McCulloch Pitts Neuron and Thresholding Logic

Last Lecture

- What, Why, How?
- Brief History

Today’s Topics

- Biological Neuron
- Artificial Neuron
- McCulloch Pitts Neuron
- Thresholding Logic

Inspiration: Animal's Computing Machinery

Neuron

- basic unit in the nervous system for receiving, processing, and transmitting information; e.g., messages such as…

"hot"

https://www.clipart.email/clipart/don t-touch-hot-stove-clipart-73647.html

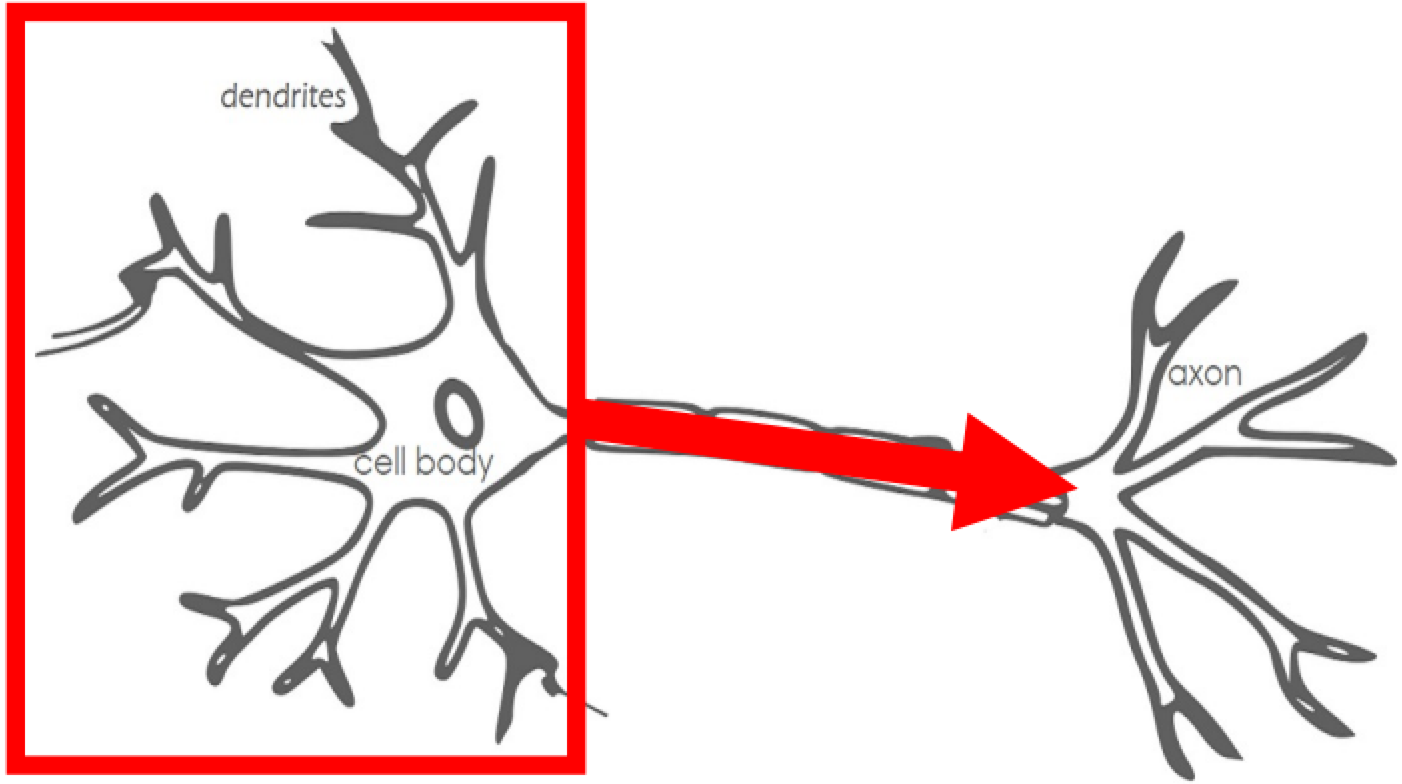
"loud"

https://kisselpaso.com/if-the-sun-city-music-fest-gets-too-loud-there-is-a-phone-number-you-can-call-to-complain/

