where

$$kw \geq \theta > (k-1)\, w$$

- The M-P neuron has no particular training algorithm.
- An analysis is performed to determine the weights and the threshold.
- It is used as a building block where any function or phenomenon is modeled based on a logic function.

A Function With An Inhibitory Input



$x_1\ AND\ !x_2{}^*$

- we have an inhibitory input i.e., $\boldsymbol{x_2}$ so whenever $\boldsymbol{x_2}$ is 1, the output will be 0.
- $\boldsymbol{x_1}$ ***AND !*** $\boldsymbol{x_2}$ would output 1 only when $\boldsymbol{x_1}$ is 1 and $\boldsymbol{x_2}$ is 0

Computation of threshold for logical AND-NOT operation

6.

Case 1: Assume that both weights w_1 and w_2 are excitatory, i.e.,

$$w_1 = w_2 = 1$$

Then for the four inputs calculate the net input using

$$y_{in} = x_1 w_1 + x_2 w_2$$

For inputs

$$(1, 1),\ y_{in} = 1 \times 1 + 1 \times 1 = 2$$

$$(1, 0),\ y_{in} = 1 \times 1 + 0 \times 1 = 1$$

$$(0, 1),\ y_{in} = 0 \times 1 + 1 \times 1 = 1$$

$$(0, 0),\ y_{in} = 0 \times 1 + 0 \times 1 = 0$$

From the calculated net inputs, it is not possible to fire the neuron for input (1, 0) only. Hence, these weights are not suitable.

Assume one weight as excitatory and the other as inhibitory, i.e.,

$$w_1 = 1,\ w_2 = -1$$

Now calculate the net input. For the inputs

$$(1, 1),\ y_{in} = 1 \times 1 + 1 \times -1 = 0$$

$$(1, 0),\ y_{in} = 1 \times 1 + 0 \times -1 = 1$$

$$(0, 1),\ y_{in} = 0 \times 1 + 1 \times -1 = -1$$

$$(0, 0),\ y_{in} = 0 \times 1 + 0 \times -1 = 0$$

From the calculated net inputs, now it is possible to fire the neuron for input (1, 0) only by fixing a threshold of 1, i.e., $\theta \geq 1$ for Y unit. Thus,

$$w_1 = 1;\ w_2 = -1;\ \theta \geq 1$$

Note: The value of θ is calculated using the following:

$$\theta \geq nw - p$$

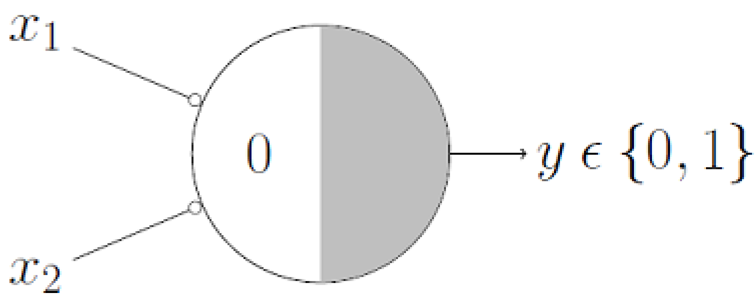
$$\theta \geq 2 \times 1 - 1 \quad \text{[for "}p\text{" inhibitory only magnitude considered]}$$

$$\theta \geq 1$$

Thus, the output of neuron Y can be written as

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

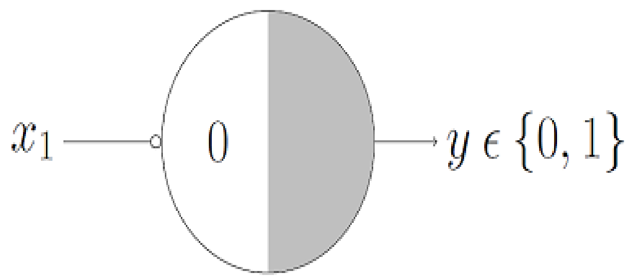
NOR Function



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

Computation of threshold for logical NOR operation (try)

NOT Function

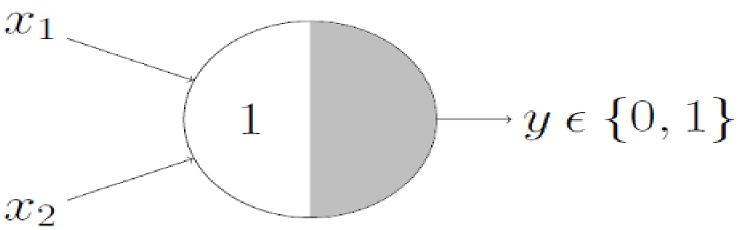


For a NOT neuron, 1 outputs 0 and 0 outputs 1. So we take the input as an inhibitory input and set the thresholding parameter to 0

Computation of threshold for logical NOT operation(try)

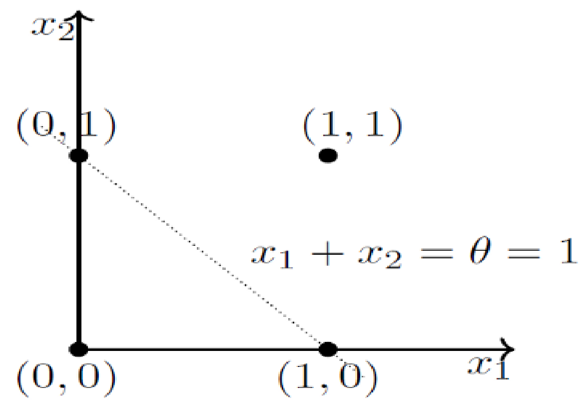
Design an NN using only MP-neuron for NAND (2 inputs) function (try!)

Geometrical interpretation of OR function MP-neuron



OR function

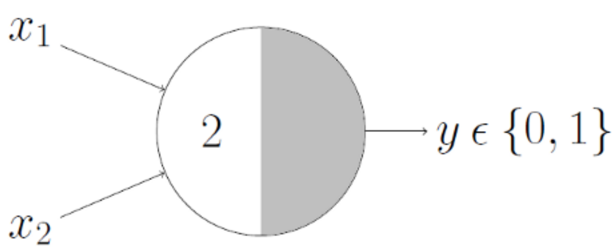
$$x_1 + x_2 = \sum_{i=1}^{2} x_i \geq 1$$



- A single MP neuron splits the input points (4points for 2 binary inputs) into two halves
- Points lying on or above the line $\sum_{i-1}^{n} x_i$-θ =0 and points lying below this line.

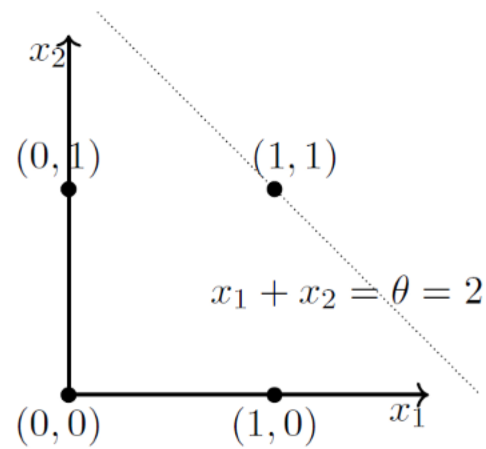
- all inputs which produce an output 0 will be on one side ($\sum_{i=1}^{n} x_i$< θ) of the line and all inputs which produce an output 1 will lie on the other side ($\sum_{i=1}^{n} x_i$> θ) of this line

