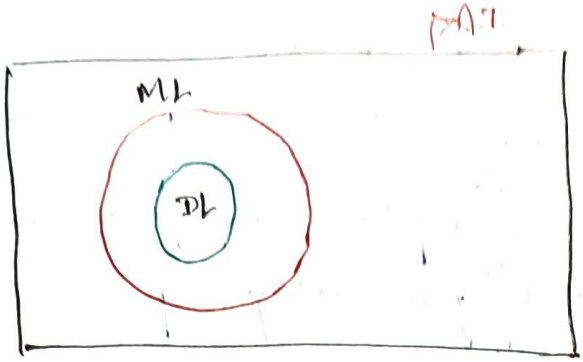
14-Jan-2023

Deep Learning



- Can we make machine learn, similarly like how we human being learns?

> Yes, scientist found a technique called deep learning which mimics the human brain with the help of multilayer ~~neural~~ neural network.

Three Neural Network in Deep learning:

① ANN → Artificial Neural Network → classification, Regression
- Image classification also can be solved
- used with tabular network.

② CNN → Convolutional Neural Network
- I/P will be Images, Videos frames
Eg: RCNN, masked, RCNN, Detection, Yolo, Vh16
- Object detection
- Computer Vision

③ RNN → Recurrent Neural Network
- I/P will be in the form of time series or text data
Eg: LSTM RNN, RNN GRU, Bidirectional LSTM RNN, Encoder Decoder, Transformers, BERT, Attention Model

Two libraries:

① Pytorch

② Tensorflow (Keras is a part of Tensorflow)

Q. Why DL becoming very popular?

Ans In 2005, Facebook, Orkut (Social Media Platform)

People started uploading photos, text etc.

Data started generating exponentially

later on many platforms came

Problem: How so they can store these data efficiently?

In 2011 → Big Data was becoming very famous.

Platform like Hadoop, Hive etc able to understand unstructured data.

In 2011–2016 → Cloudera, Hortonworks → started storing data

S3 Bucket → AWS → store data in any format

In 2015 → Problem: what will they do with the stored data?

Can we use the data to make the product better?

→ Lot of research were happening

Deep learning comes into picture.

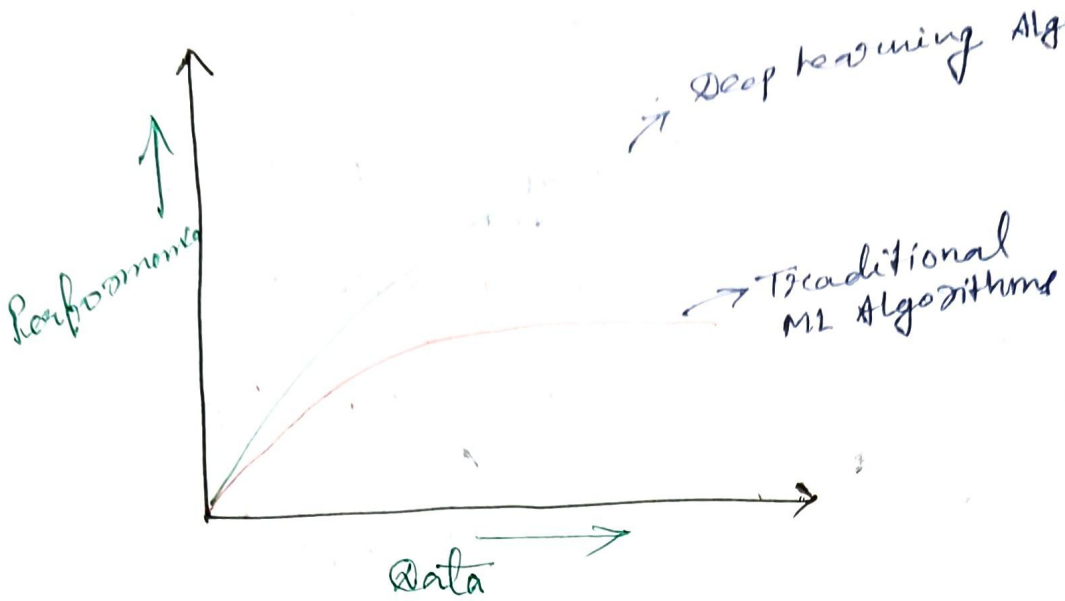
→ For training huge data → huge no. of GPUs required.

Hardware Requirement: GPUs

eg: NVIDIA GPUs

→ GPUs price started decreasing.

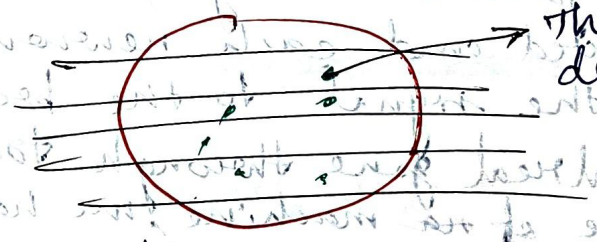
> Huge amount of data is getting generated and it is suitable for deep learning model.



