Tutorial- 0

Deep Learning session :

1. ANN
2. CNN
3. RNN

i. Forward Propagration
ii. Backward Propagration
iii. Optimizers
iv. Loss function
v. Activation function

Tensorflow 2.0
{5 day session}

Prequisites :

1. Static live session
2. ML live session

Tutorial- 1

Day-1 Deep Learning community Session

Day-1 : Deep Learning

Motive : { Clear their Basics, Maths, Interview Preparation }

Agenda :

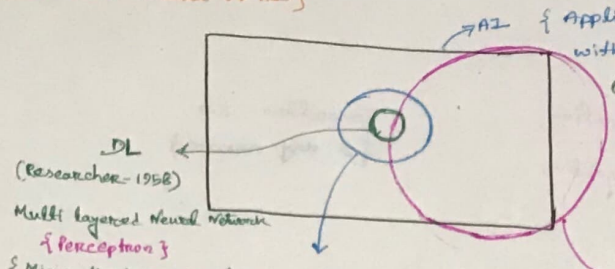
1. Deep learning → Perceptron { AI Vs ML Vs DL Vs DS }
2. Forward Propagration
3. Backward Propagration
4. Loss function
5. Activation function
6. Optimizer

Prerequits

1. Python
2. ML
3. Stats

AI vs ML vs DL vs DS → Deep learning

{ML is a subset of AI}



{Application which can do its own task without any human intervention}

① Netflix App
② Self Driving cars
③ Amazon Application
④ Sofia Robot
⑤ Youtube

ML → Statstool to analyze the data, Visualize the data, Predictions, Forecasting, clustering

→ DS

["Computer Vision" is the part of AI]

* Why Deep learning is Becoming popular ?

=> before 2005 → ORRUT, Facebook (web 2.0), Instagram, Linkedin, Twitter

Data → Exponentially (increasing ↑↑↑)

2008 → {Big data} → Efficiently

2013 → Company had huge amount of data.

Ex: Netflix

↓

{AI became popular}

Seamless Products

* Panasonic:

Ac's, Tv's, Refrigerator (Data)

↓

Model → Reduce the electricity bill (decrease ↓↓)

↓

Subscription Basis

↓

Generate revenues, Better Decission

{RTX Titan, RTX 3050}

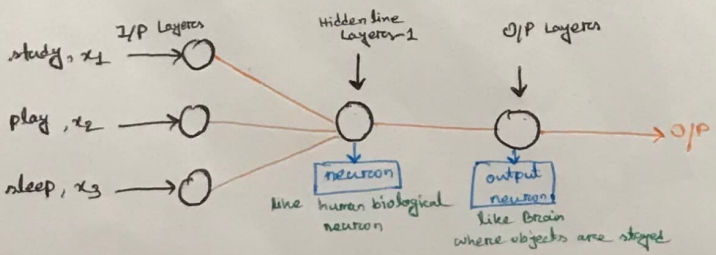
② Hardware Advancement (NVDIA) → GPU's → Training the model

GPU's → cost ↓↓↓

Perceptron (DL model):

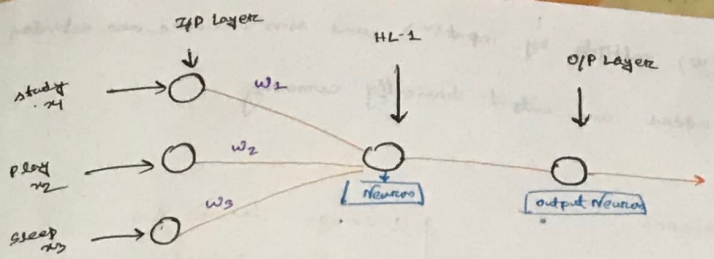
{Single Layered Neural Network}

Binary classification



Dataset (student pass or not):

