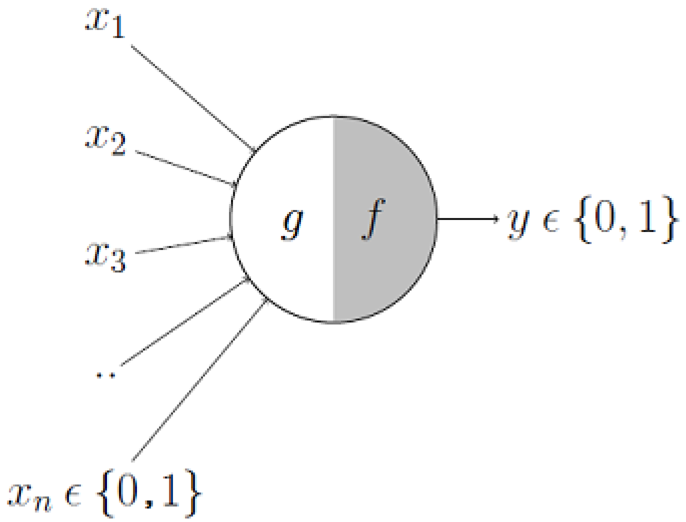
McCulloch-Pitts Neuron

McCulloch-Pitts Neuron

- The first computational model of a neuron was proposed by Warren MuCulloch (neuroscientist) and Walter Pitts (logician) in 1943.



- It may be divided into 2 parts.
  - The first part, ***g*** takes an input (ahem dendrite ahem), performs an aggregation
  - Based on the aggregated value the second part, ***f*** makes a decision.

McCulloch-Pitts Neuron

Lets suppose that we want to predict our own decision, whether to watch a random football game or not on TV.

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 *isPremierLeagueOn* (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 *isManUnitedPlaying* (I am a big Man United fan. GGMU!) and so on.

McCulloch-Pitts Neuron

- 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 output will always be 0
  - the neuron will never fire, so ***x_3*** is an inhibitory input.
- **Excitatory inputs**
  - NOT the ones that will make the neuron fire on their own but they might fire it when combined together.

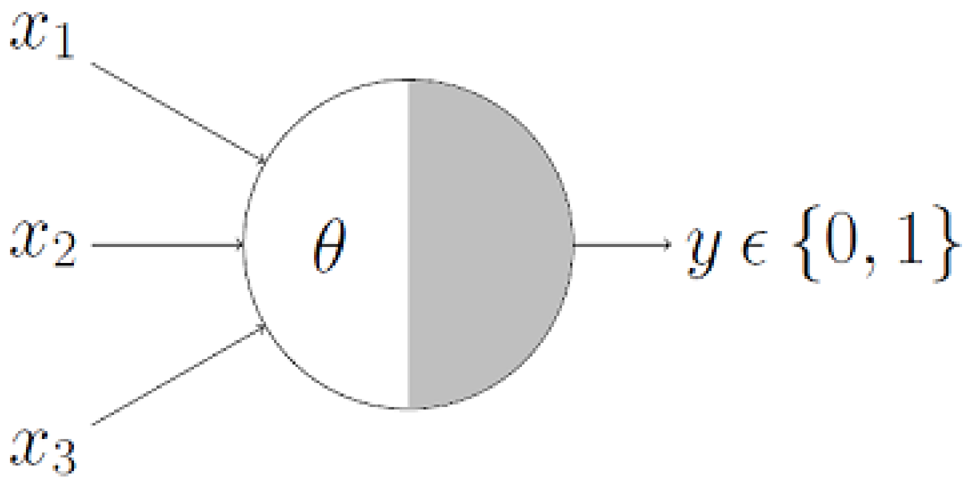
McCulloch-Pitts Neuron

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

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

- ***g*(x)** is just a sum of the inputs — a simple aggregation.
- ***theta*** here is called thresholding parameter.
- For example, if one always watch the game then the sum turns out to be 2 or more, the ***theta*** is 2 here. This is called the Thresholding Logic.

