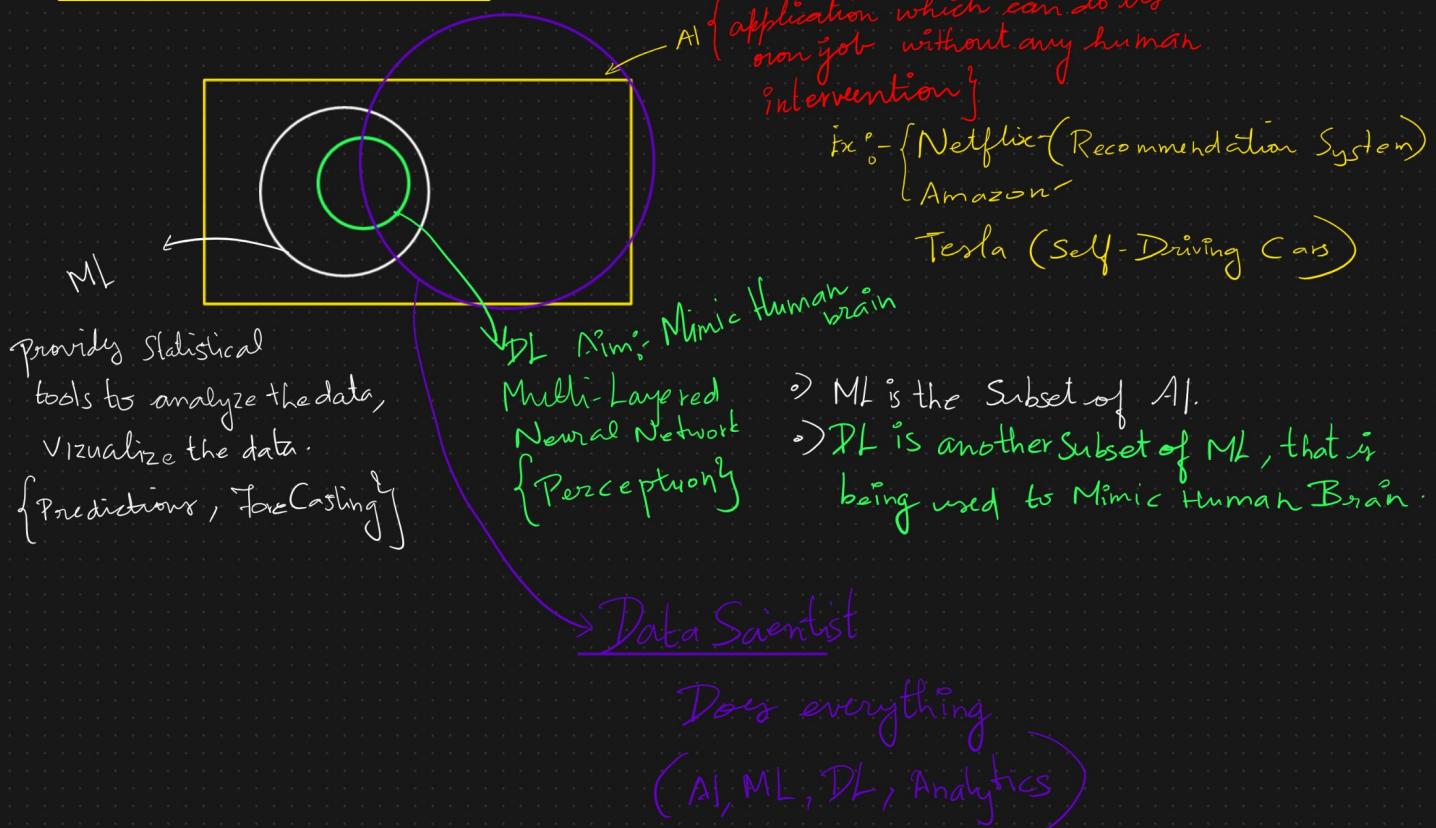


# Deep Learning (Day -1)

## Why Data Scientists are paid very much?

AI vs ML vs DL vs DS



## Why Deep Learning is becoming Famous?

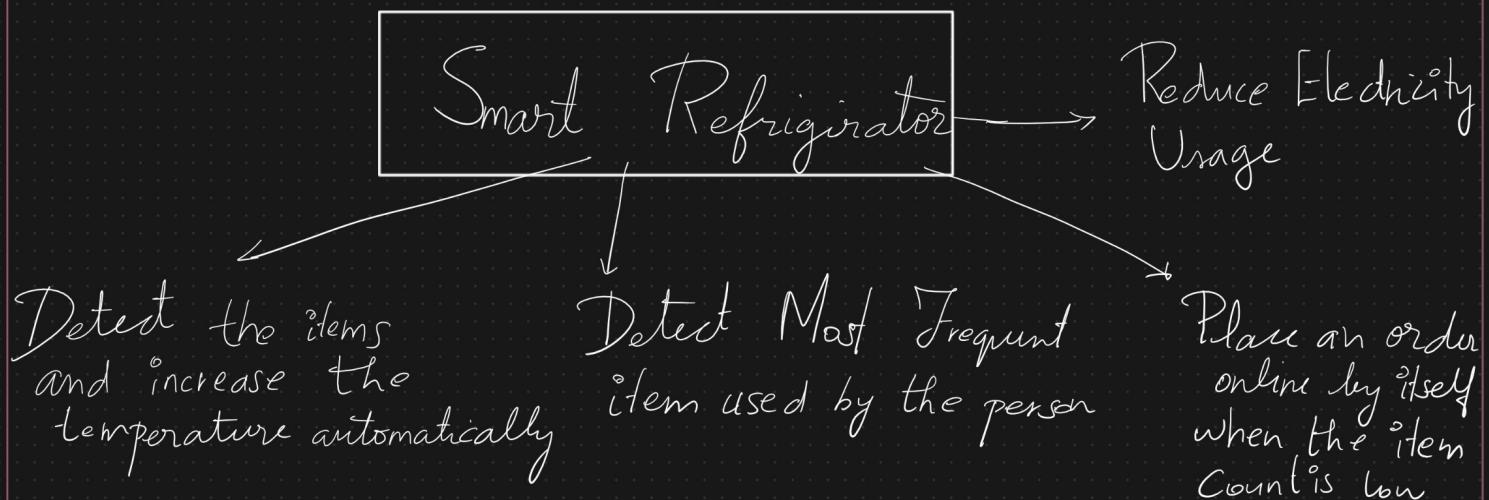
- ① 2005 → Facebook, Instagram, Twitter, LinkedIn, WhatsApp were into action. And Started Generating Huge Chunks of Data. (Exponential Increase)