"spicy"

https://www.babycenter.com/404_when-can-my-baby-eat-spicy-foods_1368539.bc

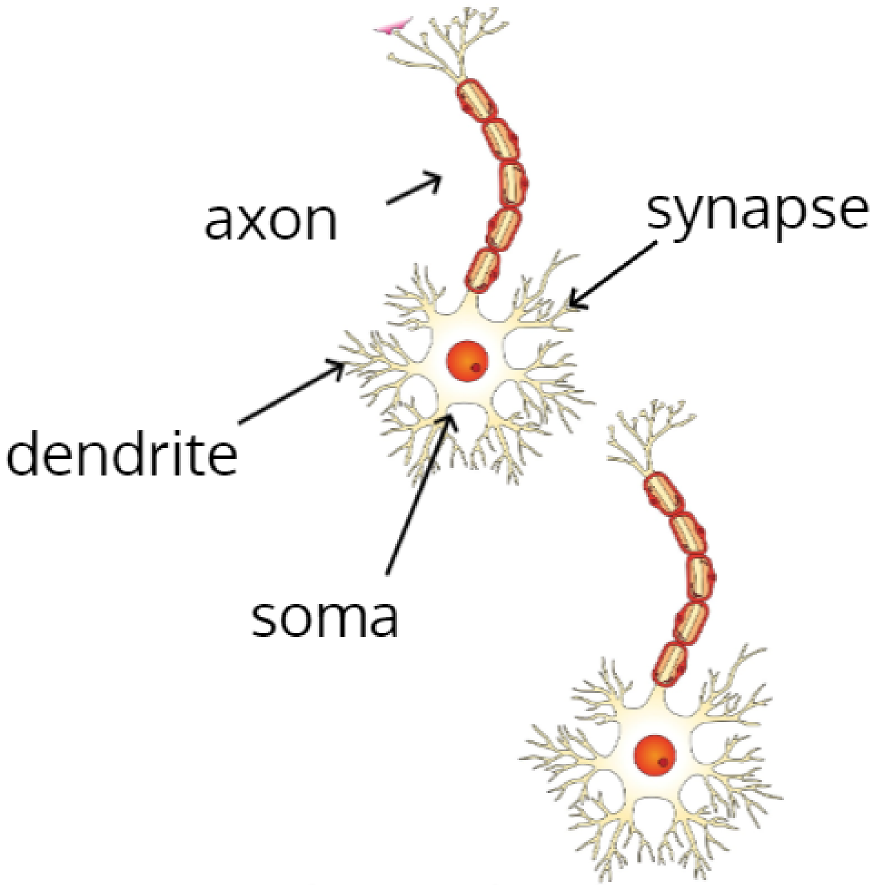
Inspiration: Neuron “Firing”



- When the input signals exceed a certain threshold within a short period of time, a neuron “fires”
- Neuron “firing” (outputs signal) is an “all-or-none” process

Image Source: https://becominghuman.ai/introduction-to-neural-networks-bd042ebf2653

Biological Neuron



Biological Neurons*

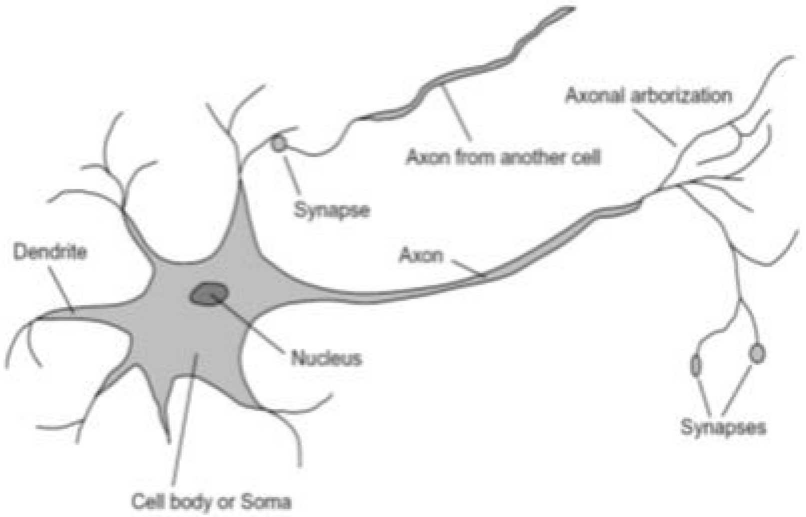
dendrite: receives signals from other neurons

