
Lec 4: Basic Principles of McCulloch-Pitts Neuron

Outline:

1. **McCulloch-Pitts Neuron Model, Architecture and Algorithm**
2. **Logic Gates**
3. **Solved Problems**

1. McCulloch-Pitts Neuron Model

A sample McCulloch-Pitts network is shown below and some of the statements can be observed in **Figure 1**

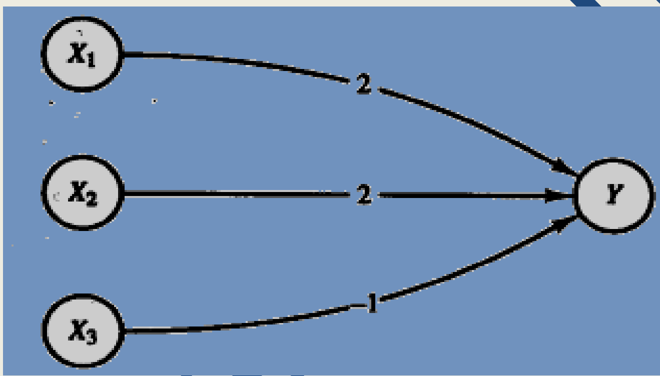


Figure 1: A simple McCulloch-Pitts model neuron Y

In particular, note that the threshold for **Y has equal 4 as** this is the only value that allows it to fire, taking into account that a neuron cannot fire if it receives a nonzero inhibitory input. Using the **McCulloch-Pitts** model we can model logic functions

Note:

It takes one time step for the signals to pass from the X units to Y; the activation of Y at time t is determined by the activations of X_1, X_2, and X_3 at the previous time, t - 1. The use of discrete time steps enables a network of McCulloch-Pitts neurons to. Model physiological phenomena in which there is a time delay

1.1 Architecture

In general, a McCulloch-Pitts neuron Y may receive signals from any number of other neurons. Each connection path is either excitatory, with weight w > 0, or inhibitory, with weight - p (p > 0). For convenience, in Figure 3, we assume there are n units, X_1, ... ,Xn, which send excitatory signals to unit Y, and m units, X_{n+1}, ••• ,X_{n+m}, which send inhibitory signals. The activation function

$$f(y\_in) = \begin{cases} 1 & \text{if } y\_in \geq \theta \\ 0 & \text{if } y\_in < \theta \end{cases} \quad \text{……………………………..(1)}$$

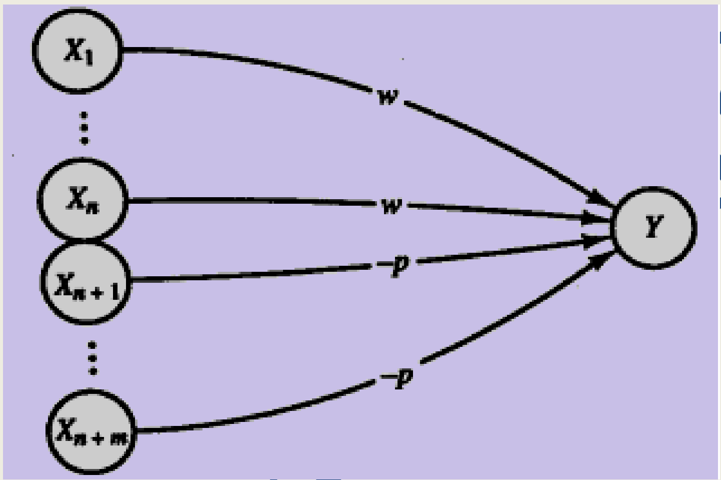


Figure 3: Architecture of McCulloch-Pitts model neuron Y

where y_in is the total input signal received and θ is the threshold. The condition that inhibition is absolute requires that θ for the activation function satisfy the inequality

$$\theta > nw - p.$$

Y will fire if it receives k or more excitatory inputs and no inhibitory inputs, where

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

Although all excitatory weights coming into any particular unit must be the same, the weights coming into one unit, say, Y_1, do not have to be the same as the weights coming into another unit, say Y_2.

1.2 Algorithm

The weights for a McCulloch-Pitts neuron are set, together with the threshold for the neuron's activation function, so that the neuron will perform a simple logic function. Since analysis, rather than a training algorithm, is used to determine the values of the weights and threshold, several examples of simple McCulloch-Pitts neurons are presented in this section. Using these simple neurons as building blocks, we can model any function or phenomenon that can be represented as a logic function.

2. Logic Gates

Logic functions will be used as simple examples for a number of neural nets. The binary form of the functions for AND, OR, and AND NOT are defined here for reference. Each of these functions acts on two input values. Denoted X_1 and X_2, and produces a single output value y.

2.1 AND Function

The AND function gives the response "true" if both input values are "true"; otherwise the response is "false." If we represent "true" by the value 1, and "false" by 0, this gives the following four training input, target output pairs:

x_1	x_2	→	y
1	1		1
1	0		0
0	1		0
0	0		0

The network in Figure 4 performs the mapping of the logical AND function. The threshold on unit Y is 2.

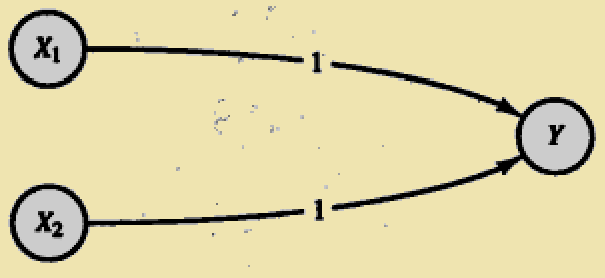


Figure 4 A McCulloch-Pitts neuron to perform the logical AND function.

2.2 OR Function

The OR function gives the response "true" if either of the input values is "true"; otherwise the response is "false." This is the "inclusive or," since both input values may be "true" and the response is still "true." Representing "true" as 1, and "false" as 0, we have the following four training input, target output pairs:

x_1	x_2	→	y
1	1		1
1	0		1
0	1		1
0	0		0

The network in Figure 5 the logical OR function. The threshold on unit Y is 2.

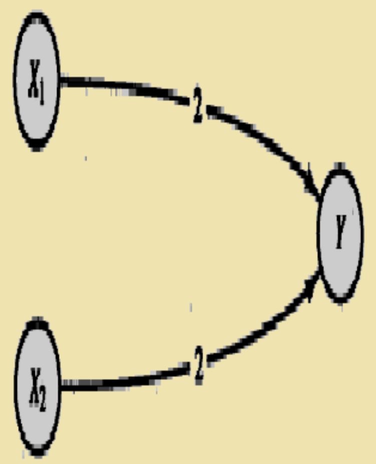


Figure 5 A McCulloch-Pitts neuron to perform the logical OR function.

2.3 AND NOT Function

The AND NOT function is an example of a logic function that is not symmetric in its treatment of the two input values. The response is "true" if the first input value, x_1, is "true" and the second input value, x_2, is "false"; otherwise the response is "false." Using a binary representation of the logical input and response values, the four training input, target output pairs are:

x_1	x_2	$\rightarrow$	y
1	1		0
1	0		1
0	1		0
0	0		0

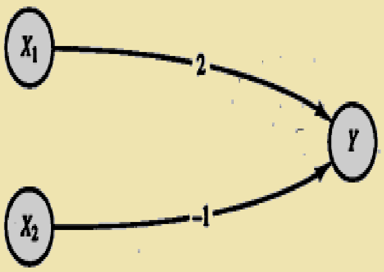


Figure 6 A McCulloch-Pitts neuron to perform the logical AND NOT function

The net in Figure 6 performs the function x_1 AND NOT x_2. In other words, neuron Y fires at time t if and only if unit X_1 fires at time $t-1$ and unit X_2 does not fire at time $t-1$. The threshold for unit Y is 2.

2.4 XOR Function

The XOR (exclusive or) function gives the response "true" if exactly one of the input values is "true"; otherwise the response is "false." Using a binary representation, the four training input, target output pairs are:

x_1	x_2	$\rightarrow$	y
1	1		0
1	0		1
0	1		1
0	0		0

The network in Figure 7 performs the logical XOR function. XOR can be expressed as

$$x_1 \text{ XOR } x_2 \leftrightarrow (x_1 \text{ AND NOT } x_2) \text{ OR } (x_2 \text{ AND NOT } x_1).$$

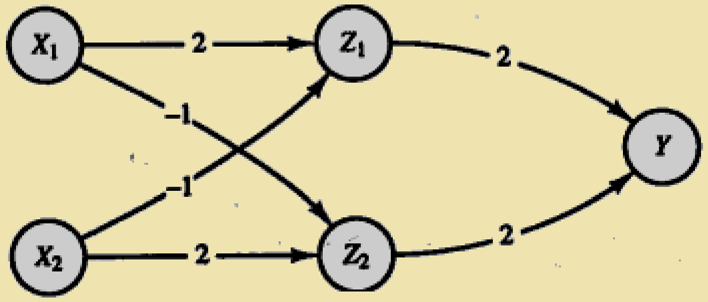


Figure 7 A McCulloch-Pitts neural net to perform the logical XOR function.

Thus, $y = x_1$ XOR x_2 is found by a two-layer net. The first layer forms

$$z_1 = x_1 \text{ AND NOT } x_2$$

and

$$z_2 = x_2 \text{ AND NOT } x_1.$$

The second layer consists of

$$y = z_1 \text{ OR } z_2.$$

Units Z_1, Z_2, and Y each have a threshold of 2.

3. Solved Problems

Q1: Design AND gate function using McCulloch-Pitts Neuron

Q2 Design OR gate function using McCulloch-Pitts Neuron

Q3: Design AND NOT gate function using McCulloch-Pitts Neuron

Q4: Design NOR gate function using McCulloch-Pitts Neuron

Q2:

The Mc culloch-Pitts activation function:

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

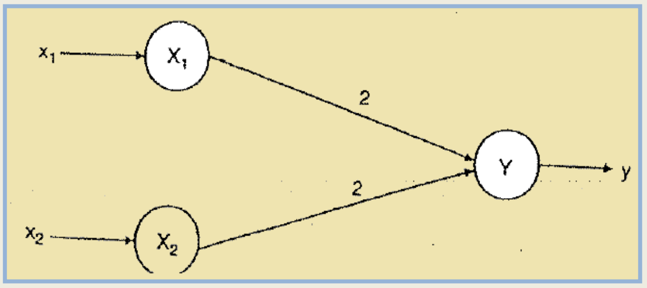
OR gate design

Truth table of OR gate

X_1	X_2	y
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**1**

The threshold of Y neuron is 2 (θ=2)

$$f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} \geq 2 \\ 0 & \text{if } y_{in} < 2 \end{cases}$$



Mcculloch-Pitts Neuron

For checking

Where $y_{in} = 2x_1+2x_2$

(a) For $x_1=x_2=1$

$y_{in} = 4$, $f(y_{in})=1$

(b) For $x_1=1$, $x_2=0$

$y_{in} = 2$, $f(y_{in})=1$

(c) For $x_1=0$, $x_2=0$

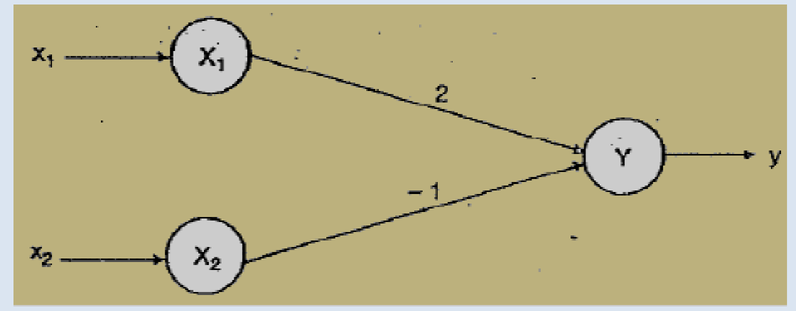
$y_{in} = 0$, $f(y_{in})=0$

Q3:

The threshold of Y neuron is 2 ($\theta=2$)

$$f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} \geq 2 \\ 0 & \text{if } y_{in} < 2 \end{cases}$$

The logic gate type is **ANDNOT** Gate



Truth table of ANDNOT gate

X_1	X_2	y
0	0	0
0	1	0
1	0	1
1	1	0

For checking

$y_{in} = 2x_1 - x_2$

(a) for $x_1=x_2=0$

$y_{in} = 0, f(y_{in})=0$

(b) for $x_1=0, x_2=1$

$y_{in} = -1, f(y_{in})=0$

(c) for $x_1=1, x_2=0$

$y_{in} = 2, f(y_{in})=1$

(d) for $x_1=1, x_2=1$

$y_{in} = 1, f(y_{in})=0$