The first in Era: McCulloch-Pitts Neuron

An artificial neuron is an information-processing system that has certain performance characteristics in common with biological neural networks. We follow certain assumptions:

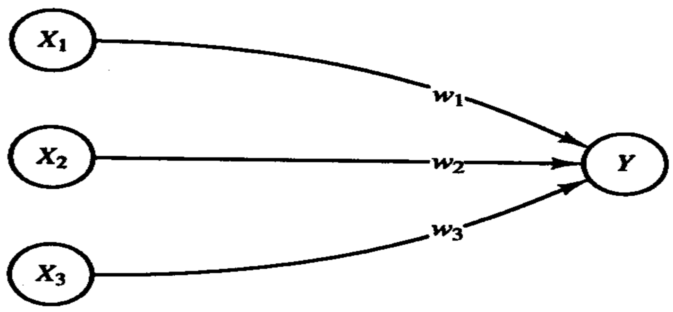
1. Information processing occurs at many simple elements called neurons.
2. Signals are passed between neurons over connection links.
3. Each connection link has an associated weight, which, in a typical neural net, multiplies the signal transmitted.
4. Each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) to determine its output signal.

A neural network is characterized by (1) its pattern of connections between the neurons, (2) its method of determining the weights on the connections (called its training, or learning, algorithm), and (3) its activation function.

Each neuron is connected to other neurons by means of directed communication links, each with an associated weight. The weights represent information being used by the net to solve a problem. Each neuron has an internal state, called its activation which is a function of the inputs it has received. A neuron sends its activation as a signal to several other neurons. It is important to note that a neuron can send only one signal at a time, although that signal is broadcast to several other neurons. Lets take one example for better understanding:

Let us assume one neuron Y takes inputs from three neurons X1, X2, and X3. The activation functions are x1, x2, x3 and weights associated with the neurons are w1, w2 and w3. The net input, y_in, will be:

$$y\_in = w1x1 + w2x2 + w3x3$$

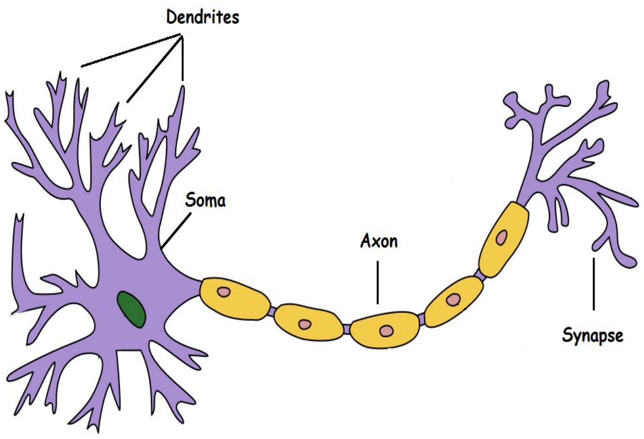


The activation function for y_in is given as

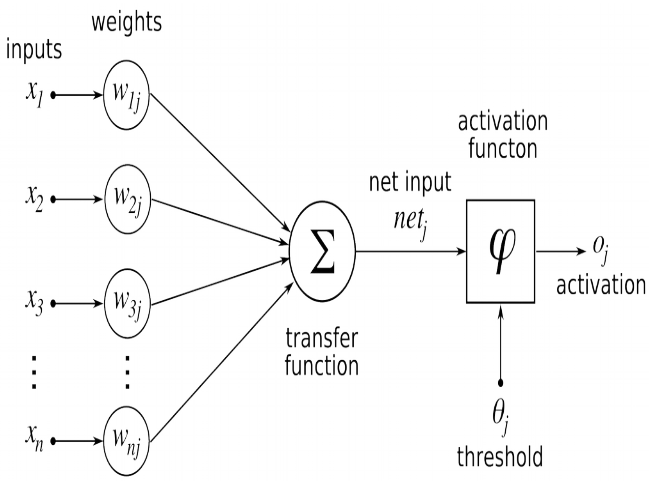
$$Y = f\ (y\_in)$$

The function ‘f (x)’ may be sigmoid, hyperbolic or any other, it depends on the type of input and output.