synapse: point of connection to other neurons

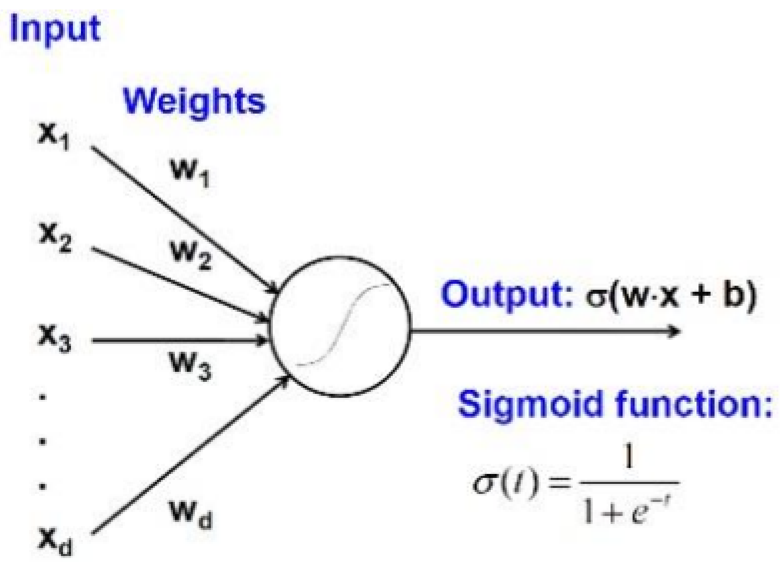
soma: processes the information

axon: transmits the output of this neuron

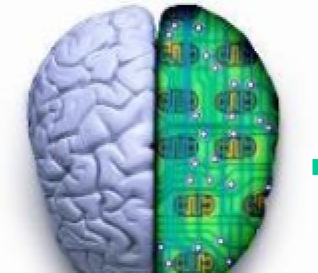
Biological Analog of Artificial Neuron



A biological neuron



An artificial neuron



– A neuron is a mathematical function modeled on the working of biological neurons

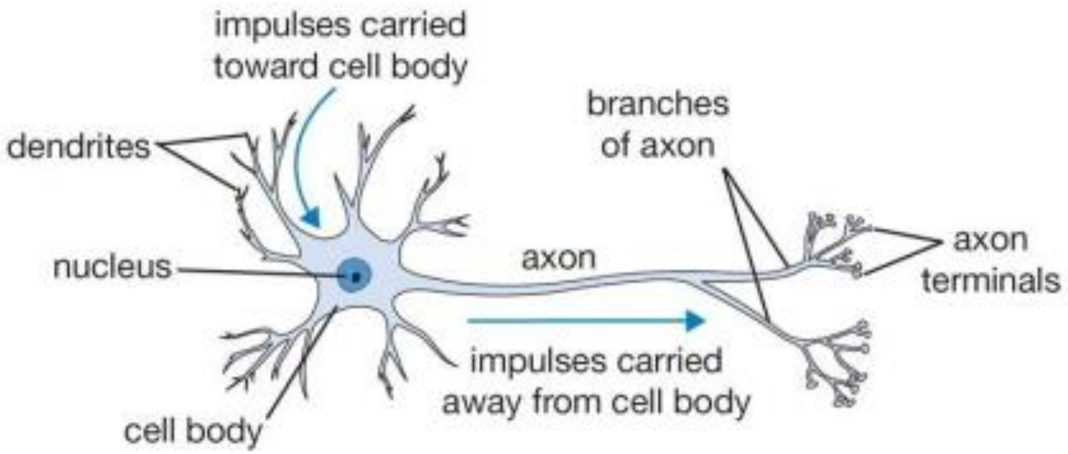
– One or more inputs are separately weighted

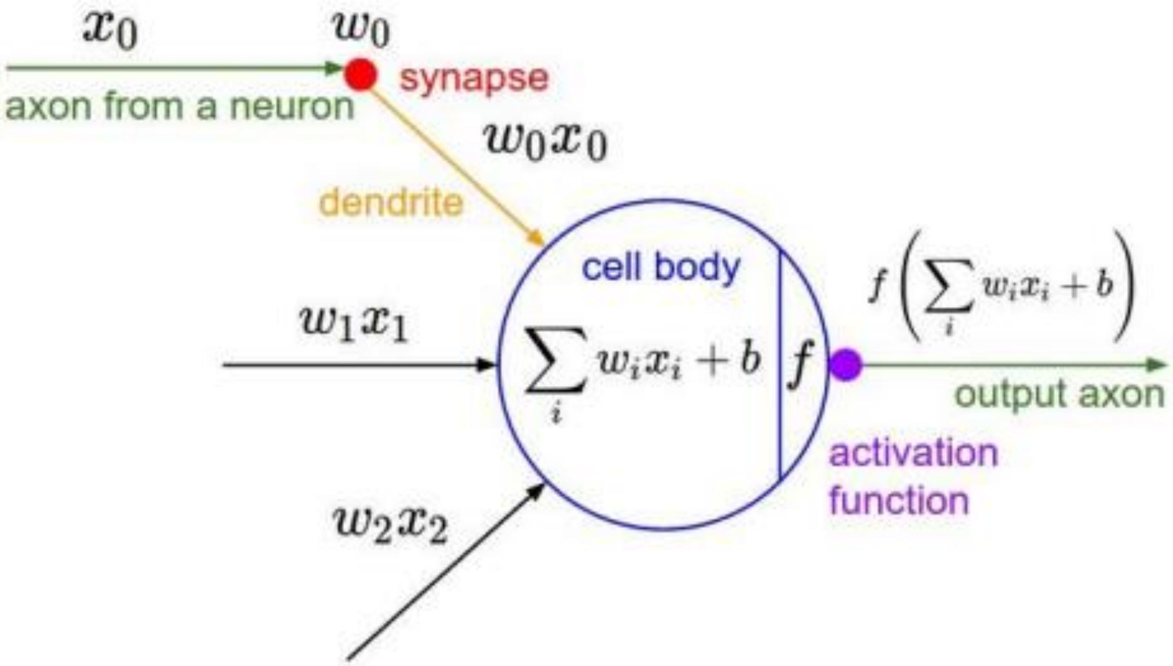
– Inputs are summed and passed through a nonlinear function to produce output

– Every neuron holds an internal state called activation signal

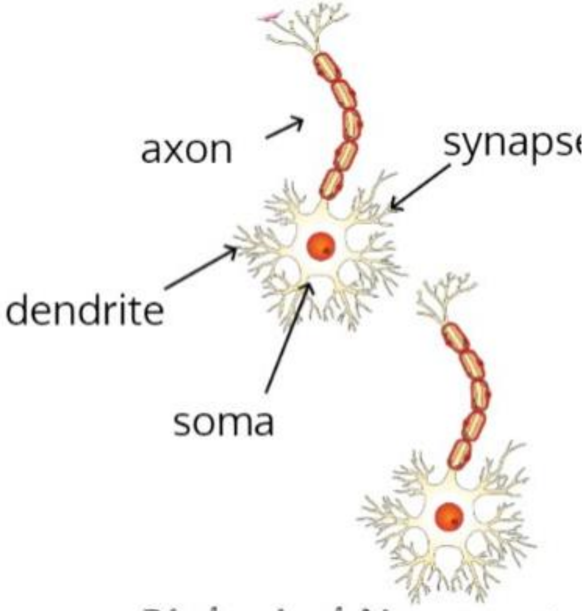
– Each connection link carries information about the input signal

Biological Analog



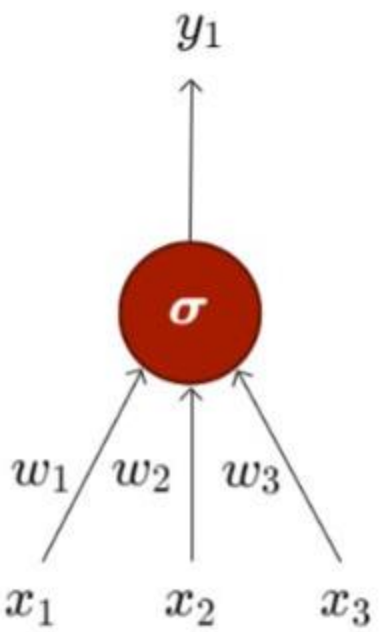


Biological vs. Artificial Neuron



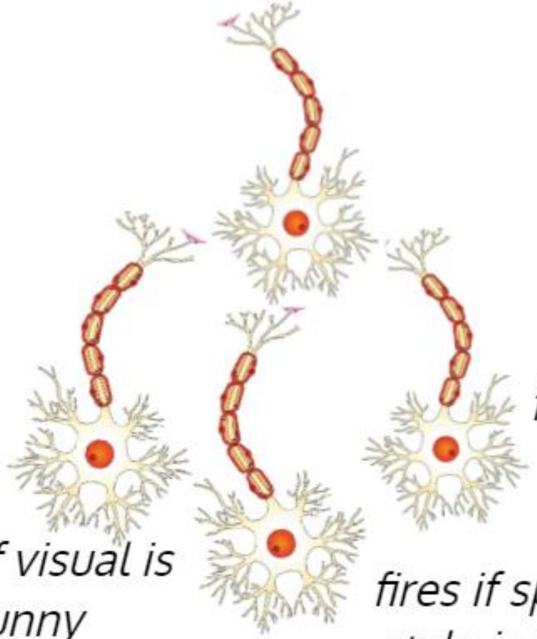
Biological Neurons*

Biological Neuron	Artificial Neuron
Cell Nucleus (Soma)	Node
Dendrites	Input
Synapse	Weights or interconnecti
Axon	Output

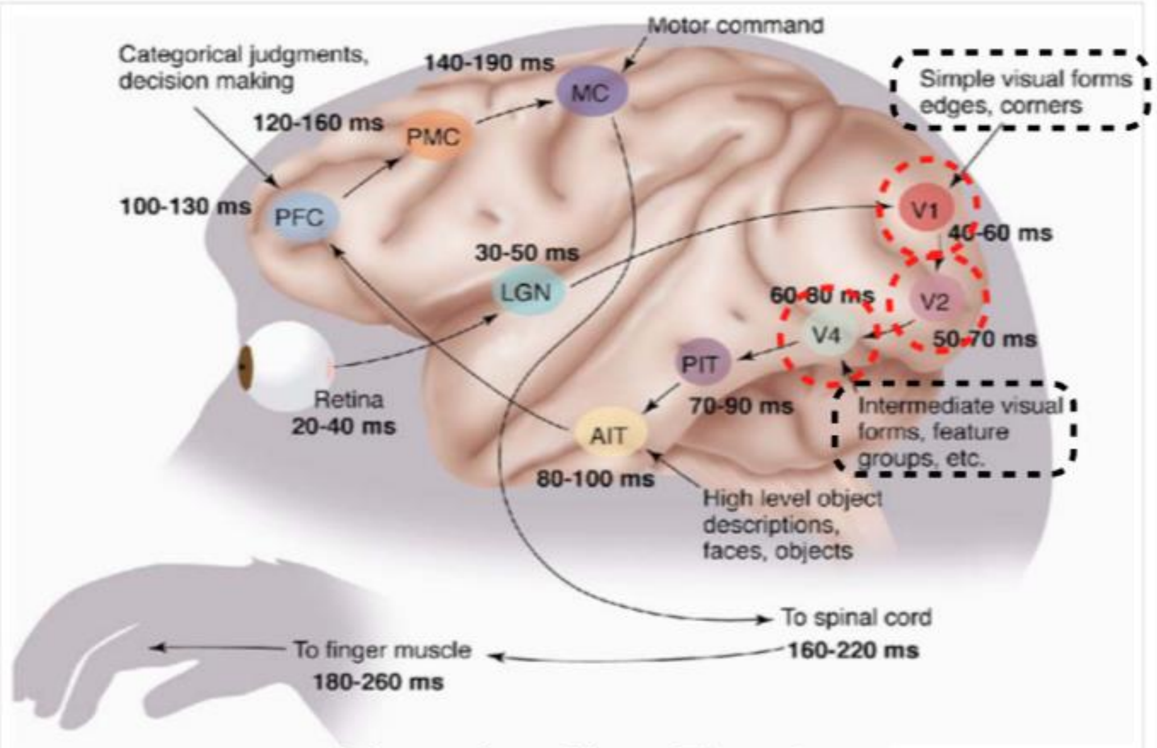


This massively parallel network also ensures that there is division of work

Each neuron may perform a certain role or respond to a certain stimulus

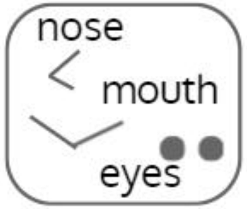


A simplified illustration



[picture from Simon Thorpe]

Layer 1: detect edges & corners



Layer 2: form feature groups

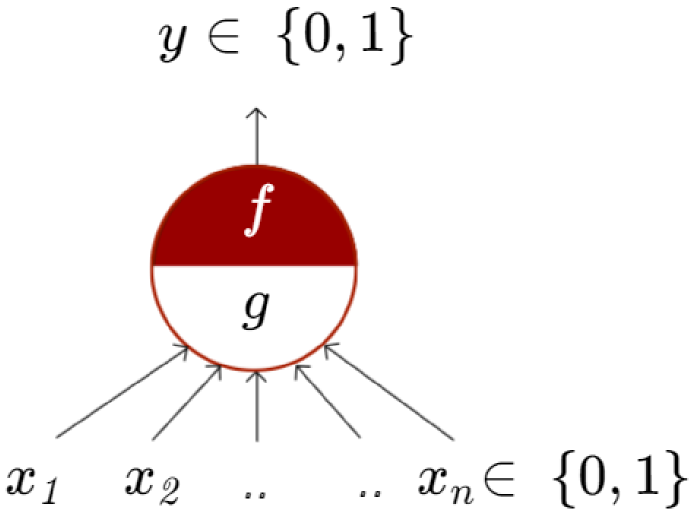
Layer 3: detect high level objects, faces, etc.

Sample illustration of hierarchical processing*

---

*Idea borrowed from Hugo Larochelle's lecture slides

McCulloch Pitts Neuron



This is called the Thresholding Logic

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

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

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

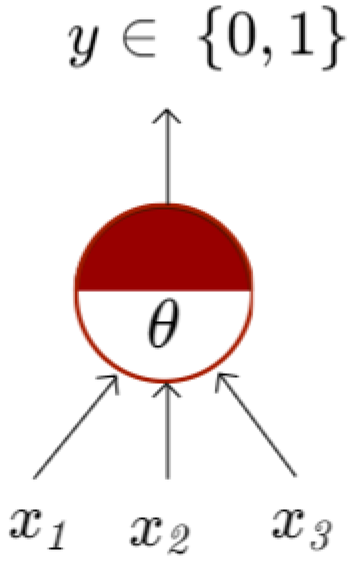
$$= 0 \quad \text{if} \quad g(x) < \theta$$

θ is called the thresholding parameter

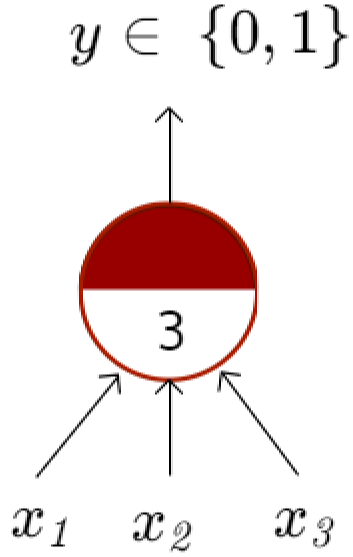
Characteristics of MP Neuron

- **Binary Inputs**: The inputs to the neuron are binary, meaning they can either be 0 or 1 (representing an inactive or active state, respectively).
- **Summation**: The neuron sums the weighted inputs. The total input to the neuron is the weighted sum of all inputs.
- **Threshold**:
- The neuron has a fixed threshold value. If the total input (weighted sum) exceeds or equals the threshold, the neuron "fires" (produces an output of 1); otherwise, it remains inactive (outputs 0).
- **Binary Output**:The output of the M-P neuron is binary, just like the input. If the sum of the weighted inputs exceeds the threshold, the output is 1; otherwise, it is 0.
- **No Learning**: The weights and threshold are predefined, and the neuron does not adapt or change with experience.

Implement AND Function

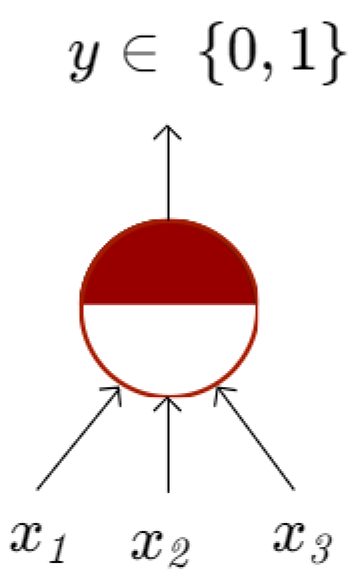


A McCulloch Pitts unit

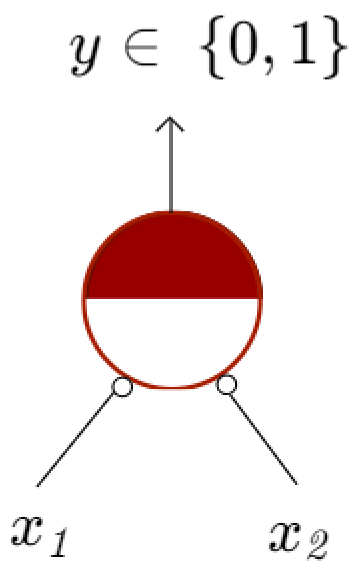


AND function

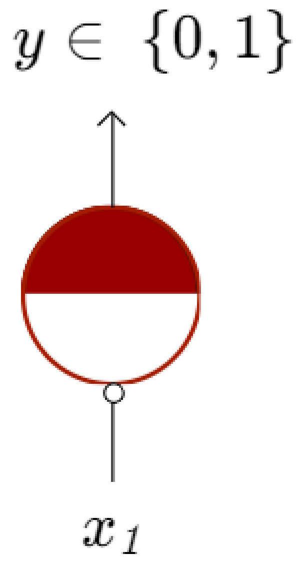
Logical Functions



OR function

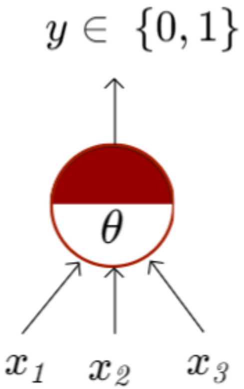


NOR function

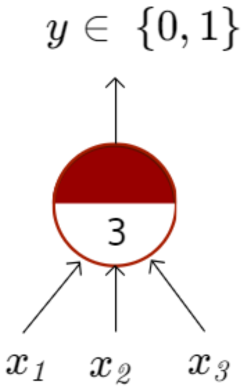


NOT function

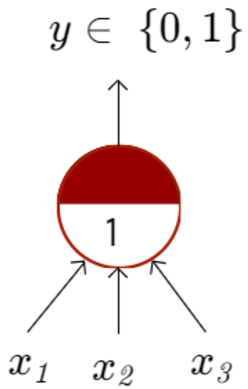
Logical Functions



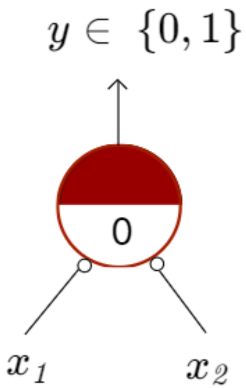
A McCulloch Pitts unit



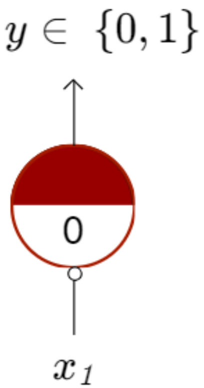
AND function



OR function



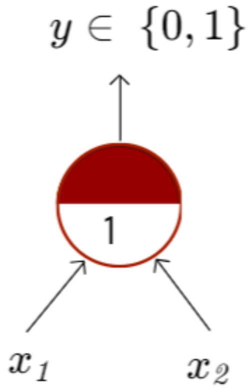
NOR function



NOT function

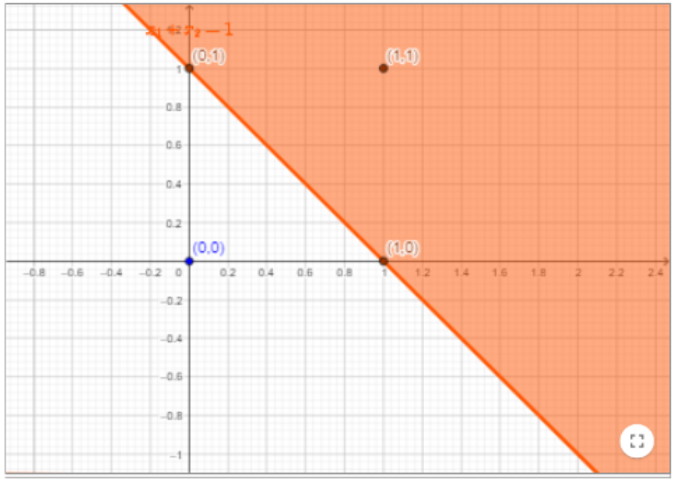
$y \in \{0,1\}$

Geometric interpretation of a MP unit



OR function

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

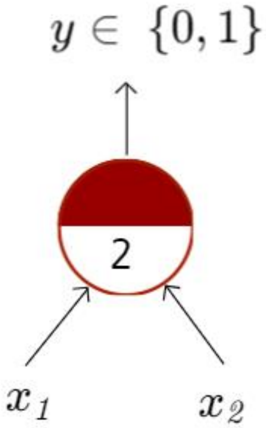


A single MP neuron splits the input points (4 points for 2 binary inputs) into two halves

Points lying on or above the line $\sum_{i=1}^{n} x_i - \theta = 0$ and points lying below this line

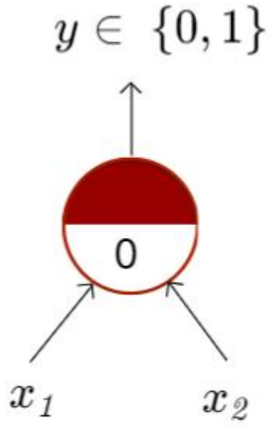
In other words, all inputs which produce an output 0 will be on one side ($\sum_{i=1}^{n} x_i < \theta$) of the line and all inputs which produce an output 1 will lie on the other side ($\sum_{i=1}^{n} x_i \geq \theta$) of this line

Geometric interpretation of a MP unit

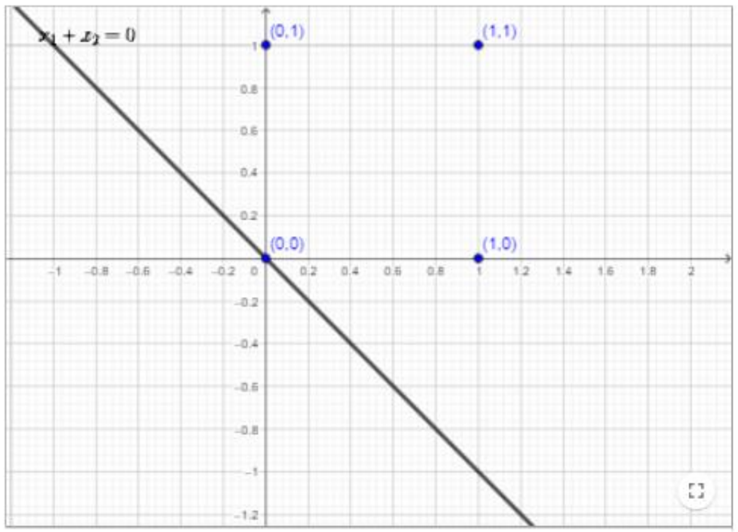


AND function

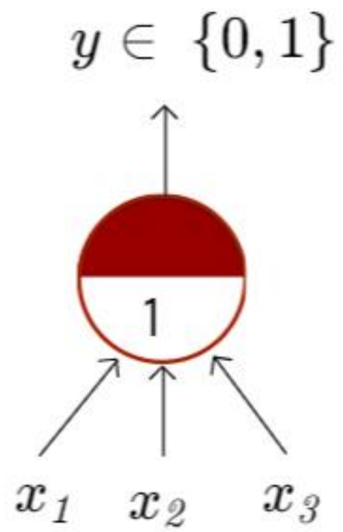
$$x_1 + x_2 = \sum_{i=1}^{2} x_i \geq 2$$



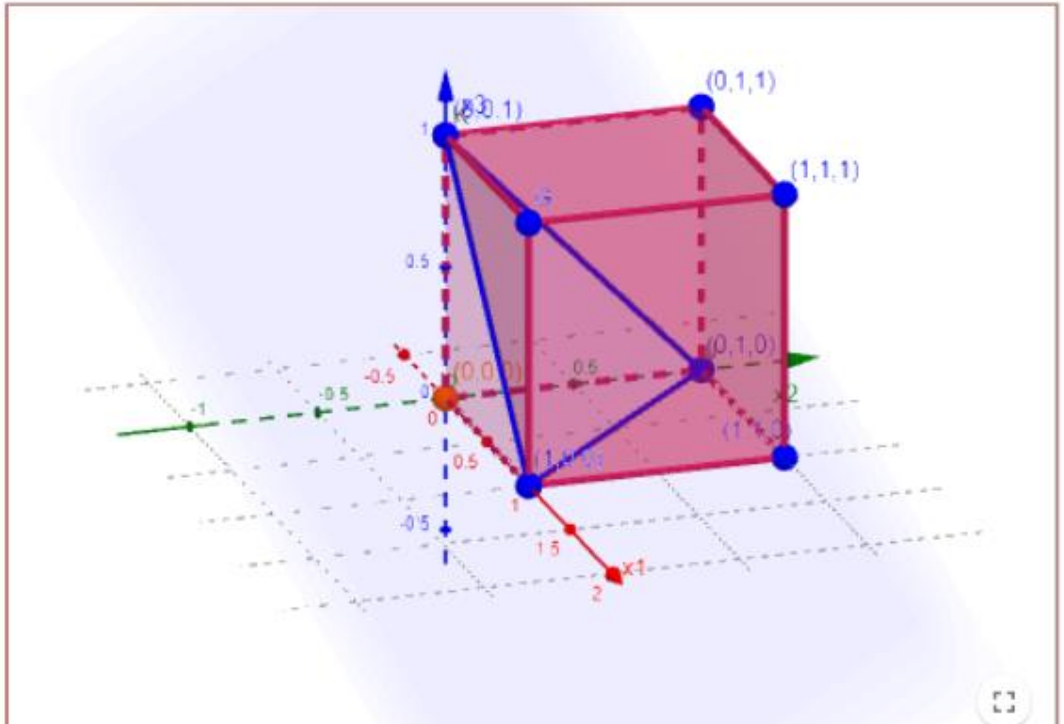
Tautology (always ON)



Geometric interpretation of a MP unit



OR



Example Scenario

- Imagine wanting to predict whether to watch a football game. The inputs (boolean values) could be:
- X1: Is Premier League on? (1 if yes, 0 if no)
- X2: Is it a friendly game? (1 if yes, 0 if no)
- X3: Are you not at home? (1 if yes, 0 if no)
- X4: Is Manchester United playing? (1 if yes, 0 if no)
- Each input can be excitatory or inhibitory. For instance, X3 is inhibitory because you can't watch the game at home.

https://www.analyticsvidhya.com/blog/2024/07/mcculloch-pitts-neuron/

Implementation

FOR AND GATE

```
#inputs
x1 = [0, 0, 1, 1]
x2 = [0, 1, 0, 1]
w1 = [1, 1, 1, 1]
w2 = [1, 1, 1, 1]
t = 2
#output
print("x1    x2    w1   w2     t     O")
for i in range(len(x1)):
    if ( x1[i]*w1[i] + x2[i]*w2[i] ) >= t:
        print(x1[i],'    ',x2[i],'    ',w1[i],'    ',w2[i],'    ',t,'    ', 1)
    else:
        print(x1[i],'    ',x2[i],'    ',w1[i],'    ',w2[i],'    ',t,'    ', 0)
```