Geometrical interpretation of AND function MP-neuron

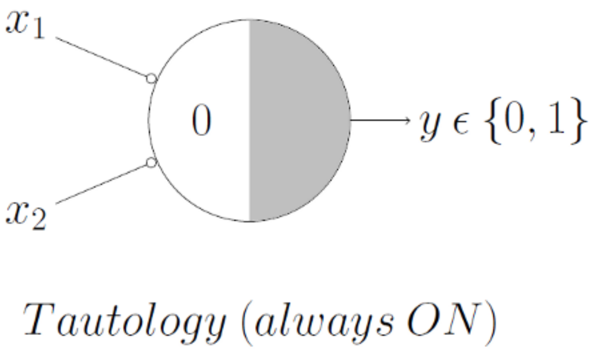


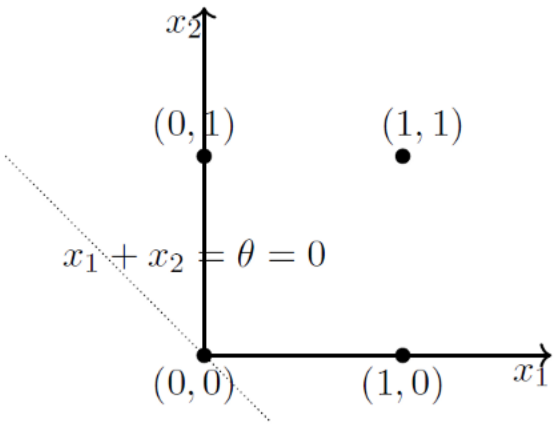
AND function

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

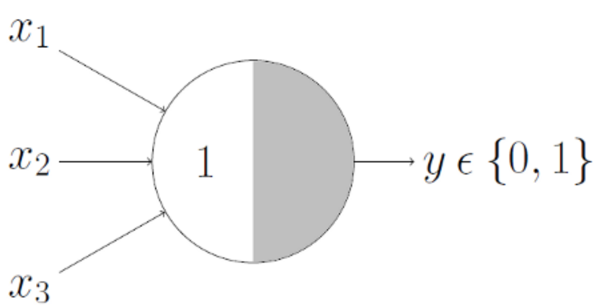


Geometrical interpretation of Tautology MP-neuron



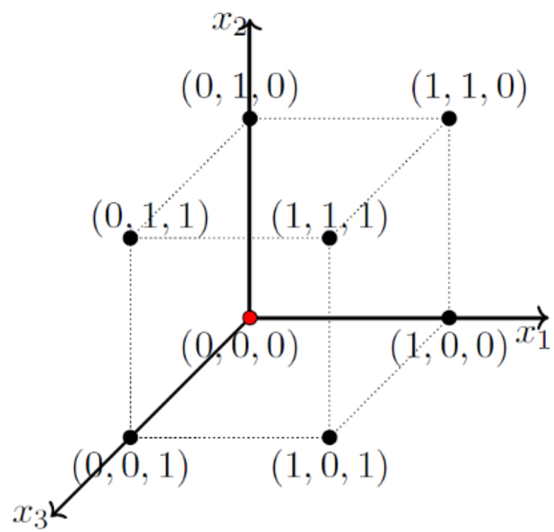


Geometrical interpretation of OR function MP-neuron with 3 inputs

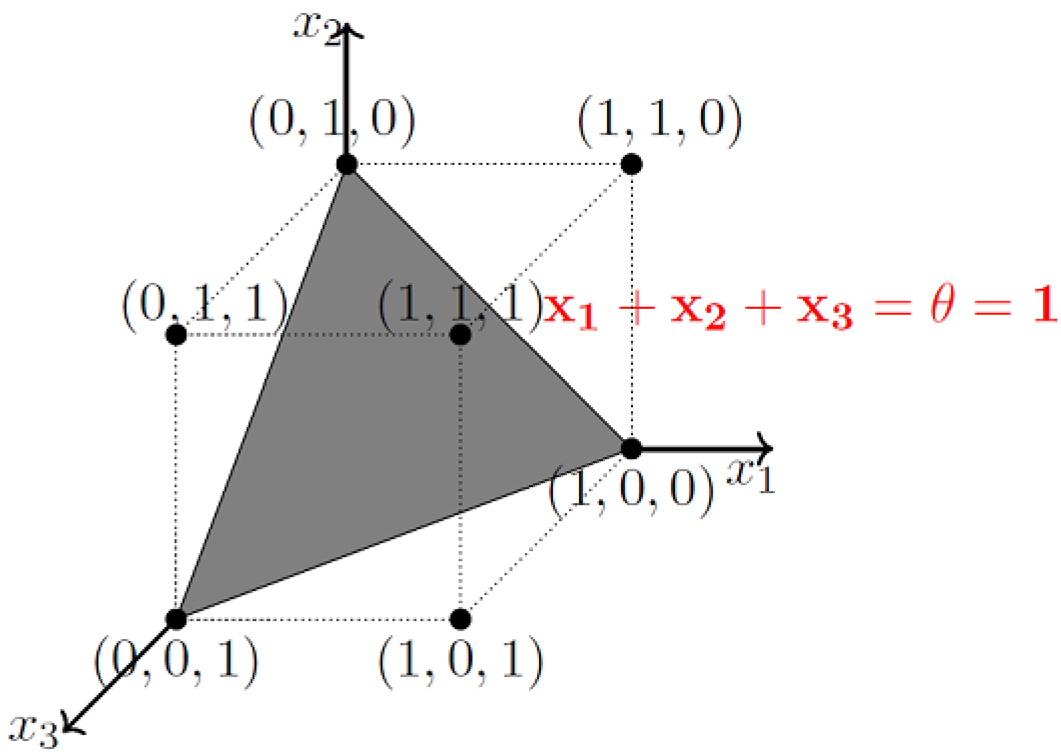


OR function

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



Geometrical interpretation of OR function MP-neuron with 3 inputs

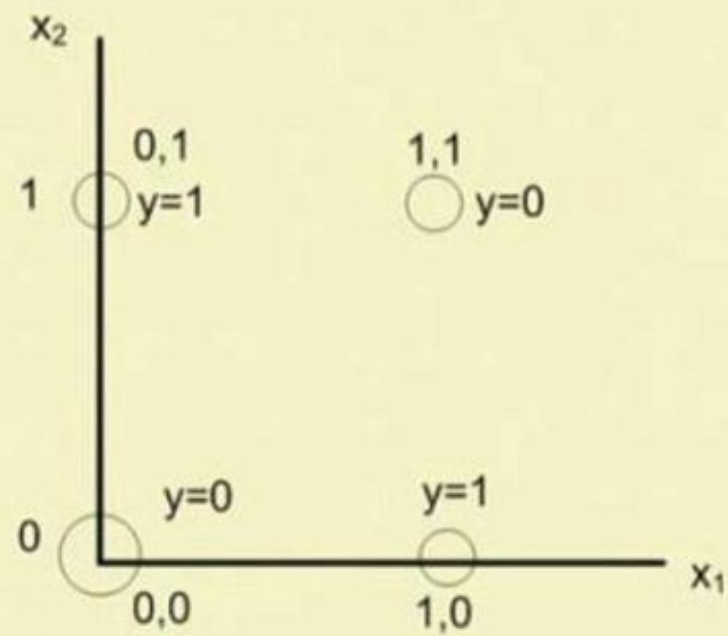


For the OR function, we want a plane such that the point (0,0,0) lies on one side and the remaining 7 points lie on the other side of the plane

XOR PROBLEM IS LINEARLY NON-SEPARABLE

x_1	x_2	Output (y)
0	0	0
0	1	1
1	0	1
1	1	0

XOR Problem

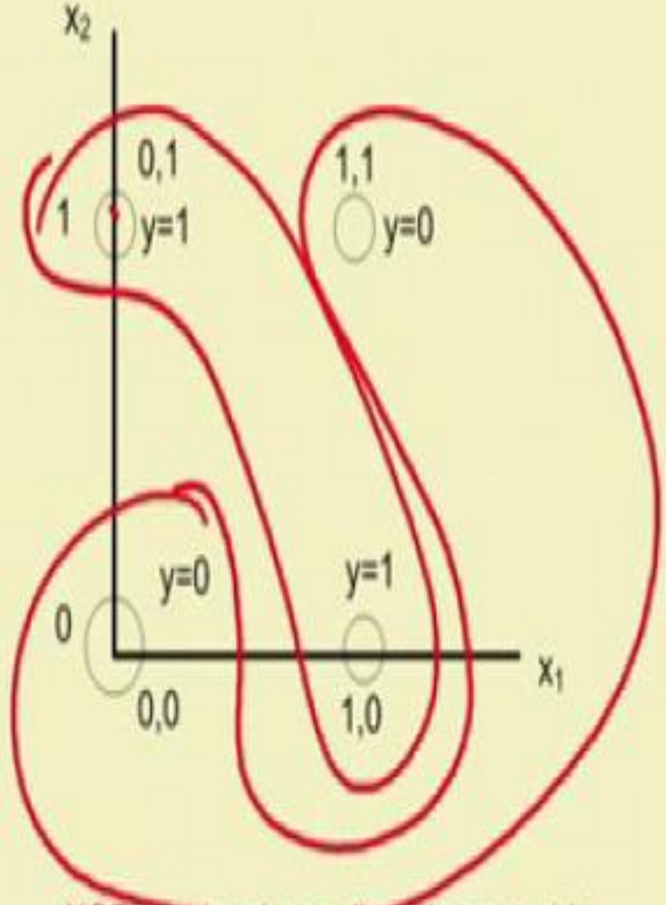


XOR-problem is non-linearly separable

XOR problem is linearly non-separable

x_1	x_2	Output (y)
0	0	0
0	1	1
1	0	1
1	1	0

XOR Problem



XOR-problem is non-linearly separable

Thus..

- A single McCulloch Pitts Neuron can be used to represent boolean functions which are linearly separable
- **Linear separability (for boolean functions)** : There exists a line (plane) such that all inputs which produce a 1 lie on one side of the line (plane) and all inputs which produce a 0 lie on other side of the line (plane)

Limitation of MP-neuron

- What about non-boolean (say, real) inputs?
- Do we always need to hand code the threshold?
- Are all inputs equal? What if we want to assign more importance to some inputs?
- What about functions which are not linearly separable? Say XOR function.

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

$$y_{in} = b_j + \sum_{i=1}^{n} xiwi$$

- The region which is called as *decision boundary* is determined by the relation $b + \sum_{i=1}^{n} xiwi = 0$

Contd..

- Linear separability of the network is based on decision-boundary line.
- If there exists weights(bias) for which training input vectors have positive response (+1 ve) input lie on one side of line and other having negative response(-ve) lie on other side of line, we conclude the problem is linearly separable

Contd..

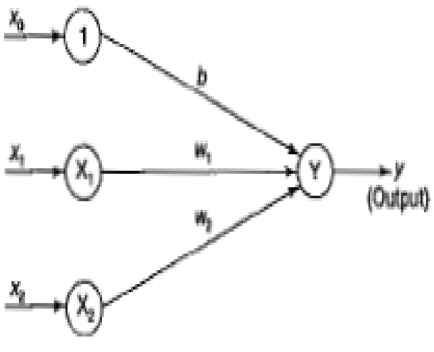


Figure 2-19 A single-layer neural net.

Net input of the network

$$Y_{in} = b + x_1w_1 + x_2w_2$$

Separating line for which the boundary lies between the values of x_1 and x_2

$$b + x_1w_1 + x_2w_2 = 0$$

If weight of w_2 is not equal to 0 then we get

$$x2 = -w_1/w_2 - b/w_2$$

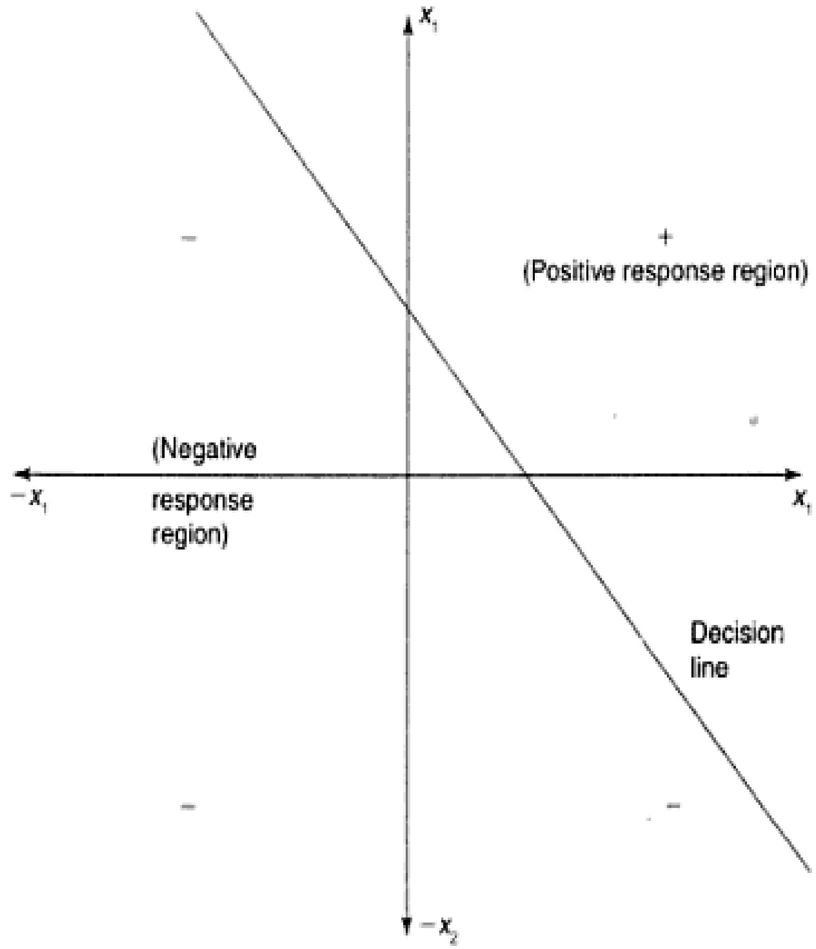
Net requirement for positive response

$$b + x_1w_1 + x_2w_2 > 0$$

During training process , from training data w1, w2 and b are decided.

Contd..

- 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.



Numericals on linear separability to be added