	Study	play	sleep	pass/fail
→	7	3	7	1
→	2	5	8	0
→	4	3	7	0



w = weights

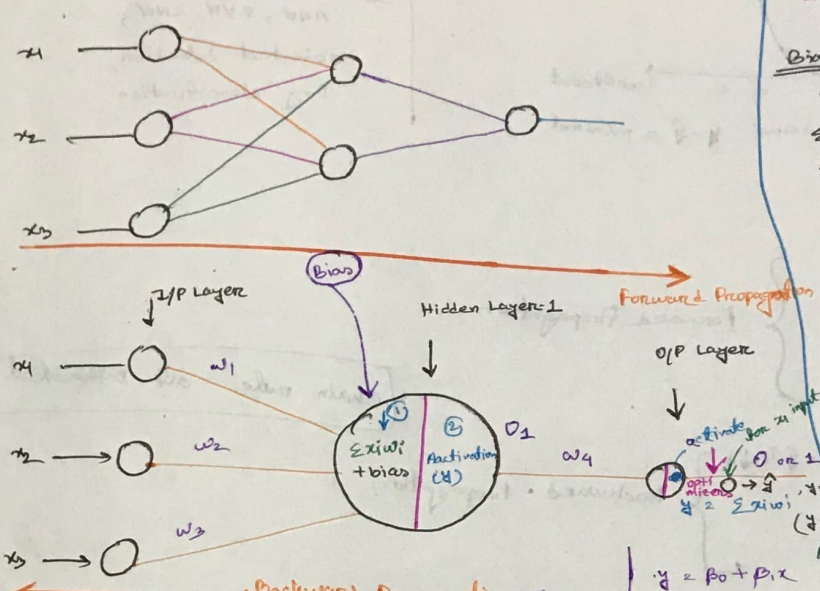
When one hot obj keeps my left hand it will automatically move. It means activated in specific reason. This signal receives brain through Neuron. But Right hand is not activated. Weight says that how much neuron should activated or deactivated.

Bias: Suppose weight is zero. When pass Input (x_1, x_2, x_3) the $\sum x_i w_i$ is to be zero. For that overcome weight should not be zero, we add another parameter is called Bias.

$y = mx + c$. here c performed as a bias. when input x is zero mx remains zero. But y doesn't zero because bias "c" adds. So that we can particular our training better we upgrade the weights

$(y - \hat{y})^2 = 1$ $(y - \hat{y})$ nearest to zero

Loss Function



$\sum x_i w_i = x_1 w_1 + x_2 w_2 + x_3 w_3$

$\therefore y = \beta_0 + \beta_1 x$

$= \beta^T x$

$y = mx + c$ (constant)

Binary classification $= w^T x$

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

0 to 1

$\geq 0.5 \rightarrow 1$

$< 0.5 \rightarrow 0$

0 to 1

Activation function

① Sigmoid

② Linear

Forward Propagation:

Loss function →

Loss function is basically find out the difference between predicted value ($\hat{y}$) and real value (y). Always aim should be that minimize the difference. Different should be nearest to zero (0). Then y and $\hat{y}$ be matching. It is called loss function.

Backward Propagation:

When difference is huge we then do backward Propagation. The main aim of back propagation is to update weight to match our real output with predicted output ($\hat{y}$).

In the ex - my predicted output $\hat{y} = 0$

& real output $y = 1$

To reduce the diff should update weights (w). This updation process is called Backward prop

What Topic Learn:

① Weights

② Neuron

③ Hidden Layers

④ Bias

⑤ Activation function

How Backward propagation happening:

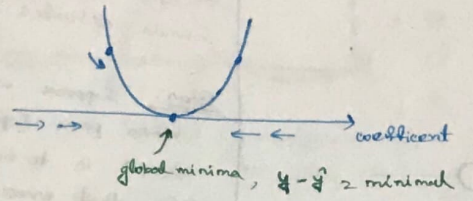
This updation process is happening because of Optimizers. It make sure that all each and every weights will update when backword propagating.

Simple Linear Regression: In simple linear regression have you hard of gradient diescent. Optimizer is gradient descent. Gradient Desent changing the co-efficients. It is combination of Forward and Backward Propagation

In Forward Propagation weights (w) multiply by input(x) and bias is added and all will added and output basically coming.

$\hat{y} = mx + c$ (weights, input, bias)

Gradient Descent :⇒ kind of optimizers



① Image classification
② Object Detection

FW & BW propagation :
ANN, RNN, CNN,
Objectest detection,
Img classification

Conclusion :

① I/P Layers
② Weights
③ Bias
④ Activation function
} Forward Propagation

⑤ Loss function $\{y - \hat{y}\}$ ↓↓
⑥ Optimizers
⑦ Update to weights
} Backward Propagation

Chain rule of Differential

Multi Layer Neural Network: