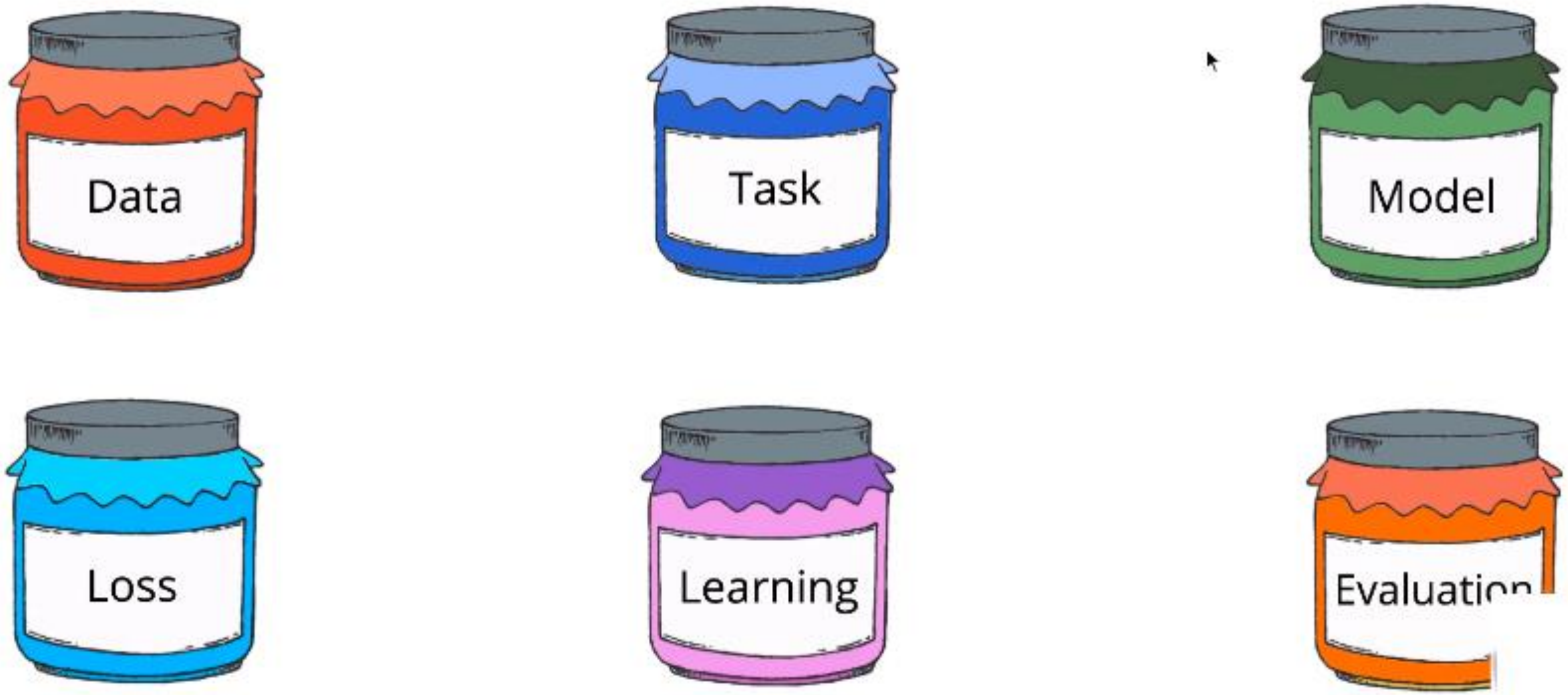
AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