- ② 2008 → {Big Data} → Became Popular,
- ③ 2013 → Companies had huge amount of Data  
{Companies wanted to use it and make their Product/Service more efficient}

That's When Data Science Started Growing.

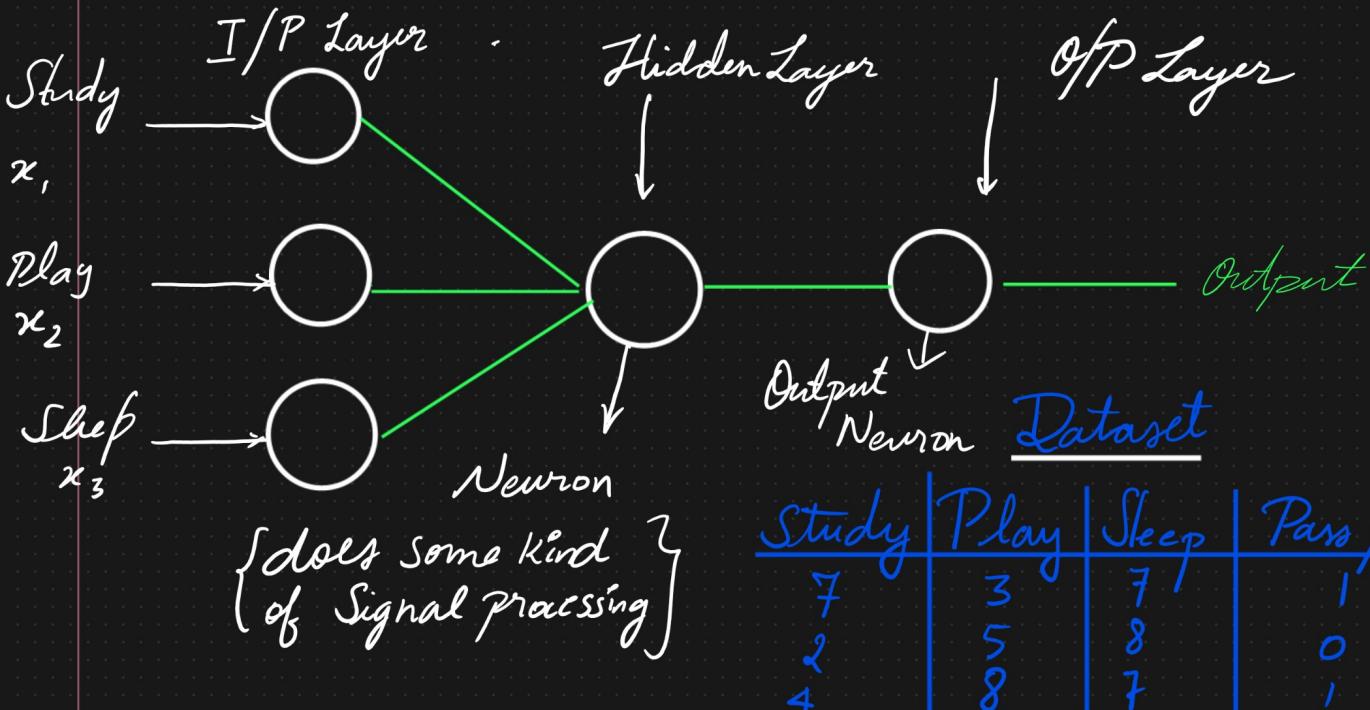
Panasonic → TV's, Refrigerator → Generated Data

↓  
Model that could use the data generated from the Device and reduce the electricity consumed by the model.

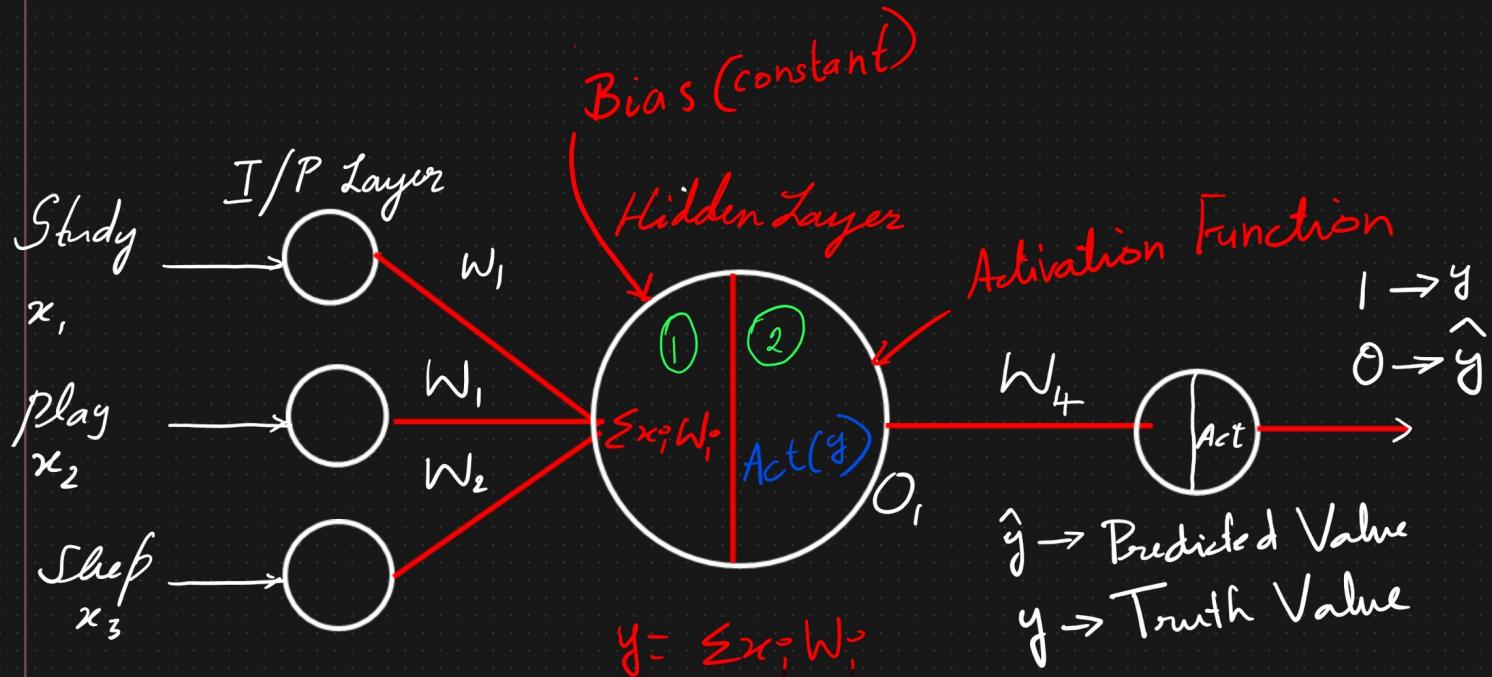
Smart Refrigerator → Use Object detection model to maintain livestock and automatically place an order online.



## \* Perception { Single Layered Neural Network }



Study	Play	Sleep	Pass / Fail
7	3	7	1
2	5	8	0
4	8	7	1



Loss Function  $\{y - \hat{y}\}$  Forward Propagation

Multiply Inputs with weights + bias → Activate

Aim of Loss function is to Compare Predicted and Actual Value.

The value of Loss function must be always closer to Zero.

$$(y - \hat{y}) = 1 - 0 = 1$$

{Output is completely wrong}

Back Propagation

Update the Weights to get desired ' $\hat{y}$ ' value  
{depends on optimizers}

$$\begin{aligned}\sum x_i w_i &= x_1 w_1 + x_2 w_2 + x_3 w_3 \\ &= W^T x\end{aligned}$$

Weights will help the neurons to act whether it should be activated, what level it should be activated.

During the training of the Neural Networks also we will be taking care that which weight should have what value so that the neuron should be activated from that level onwards.

- Weights basically say how much the Neuron should get activated or deactivated -
- Initially when the weights are passed as zero, there is no use.  
To overcome that we use bias.

## Sigmoid Activation Function

Activation Function decides which neuron to activate or deactivate.

$$\text{Sigmoid} = \frac{1}{1 + e^{-y}} \Rightarrow \frac{1}{1 + e^{-(\sum x_i w_i + b)}}$$

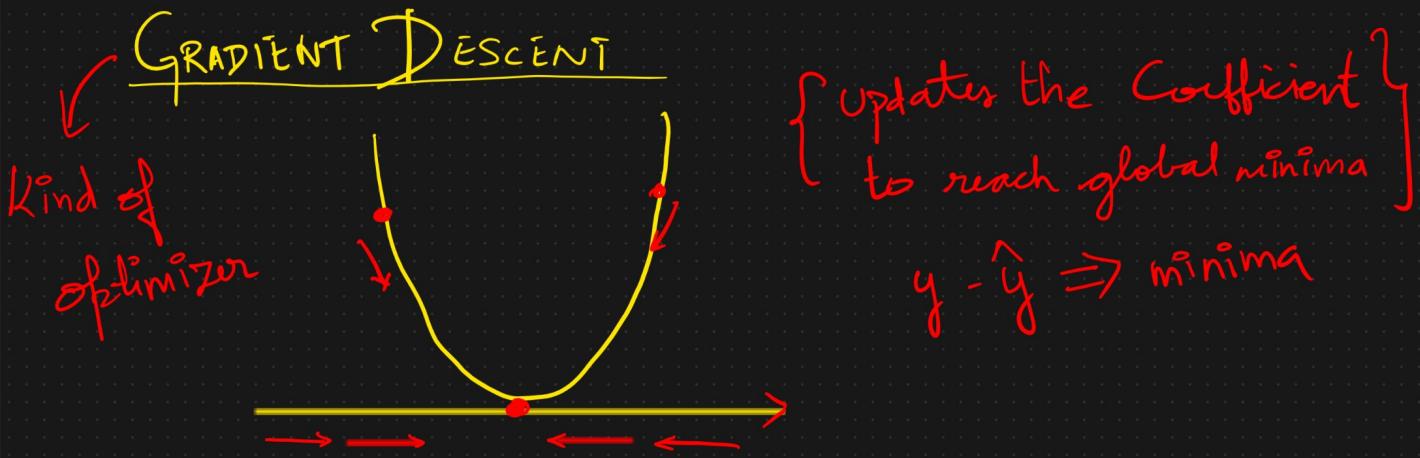
(Binary Classification)



## In Short (Till now)

- Weights {how much a Neuron Should Get Activated}
- Activation function {decides which Neuron to activate}
- Neuron {Mathematical Function that collects and classifies Information}