M-P Neuron: A Concise Representation



Representation just denotes that,

for the boolean inputs ***x_1***, ***x_2*** and ***x_3***

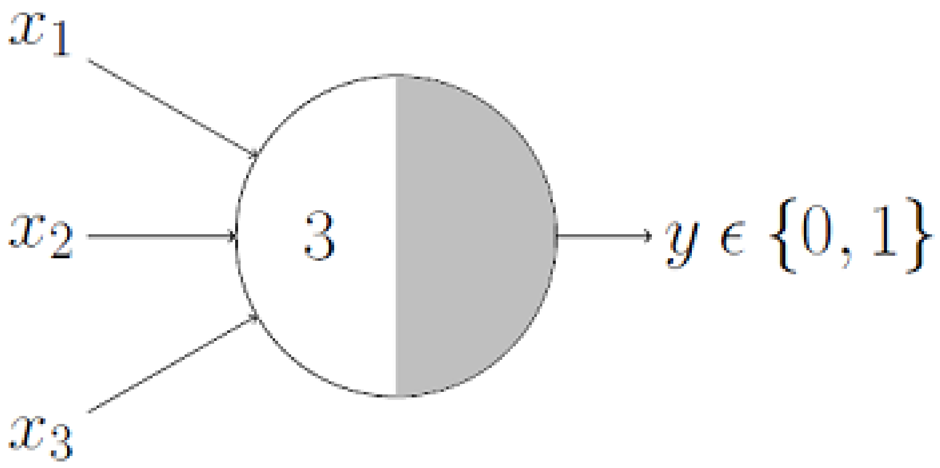
if the ***g*(x)** i.e., **sum ≥ theta**, the neuron will fire otherwise, it won't.

Boolean Functions Using M-P Neuron

- Inputs are all boolean and the output is also boolean
- Essentially, the neuron is just trying to learn a boolean function.
- A lot of boolean decision problems can be represented by the M-P neuron based on appropriate input variables—
  - like whether to continue reading a particular post
  - whether to watch movie etc..

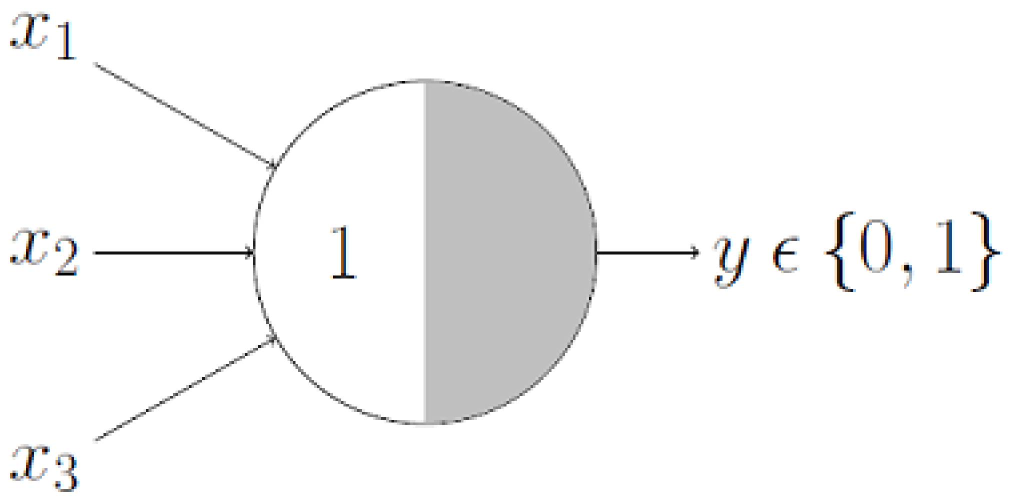
Boolean Functions Using M-P Neuron -AND Function

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

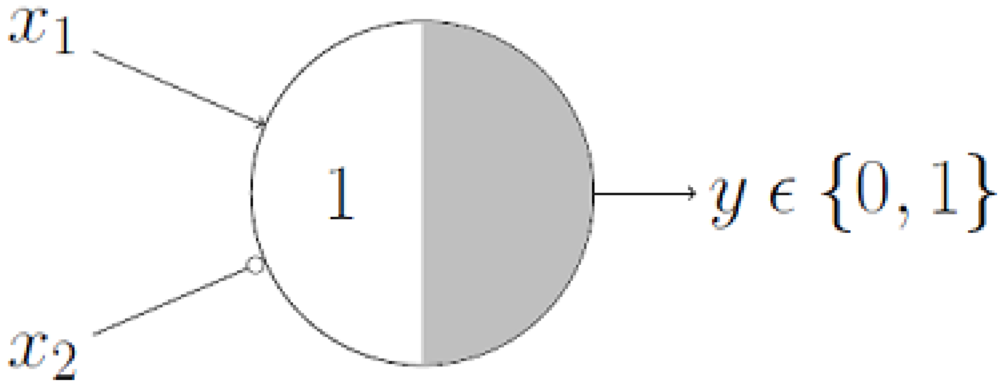


Boolean Functions Using M-P Neuron - OR Function

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



Boolean Function With An Inhibitory Input –AND function



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

Here, we have an inhibitory input i.e., ***x_2*** so whenever ***x_2*** is 1, the output will be 0.

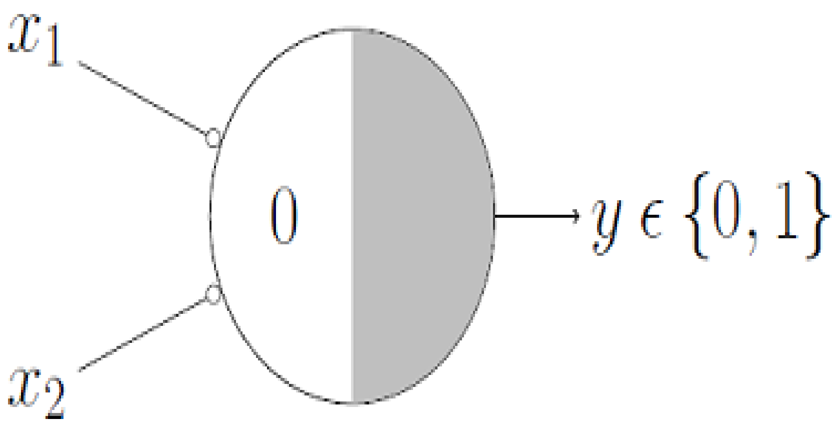
Keeping that in mind, we know that ***x_1 AND !x_2*** would output 1 only when ***x_1*** is 1 and ***x_2*** is 0

So it is obvious that the threshold parameter should be 1.

Boolean Function With An Inhibitory Input- AND function

- Lets verify that, the ***g*(x)** i.e., $x_1 + x_2$ would be ≥ 1 in only 3 cases:
  - 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 (because of inhibition)
- And we also know that ***x_1 AND !x_2*** would output 1 for Case 1 (above) so our thresholding parameter holds good for the given function.

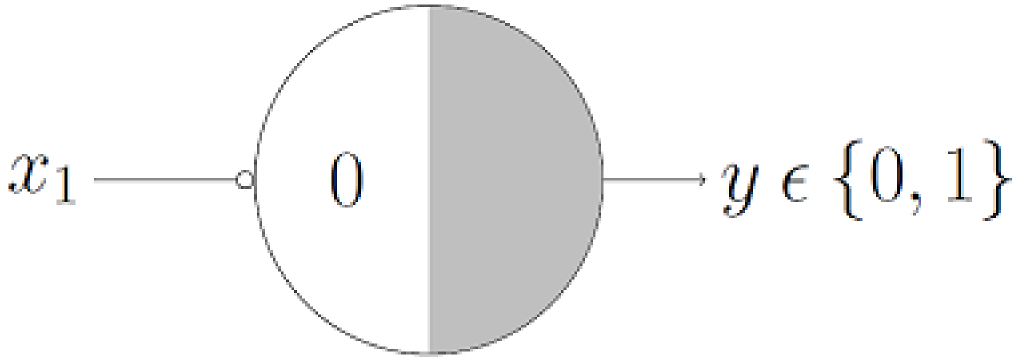
Boolean Function With An Inhibitory Input - NOR Function



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.

Boolean Function With An Inhibitory Input - NOT Function



For a NOT neuron, 1 outputs 0 and 0 outputs 1.

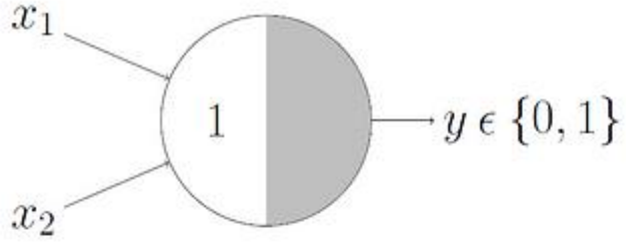
So we take the input as an inhibitory input and set the thresholding parameter to 0. It works!

Can any boolean function be represented using the M-P neuron?

Before you answer that, lets understand what M-P neuron is doing geometrically.

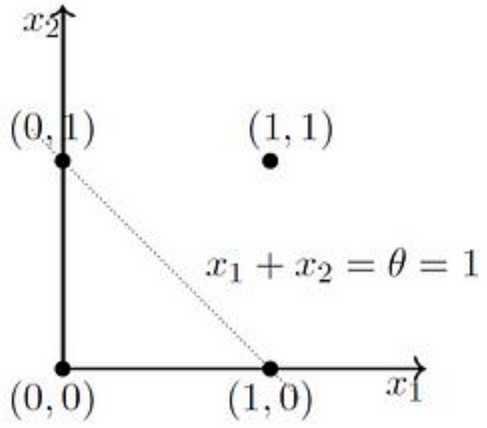
Geometric Interpretation Of M-P Neuron

- **OR Function**
  - We already discussed that the OR function's thresholding parameter ***theta*** is 1The inputs are obviously boolean,
  - so only 4 combinations are possible — (0,0), (0,1), (1,0) and (1,1).
  - Now plotting them on a 2D graph and making use of the OR function's aggregation equation i.e., ***x_1 + x_2 ≥ 1*** using which we can draw the decision boundary as shown in the graph below(Mind you again, this is not a real number graph)



OR function

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



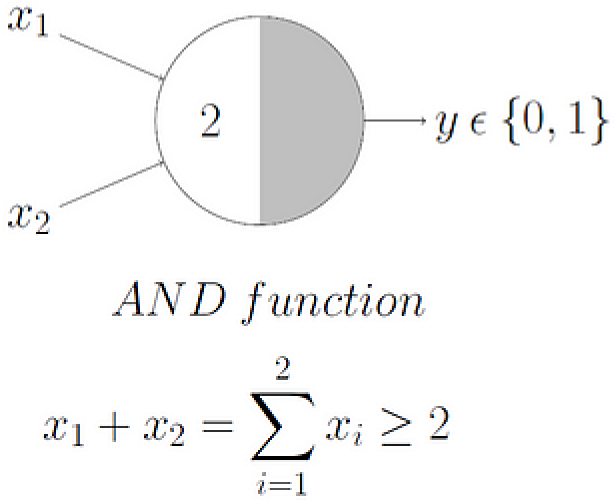
- The aggregation equation i.e., ***x_1 + x_2 =1*** graphically show that:
  - All those inputs whose output when passed through the OR function M-P neuron
  - lie ON or ABOVE that line output 1
  - all the input points that lie BELOW that line are going to output 0

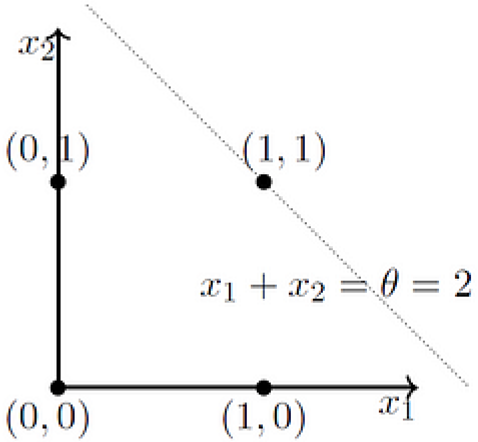
Geometric Interpretation Of M-P Neuron

- The M-P neuron just learnt a linear decision boundary
- The M-P neuron is splitting the input sets into two classes — positive and negative.
- Positive ones (which output 1) are those that lie ON or ABOVE the decision boundary
- Negative ones (which output 0) are those that lie BELOW the decision boundary.

Geometric Interpretation Of M-P Neuron

AND Function

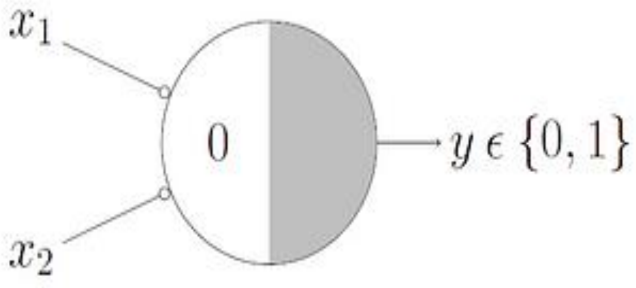




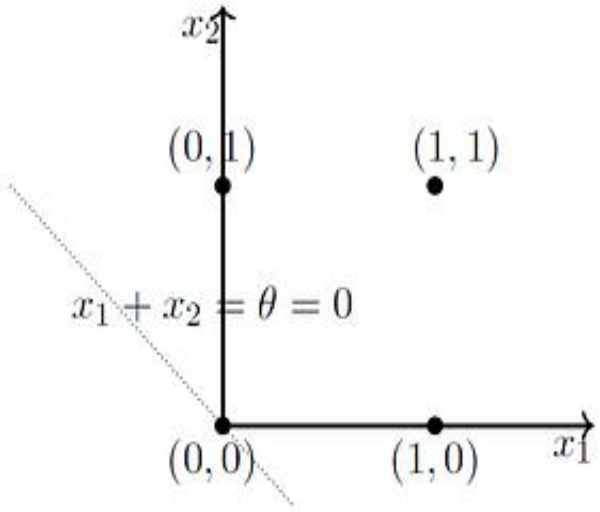
- In this case, the decision boundary equation is ***x_1 + x_2 =2***.
- Here, all the input points that lie ON or ABOVE, just (1,1), output 1 when passed through the AND function M-P neuron.
- It fits! The decision boundary works!

Geometric Interpretation Of M-P Neuron

- **Tautology**



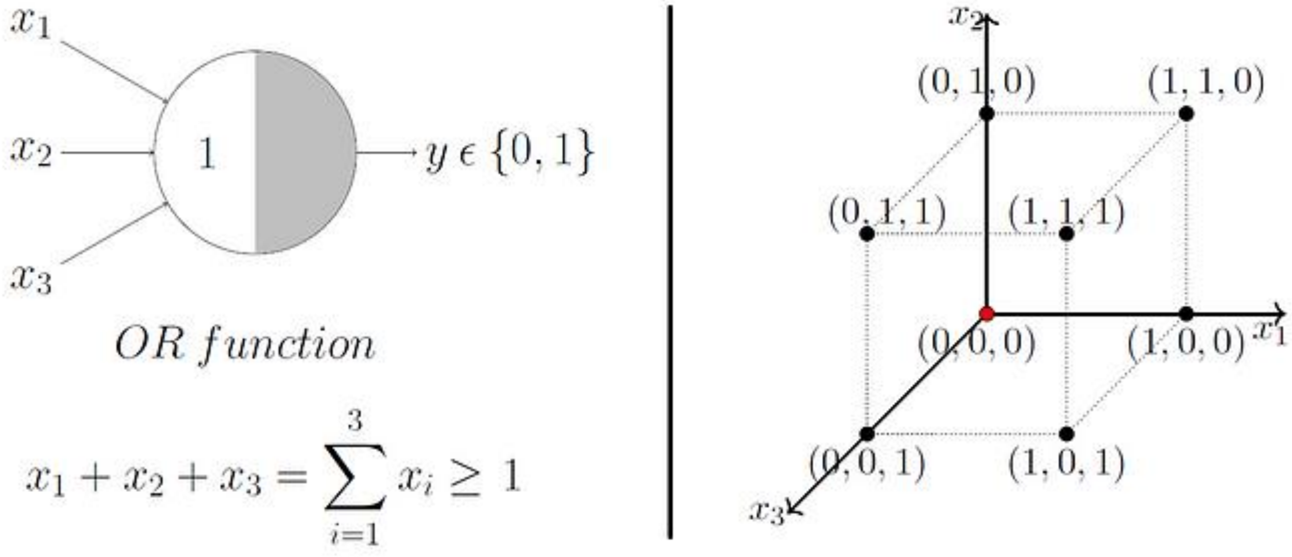
Tautology (always ON)



- In this case, the decision boundary equation is ***x_1* + *x_2* =0**.
- Here, all the input points that lie ON or ABOVE, just (0,0), output 1 when passed through the AND function M-P neuron.

Geometric Interpretation Of M-P Neuron

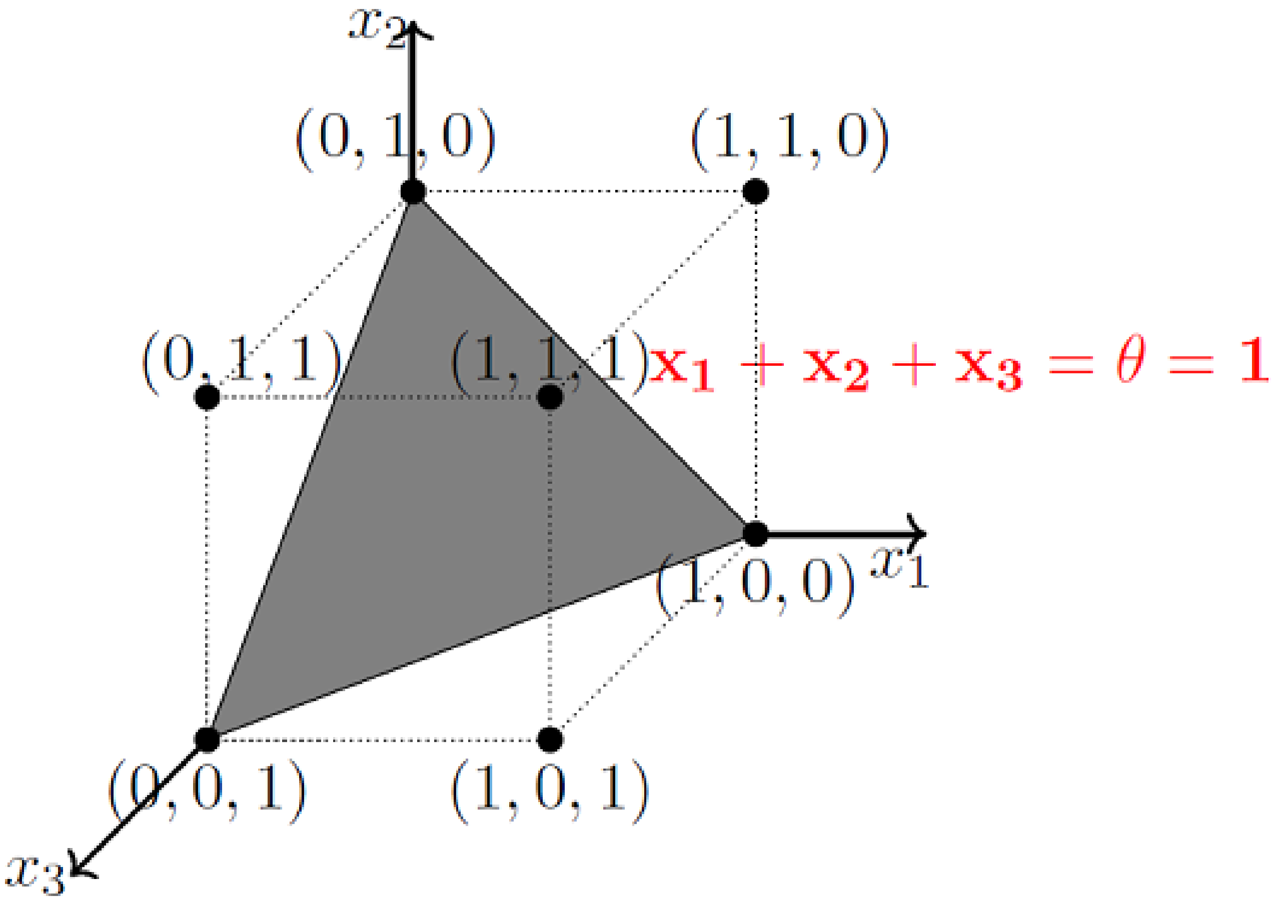
OR Function With 3 Inputs



- In this case, the possible inputs are 8 points — (0,0,0), (0,0,1), (0,1,0), (1,0,0), (1,0,1),... you got the point(s).
- We can map these on a 3D graph and this time we draw a decision boundary in 3 dimensions.
- *"Is it a bird? Is it a plane?"*
- Yes, it is a PLANE!

Geometric Interpretation Of M-P Neuron

The plane that satisfies the decision boundary equation ***x_1* + *x_2* + *x_3* = 1**



- All the points that lie ON or ABOVE that plane (positive half space) will result in output 1 when passed through the OR function M-P unit

- All the points that lie BELOW that plane (negative half space) will result in output 0.

Geometric Interpretation Of M-P Neuron

- By hand coding a thresholding parameter,
  - M-P neuron is able to conveniently represent the 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).

Limitations Of M-P Neuron

- Can’t handle Non-boolean (say, real) inputs
- Always need to hand code the threshold
- All inputs are considered equal. How to assign more importance to some inputs?
- What about functions which are not linearly separable? Say XOR function.