Now let’s have a look at the relationship between biological model and artificial model of neuron. A biological neuron has three types of components that are similar to that in artificial neuron: its dendrites, soma (cell body) and axon.



A Biological Neuron:[Wikipedia](#)



An artificial Neuron:[Wikimedia](#)

Biological Neuron	Artificial Neuron
Cell	Neuron
Dendrites	Weight or interconnections
Soma (Cell Body)	Net input
Axon	Output

Dendrites receive many signals from cells. The signals are electric impulses that are transmitted across a synaptic gap by means of chemical process, the modified signal by the action of chemical signal acts in a similiar fashion as of the function of weights in artificial neuron. The soma, or cell body, sums the incoming signals and the net input transmits a signal over its axon to other cells.

There are different types of neural nets and the earliest of them is McCulloch-Pitts Neuron, so let’s have a look on McCulloch-Pitts.

McCulloch Pitts Neuron

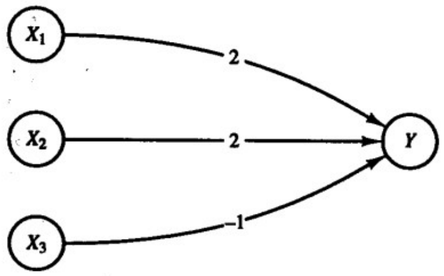
The McCulloch-Pitts neuron belongs to one of the earliest neural nets of the era. It was discovered by McCulloch and Pitts in 1943. It displays several important features found in later neural networks. There are certain requirements of McCulloch-Pitts Neuron which need to be fulfilled:

1. The activation function of the neuron is in binary format (either 0 or 1)
2. They are connected by directed and weighted paths.
3. If the weights are positive than it is called excitatory path or else inhibitory path. Positive paths possess same weights.
4. Each neuron has fixed threshold such that if the net input to the neuron is greater than the threshold than the neuron fires.
5. The threshold is set so that inhibiton is absolute, even if we include one inhibitory path than the neuron will not fire.
6. It take one time step for a signal to pass over one connection link.

The activation function for a unit Y is

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

here θ is the threshold limit.



Suppose lets look at the above example,
Here, our Y has the threshold limit of 4, so it will take X_1 and X_2 both collectively to go to fire Y and if X_3 is also considered than it leads to inhibition, this also brings us to conclusion that

$\theta > nw - p$

n = no. of positive weights
w = positive weight
p = inhibitor weight

The weights for MPN are set, together with threshold for the neuron's activation function, so that the neuron will perform a simple logic function. Simple MPNs have generally threshold of 2. The binary of AND, OR and AND NOT.

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

AND

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

OR

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

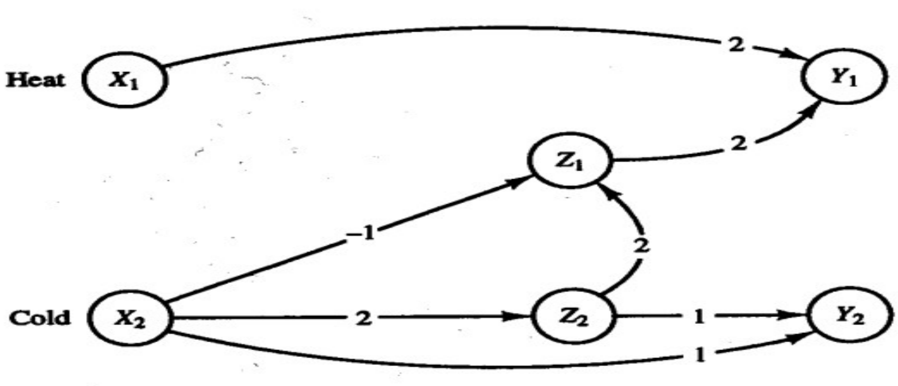
AND NOT

AND NOT is the special case of logic function in which response is true only if x1 is true and x2 is false and the response is false in rest of the cases.

Let's dive deep into MPN by taking one application,

Modelling the Perception of Hot and Cold with MPN

It is well-known and interesting physiologiocal phenomenon that if a cold stimulus is applied to a person's skin for a very short period of time, the person will perceive heat. However, if the same stimulus is applied for longer period, the person will perceive cold.

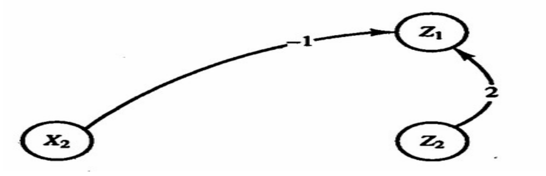


The use of discrete time steps enables the network of MPN to work. Here, X1 and X2 are receptors for hot and cold respectively, and the neurons Y1 and Y2 are the counterpart receptors. Z1 and Z2 are the intermediate states in the process.

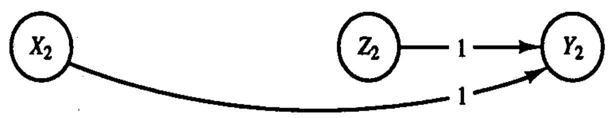
Cold:
We start building for cold first, it goes from X2 to Z2 with weight 2, depicting the intermediate trasition (one time step, small duration). We know that for a short period of time cold receptor perceives hot and if:

i. Cold is cutoff than it continues to perceive hot.
ii. Cold is not cutoff than it perceives cold in next time stamp.

Now Z2 will have two options Y1 or Z1. So we will make one edge going to Z1 with weight 2. And another -1 weighted edge from X2 depicting (not cold and in the transition state).



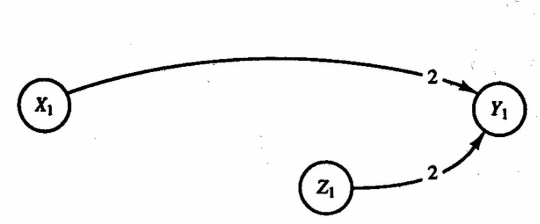
Neurons that determine the response of Z1



Neurons that determine the response of Y2

And for cold receptor Y2 to perceive cold, X2 must remain 1(cold) and currently, we are in the transition state Z2, so we use AND function here with weights 1 from both X2 and Z2 meeting at Y2 taking in consideration both the cases that X2 must be cold and Z2 must be on.

Hot:
Now, we look at the hot receptor Y1, it can be true for two conditions, (1) it directly receives hot (X1) from start and (2) it receives hot from Z1, so we implement an OR function for the required function.



Lets have a look at couple of examples: *(Assuming 1 for cold and 0 for hot)*

Input is 1111

- For 1st layer, we perceive hot, we get the input as 1 so it goes into X2 than it goes to intermediate state Z2, it outputs the result hot for 1st time
- Second time, we perceive cold as we check the X2, here its still in touch with cold, so we get cold and Y2 as output
- Similarly for 3rd and 4th layers, we get cold.
- So final output hot, cold, cold, cold.

Input is 1010

- For 1st layer, we perceive hot, we get the input as 1 so it goes into X2 than it goes to intermediate state Z2, it outputs the result hot for 1st time
- Second time, we again perceive hot as we check the X2, here contact is not in touch, so we get hot and Y1 as output
- Now again we perceive hot, this time the network resets itself and checks again the first condition
- Same as the second time, we perceive hot.
- Final output is hot, hot, hot, hot.

Run yourself the above problem in jupyter notebook and for solution click here
https://github.com/udisinghania/hot_cold_perception/blob/master/hot_cold_final_interactive.ipynb