Introduction to Deep Learning

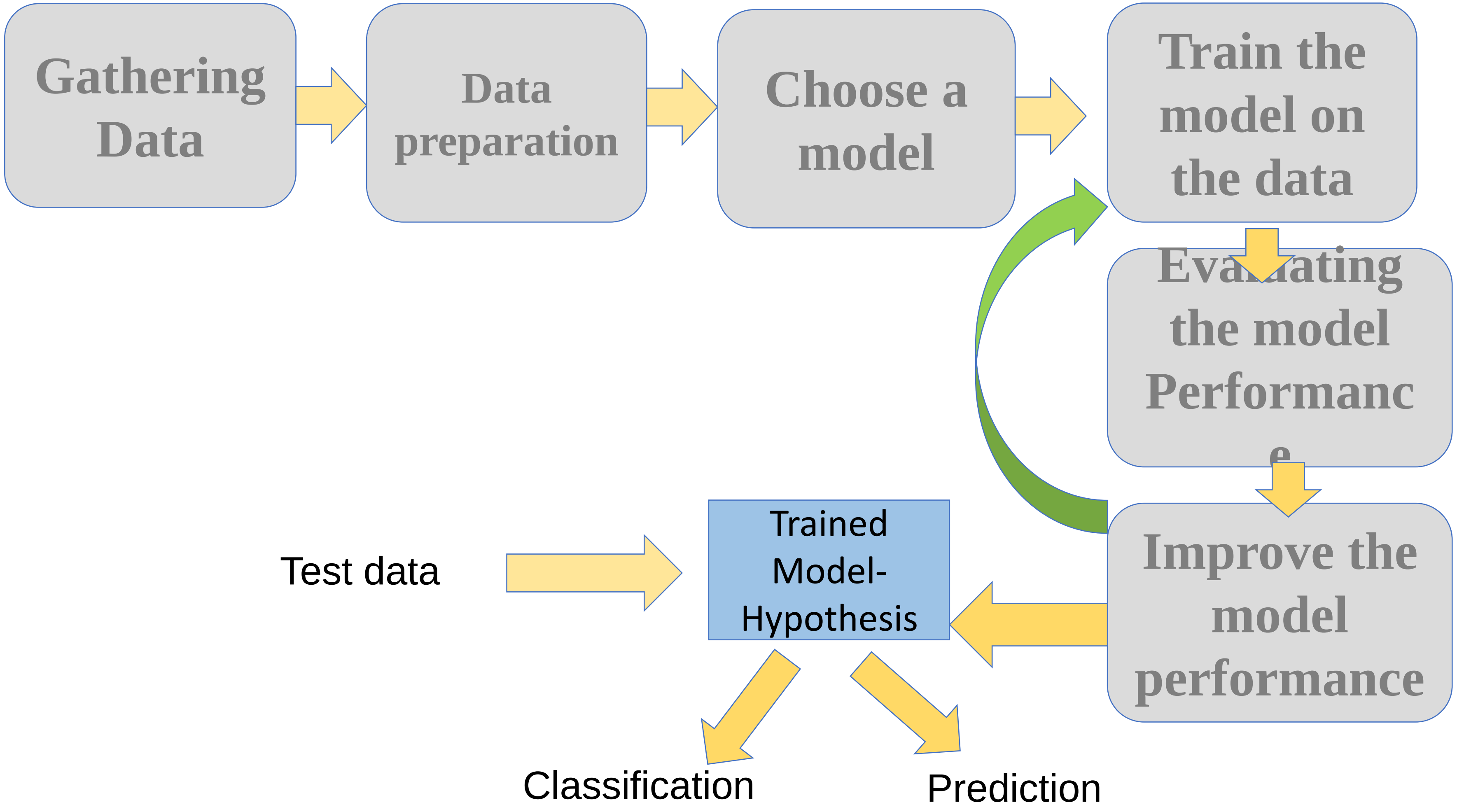
Introduction to Deep Learning

Citation Note: The concept, the content, and the structure of this article were inspired by the awesome lectures and the material offered by Prof. ***Mitesh M. Khapra*** *on NPTEL's Deep Learning course*

Ingredients- ML model



Recap :Supervised Learning



Recap

Id	Cl.thickness	Cell.size	Cell.shape	Marg.adhesion	Epith.c.size	Bare.nuclei	Bl.cromatin	Normal.nucleoli	Mitoses	Class
1000025	5	1	1	1	2	1	3	1	1	benign
1002945	5	4	4	5	7	10	3	2	1	benign
1015425	3	1	1	1	2	2	3	1	1	benign
1016277	6	8	8	1	3	4	3	7	1	benign
1017023	4	1	1	3	2	1	3	1	1	benign
1017122	8	10	10	8	7	10	9	7	1	malignant
1018099	1	1	1	1	2	10	3	1	1	benign
1018561	2	1	2	1	2	1	3	1	1	benign
1033078	2	1	1	1	2	1	1	1	5	benign
1033078	4	2	1	1	2	1	2	1	1	benign
1035283	1	1	1	1	1	1	3	1	1	benign
1036172	2	1	1	1	2	1	2	1	1	benign
1041801	5	3		3	2	3	4	4	1	malignant

Train the model with the above data set

Is my cancer Malignat or benign

NO it is benign

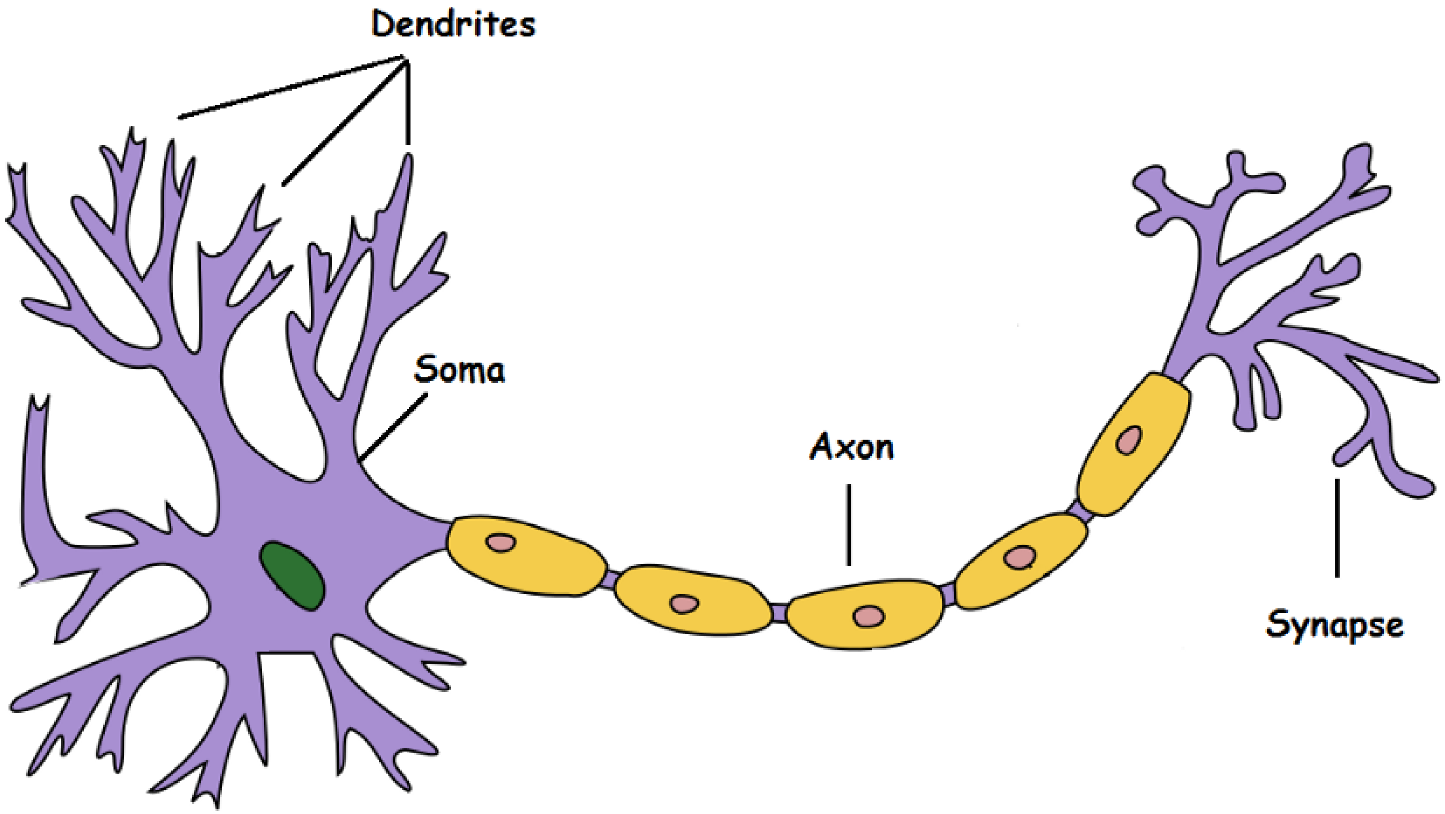
Test the model with a new real time input.

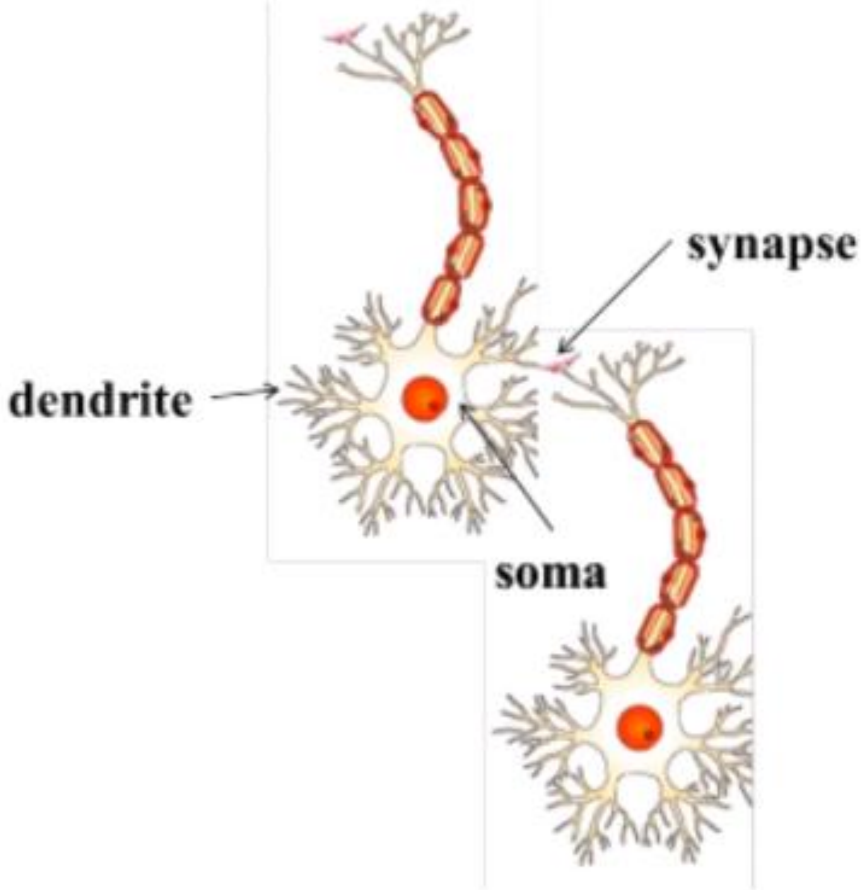
1041801	5	3	3	3	2	3	4	4	1

Biological Neuron

Takes an input, processes it, throws out an output.

Dendrite: Receives signals from other neurons
Soma: Processes the information
Axon: Transmits the output of this neuron
Synapse: Point of connection to other neurons





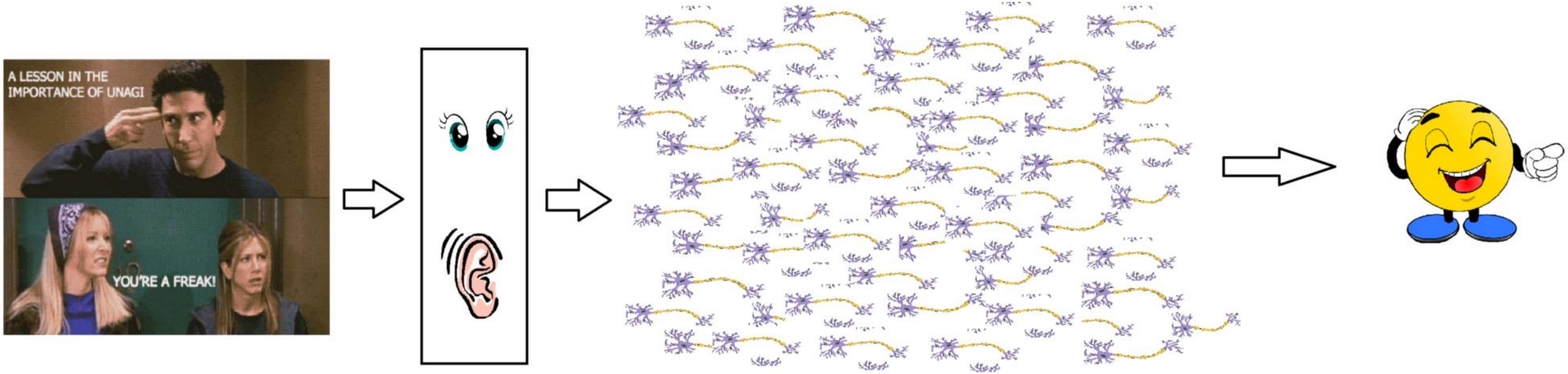
Courtesy: Towardsdatascience.com

Biological Neuron

Each neuron gets fired/activated only when its respective criteria is met

Our sense organs interact with the outer world and send the visual and sound information to the neurons

. Let's say you are watching a video clip. Now the information your brain receives is taken in by the **"laugh or not"** set of neurons that will help you make a decision on whether to laugh or not.

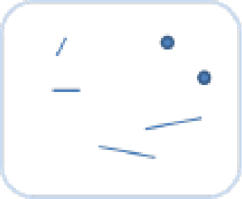


There is a massively parallel interconnected network of 10^{11} neurons (100 billion) in our brain and their connections are not simple

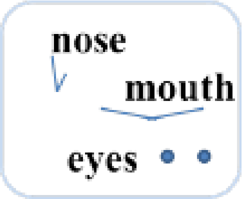
Courtesy: Towardsdatascience.com

Neurons are arranged in a hierarchical fashion

Neurons are arranged in a hierarchical fashion and each layer has its own role and responsibility.
To detect a face, the brain could be relying on the entire network and not on a single layer.

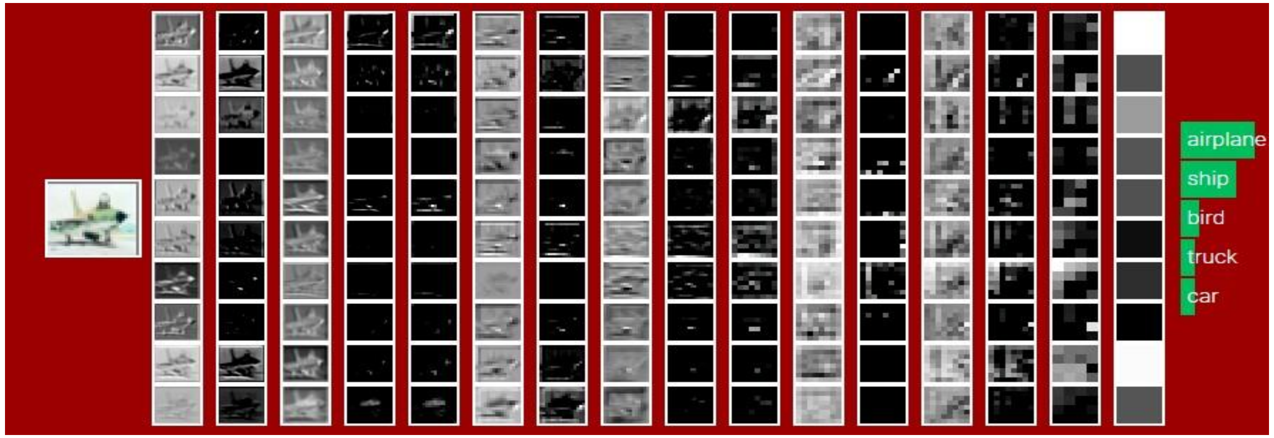


Layer 1: detect edges & corners



Layer 2: form feature groups

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



Courtesy :Sample illustration of hierarchical processing. Credits: Mitesh M. Khapra's lecture slides

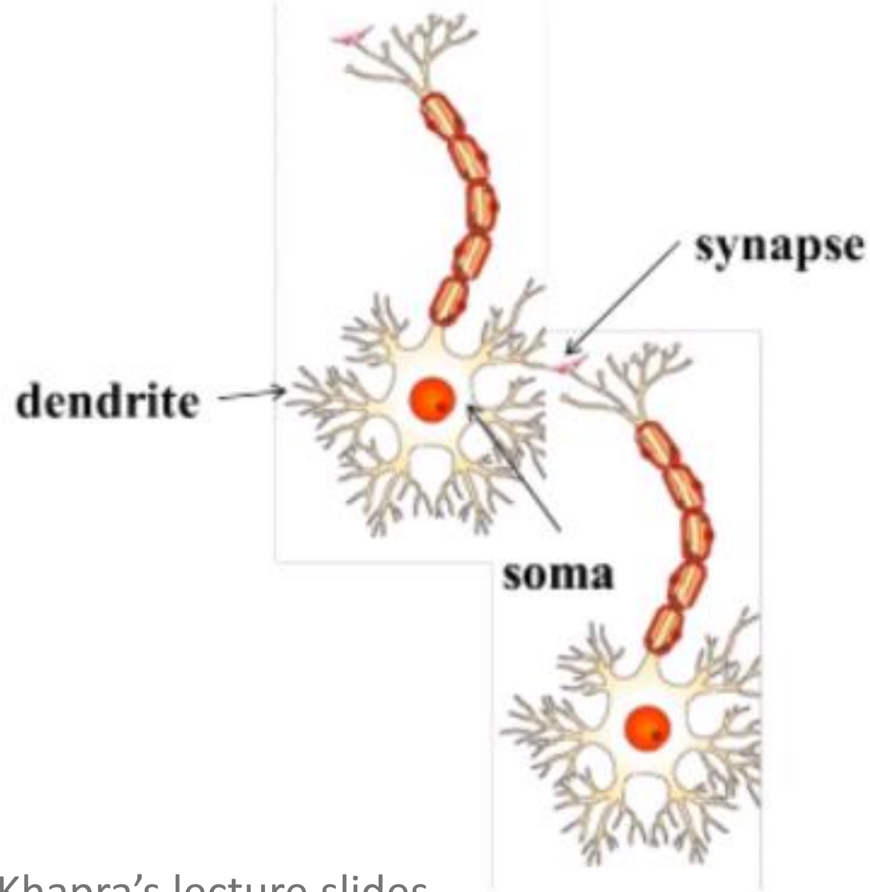
The very first step towards the perceptron we use today was taken in 1943 by McCulloch and Pitts, by mimicking the functionality of a biological neuron.

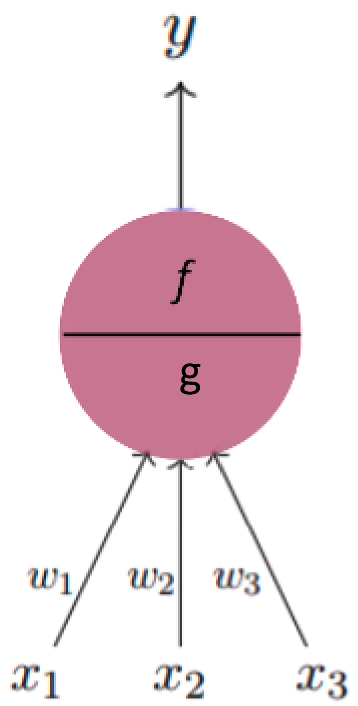
Recall Biological Neuron

Artificial Neuron

The fundamental Building block of Deep Learning

The most fundamental unit of deep neural networks is called an *artificial neuron/perceptron*





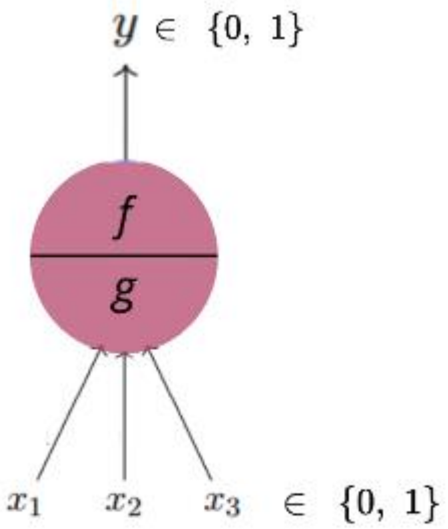
Courtesy :: Mitesh M. Khapra's lecture slides

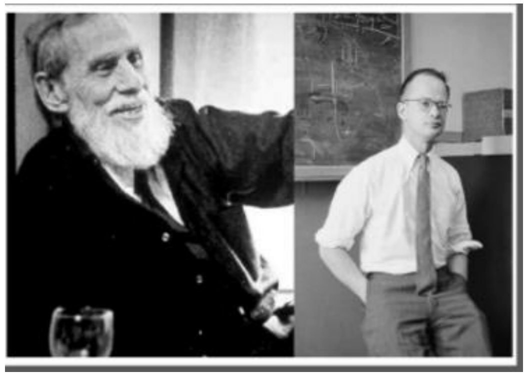
Artificial Neuron

“threshold logic” to mimic the thought process

The fundamental Building block of Deep Learning

Artificial Neural Networks





McCulloch-Pitts Neuron

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

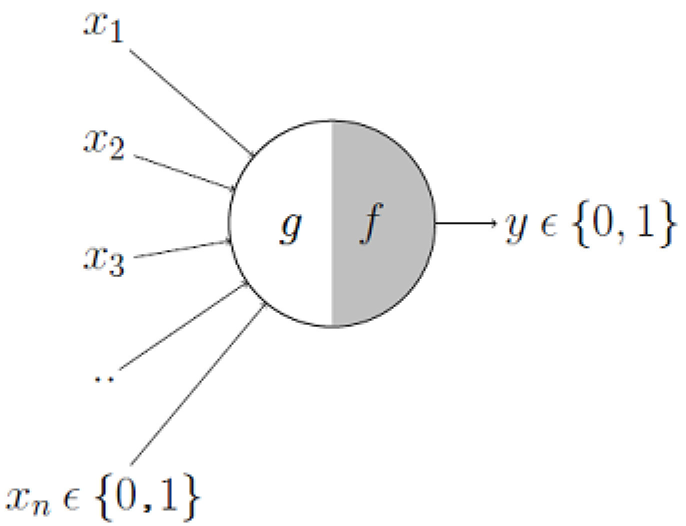
It may be divided into 2 parts. The first part, ***g*** takes an input (ahem dendrite ahem), performs an aggregation and based on the aggregated value the second part, ***f*** makes a decision.

Lets suppose that I want to **predict my own decision**, **whether to watch a random football game or not on TV?.**

The inputs are all boolean i.e., {0,1} and my 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.

Excitatory or inhibitory inputs.



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

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

These inputs can either be *excitatory* or *inhibitory*.

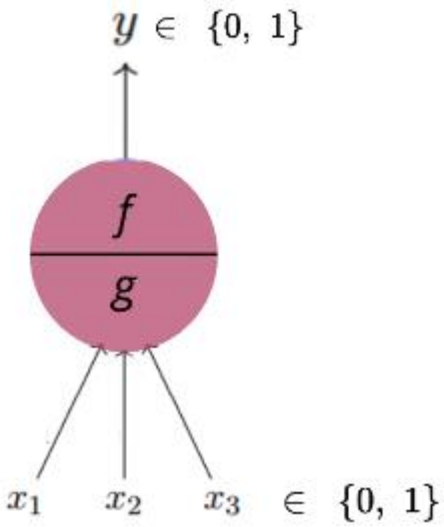
Inhibitory inputs are those that have maximum effect on the decision making irrespective of other inputs

i.e., if ***x_3*** is 1 (not home) then my output will always be 0 i.e., the neuron will never fire, so ***x_3*** is an inhibitory input

Excitatory inputs are NOT the ones that will make the neuron fire on their own but they might fire it when combined together

The mathematical Model- McCulloch-Pitts Neuron (MP Neuron)

- McCulloch and Pitts proposed a highly simplified computational model of the neuron.
- ***g*** aggregates the inputs and the function ***f*** takes a decision based on this aggregation.
- The inputs can be excitatory or inhibitory



These inputs can either be *excitatory* or *inhibitory*

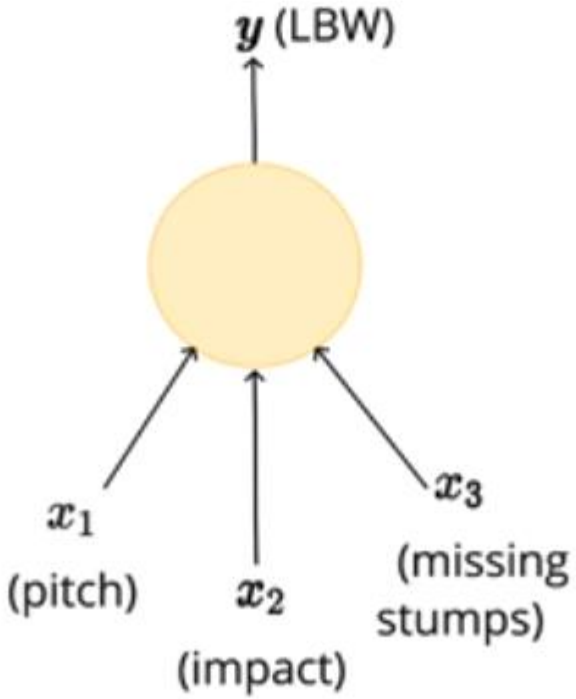
$y = 0$ if any x_i is inhibitory, else

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

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

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

What task can be done with MP Neuron



Pitch in line	Impact	Missing stumps	Is it LBW? (y)
1	0	0	0
0	1	1	0
1	1	1	1
0	1	0	[illegible]

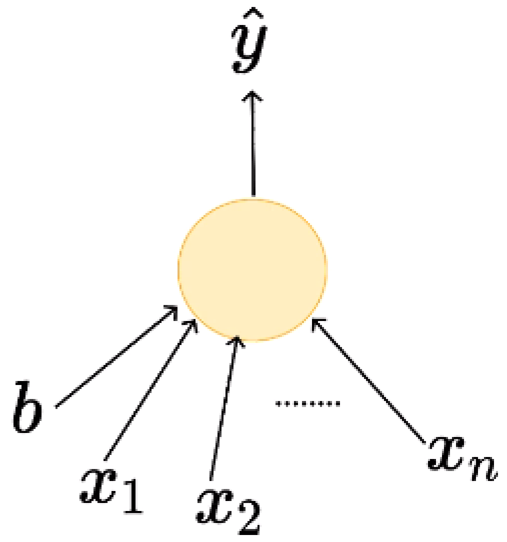
Courtesy :: Mitesh M. Khapra's lecture slides

Loss Function

We aim at minimizing the loss function

Launch (within 6 months)	0	1	1	0	0	1	0	1	1	0
Weight (<160g)	1	0	1	0	0	0	1	0	0	1
Screen size (<5.9 in)	1	0	1	0	1	0	1	0	1	0
dual sim	1	1	0	0	0	1	0	1	0	0
Internal memory (>= 64 GB, 4GB RAM)	1	1	1	1	1	1	1	1	1	0
NFC	0	1	1	0	1	0	1	1	1	0
Radio	1	0	0	1	1	1	0	0	0	0
Battery(>3500mAh)	0	0	0	1	0	1	0	1	0	0
Price > 20k	0	1	1	0	0	0	1	1	1	0
Like? (y)	1	0	**1**	**0**	1	1	**0**	**1**	0	0
prediction $\hat{y}$	1	0	**0**	**1**	1	1	**1**	**0**	0	0
loss	0	0	1	-1	0	0	-1	1	0	0

$$loss = \sum_i y_i - \hat{y}_i$$



$$\hat{y} = \sum_{i=1}^{n} x_i \geq b$$

$$\hat{y} = x_1 + x_2 \geq b$$

$$\boxed{loss = \sum_i (y_i - \hat{y}_i)^2}$$

Courtesy :: Mitesh M. Khapra's lecture slides

Training data

Launch (within 6 months)	0	1	1	0	0	1	0	1	1	0
Weight (<160g)	1	0	1	0	0	0	1	0	0	1
Screen size (<5.9 in)	1	0	1	0	1	0	1	0	1	0
dual sim	1	1	0	0	0	1	0	1	0	0
Internal memory (>= 64 GB, 4GB RAM)	1	1	1	1	1	1	1	1	1	0
NFC	0	1	1	0	1	0	1	1	1	0
Radio	1	0	0	1	1	1	0	0	0	0
Battery(>3500mAh)	0	0	0	1	0	1	0	1	0	0
Price > 20k	0	1	1	0	0	0	1	1	1	0
Like? (y)	1	1	1	0	0	1	1	1	0	0
predicted	1	1	0	1	1	1	1	0	0	0

Test data

1	0	0	1
0	1	1	1
0	1	1	1
0	1	0	0
1	0	0	0
0	0	1	0
1	1	1	0
1	1	1	0
0	0	1	0
0	1	0	0
0	1	1	$\boxed{0}$

✓ ✓ ✗ ✓

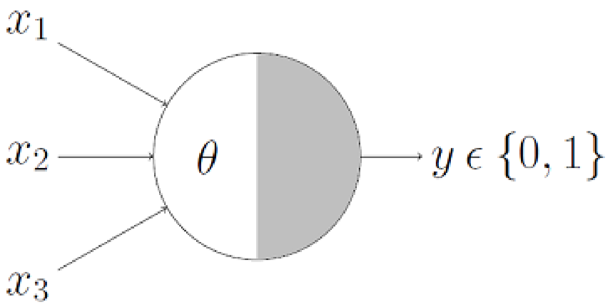
Learning

MODEL

Predictions

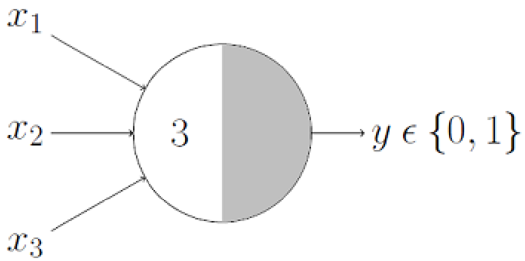
$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{3}{4} = 75\%$$

Boolean Functions Using M-P Neuron



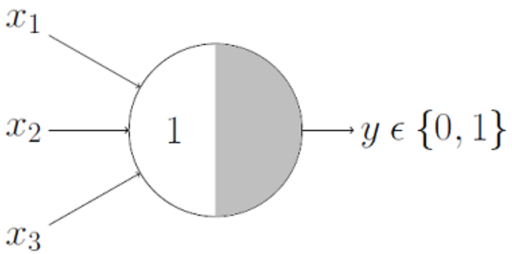
sum ≥ theta, the neuron will fire otherwise, it won’t.

AND Function



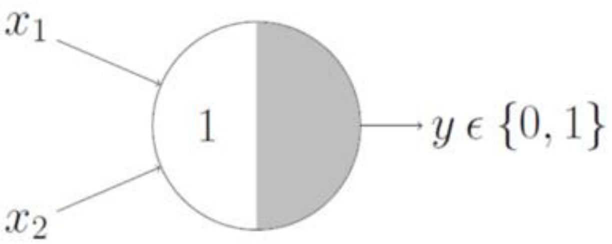
An AND function neuron would only fire when ALL the inputs are ON i.e., ***g*(x)** ≥ 3 here.

OR Function



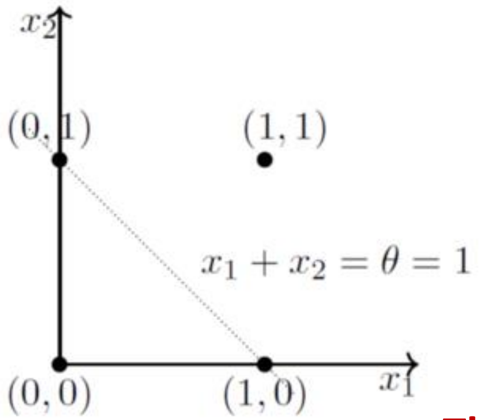
neuron would fire if ANY of the inputs is ON i.e., ***g*(x)** ≥ 1 here.

Geometric Interpretation OR function



OR function

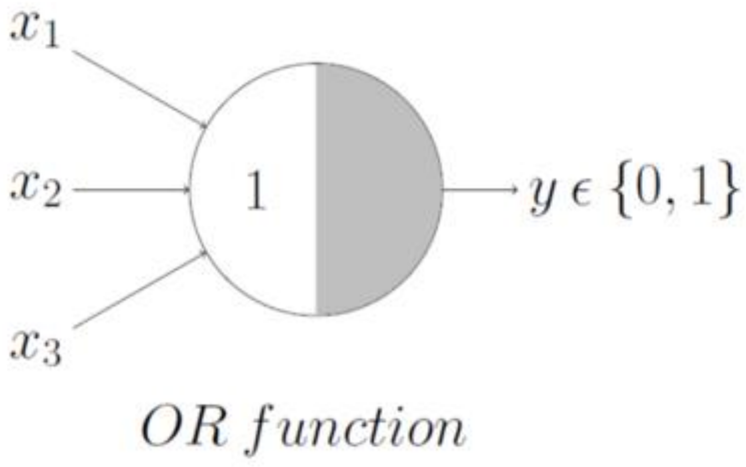
$$x_1 + x_2 = \sum_{i=1}^{2} x_i \geq 1$$



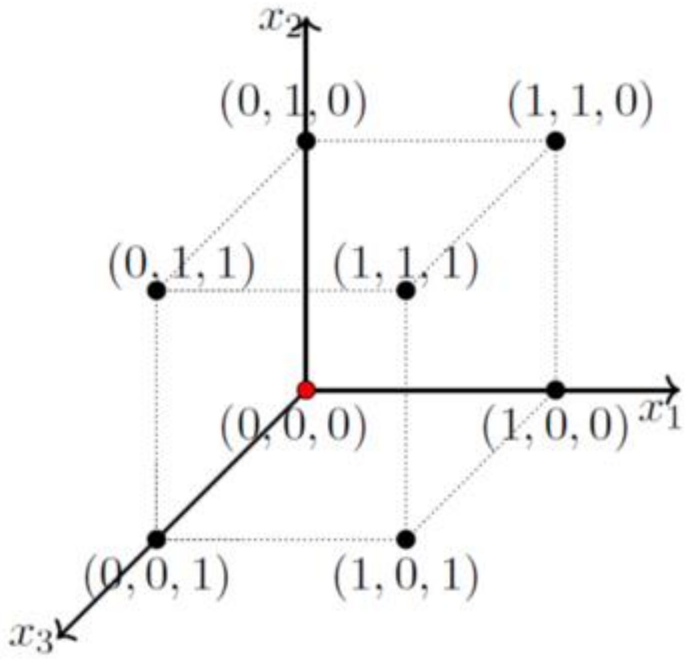
***x_1 + x_2 =*1** to graphically show that all those inputs whose output when passed through the OR function M-P neuron lie ON or ABOVE that line and all the input points that lie BELOW that line are going to output 0.

The M-P neuron just learnt a linear decision boundary!

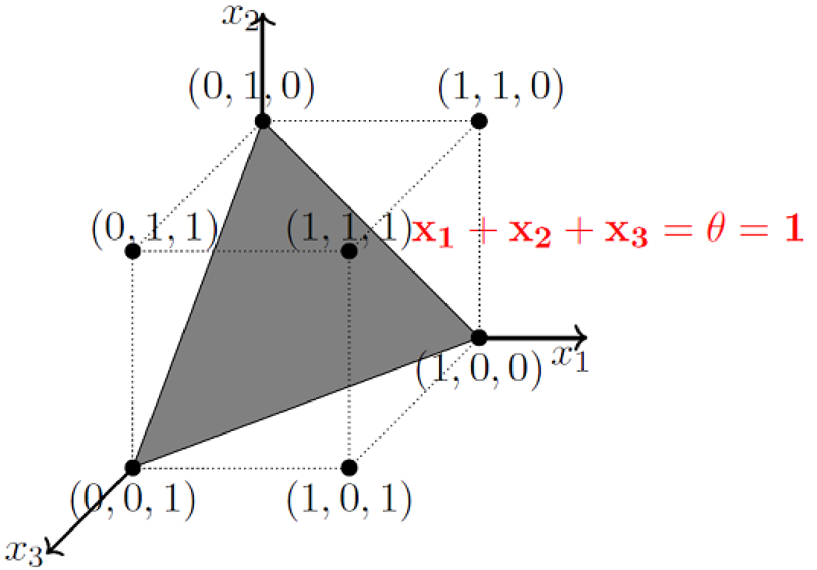
OR Function With 3 Inputs



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



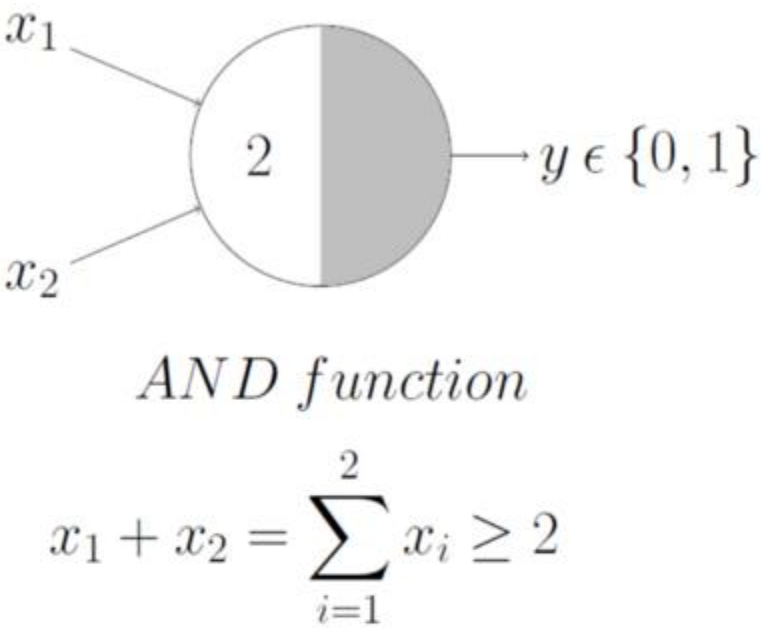
The plane that satisfies the decision boundary equation ***x_1 + x_2 + x_3 = 1*** is shown below:

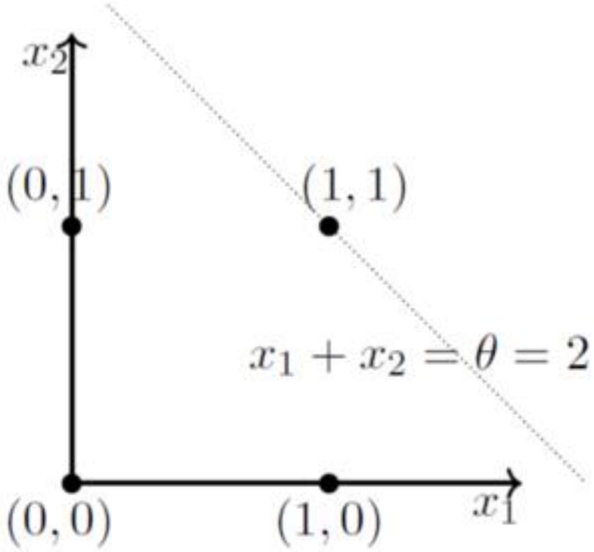


Lets just generalize this by looking at a 3 input OR function M-P unit. In this case, the possible inputs are 8 points — (0,0,0), (0,0,1), (0,1,0), (1,0,0), (1,0,1),… you got the point(s). We can map these on a 3D graph and this time we draw a decision boundary in 3 dimensions.

Boolean Functions Using M-P Neuron

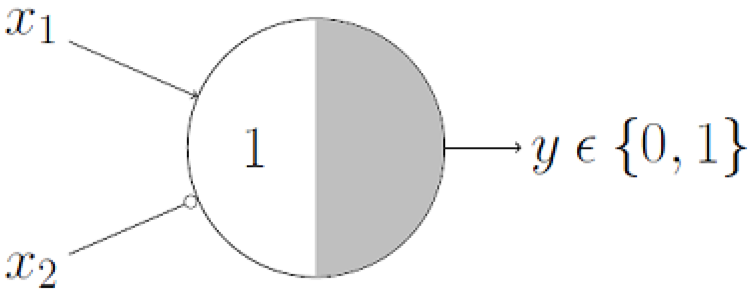
AND Function





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!

Boolean Functions Using M-P Neuron



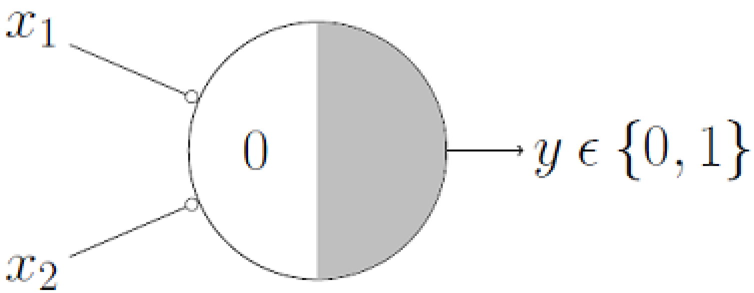
$x_1 \ AND \ !x_2{}^*$

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

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.

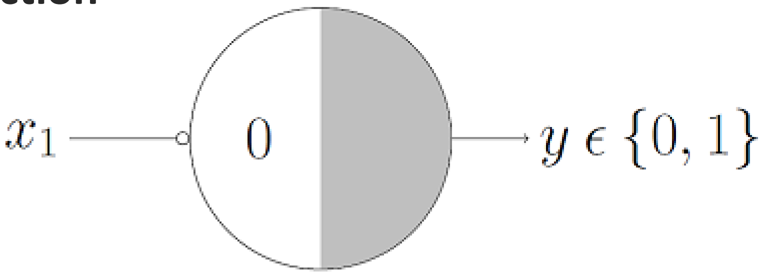
g*(x)** i.e., ***x_1 + ***x_2*** would be ≥ 1

NOR Function



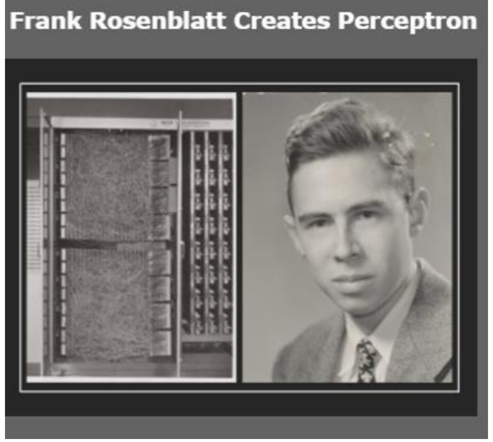
For a NOR neuron to fire, we want ALL the inputs to be 0 so the thresholding parameter should also be 0 and we take them all as inhibitory input.

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!

Limitations Of M-P Neuron



Frank Rosenblatt

- 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.

Overcoming the limitations of the M-P neuron, **Frank Rosenblatt**, an American psychologist, proposed the classical **perception model**, the mighty *artificial neuron*, in 1958. It is more generalized computational model than the McCulloch-Pitts neuron where weights and thresholds can be learnt over time

Namah Shivaya