> Traditional ML algorithms get stagnate after sometime, but this is not the case in DL.

> DL is been used in many domains.

① Medical: Prediction of disease, X-rays, bone crack, brain scan - 3D image. (Detect tumor or not?)



- Good amount of data should be present.

② Ecommerce

③ Retail

④ Logistic

> complex problem can be solved using DL.

① Perceptron [Artificial Neuron or Neural Network unit]

↓

Single layer Neural network

→ only used for binary classification.

1. I/P layer
2. Hidden layer
3. Weights
4. Activation function
5. Bias
6. O/p layer.

Binary classification

↓

Linear seperable

> Perceptron plays an important role.

> Neuron sends and process signal and send to the brain.

> Eyes are I/P layer for neuron.

> When signal passes to eyes, lot of neurons are interconnected and each neuron process and transmit the signal to the brain.

> our body has already gone thorough training since born but in case of machine, we have to train.

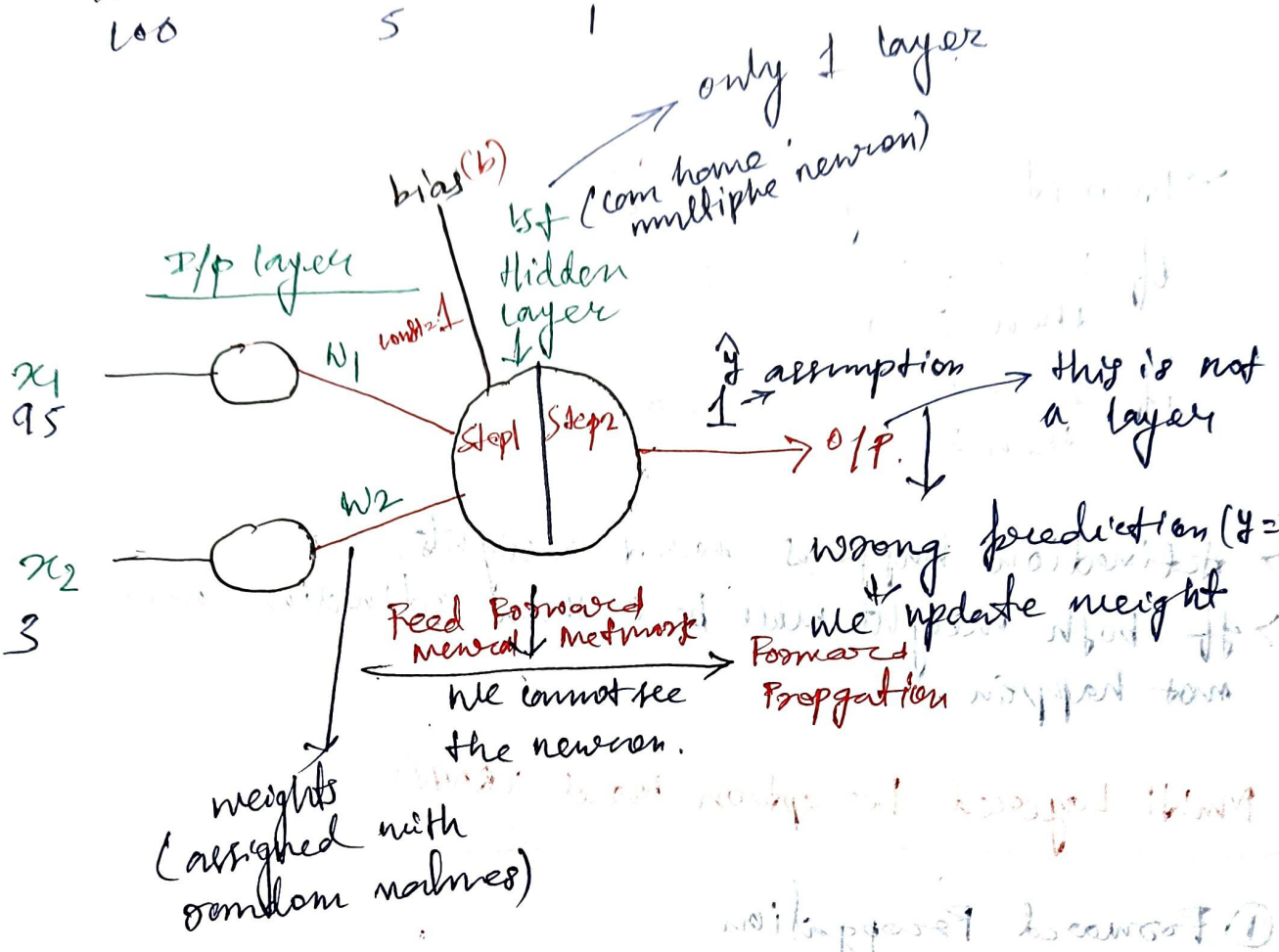
> Hidden layer are interconnected neuron.