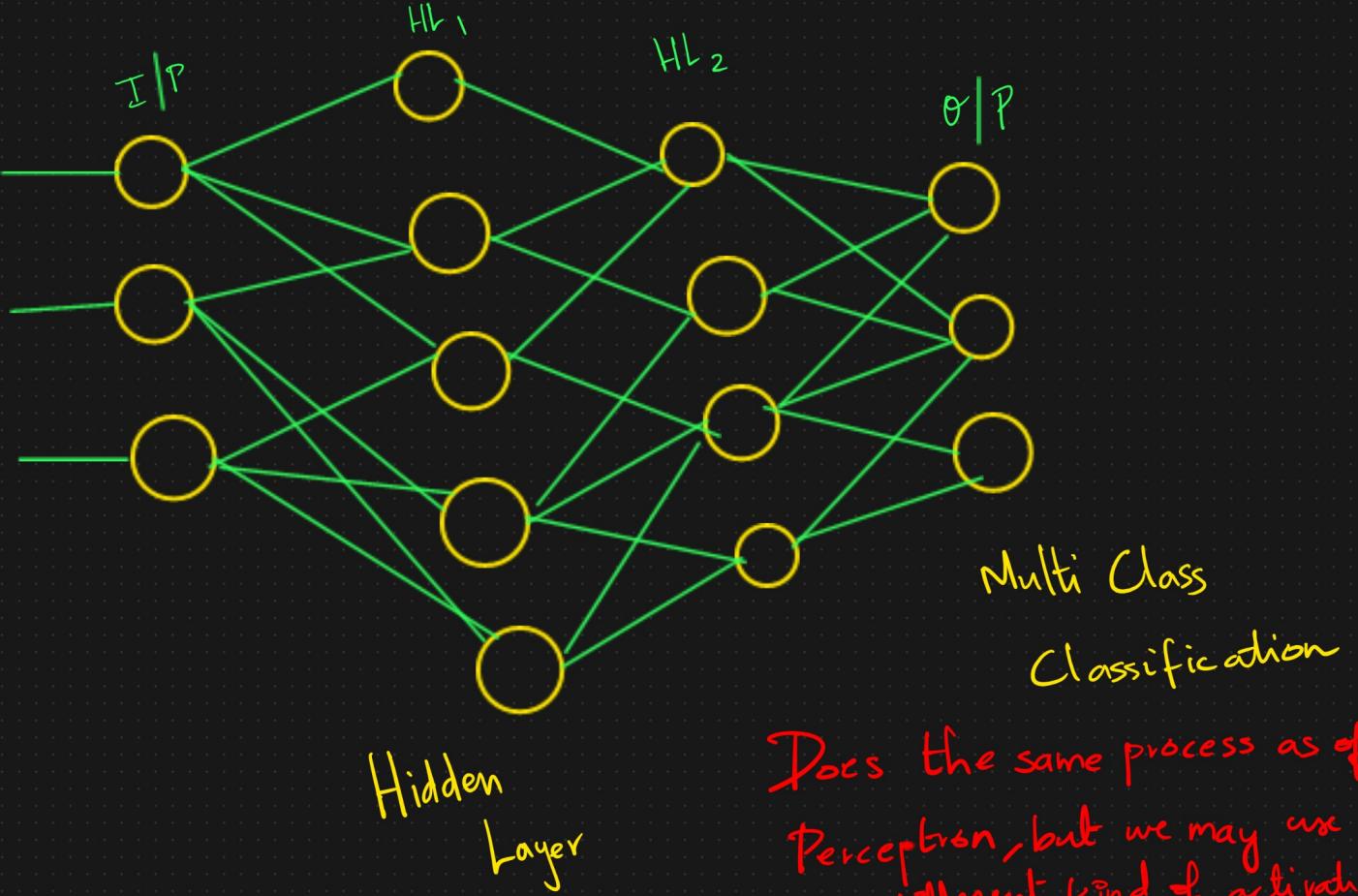
# Optimizers in Back Propagation



## Conclusion

- ① I/P Layers
  - ② Weights Get Added
  - ③ I/P \* Weights + bias
  - ④ Activation Function
  - ⑤ Loss Function  $\{y - \hat{y}\}$  to minimize
  - ⑥ Optimizer
  - ⑦ Update the weights
- Forward Propagation
- Backward Propagation

## Multi-Layered NN



Does the same process as of  
Perception, but we may use  
a different kind of activation  
function, Loss function