> output layer will be our reaction.

Eg

Dataset

x_1 IQ	x_2 No. of study hours	y O/P Pass/Fail
95	3	0
110	4	1
100	5	1



~~In Step1 $x_1 * w_1 + x_2 * w_2 + b$ (constant = 1)~~

Step1

$$z = x_1 * w_1 + x_2 * w_2 + 1 * b$$ → intercept

$$z = \sum_{i=1}^{n} x_i w_i + b$$

linear Regression Equation.

> this creates a linear line so we can solve linear seperable problem.

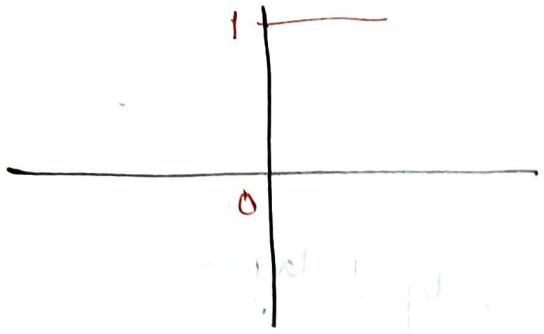
> 'b' will be intercept

Step2

> we apply activation function. called step function.

Step function

> Threshold = 0.



> Transforming Z to [0.00 1].

if Z = 5, ≥ Z > 1
then Z = 1

if Z = -2, Z < 0
then Z >= 0

> Activation happens w.r.t weights.
> If both weight value be 0 then activation value will not happen..

② Multi Layered Perceptron Model (ANN)

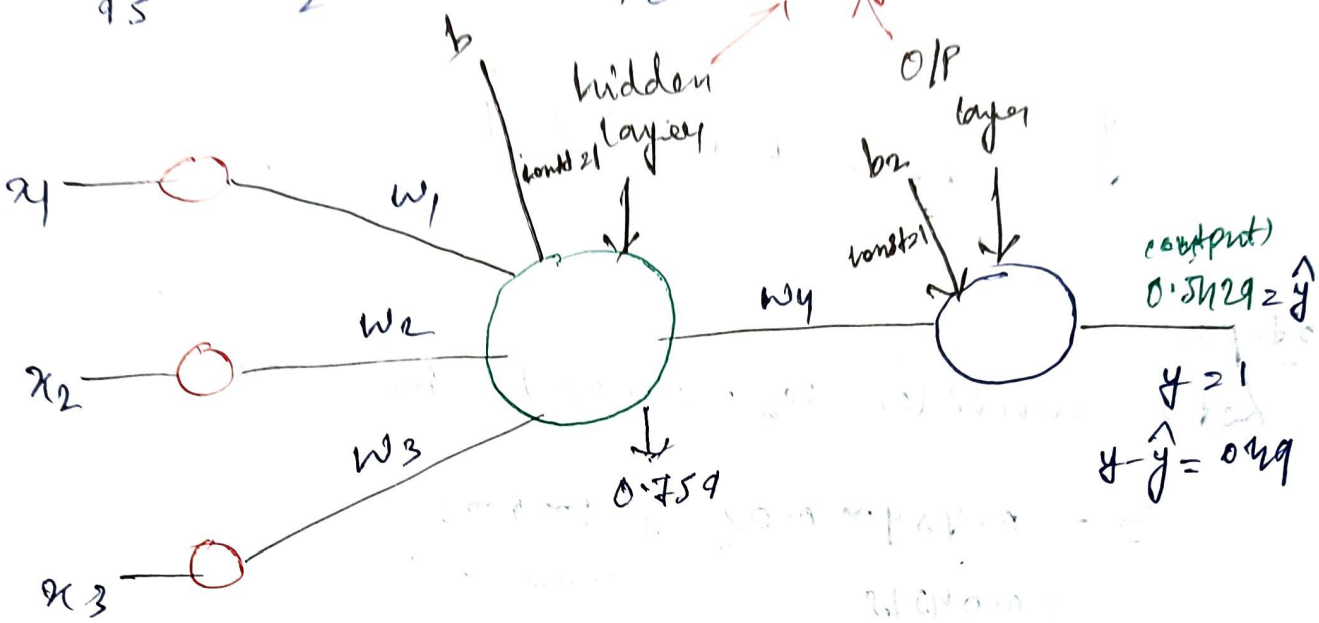
① Forward Propagation
② Backward Propagation
③ Loss Function
④ Activation Function
⑤ Optimizers

2 layer neural network

Dataset

same in 1 layer neural network

(x_1) ID	(x_2) study hrs	(x_3) Play hrs	O/P (Pass / Fail)
95	4	4	1
100	5	2	1
95	2	7	0



Let's consider $[w_1 \; w_2 \; w_3]$ $b = 0.01$

$[0.01 \; 0.02 \; 0.03]$

Forward Propgation

Step1

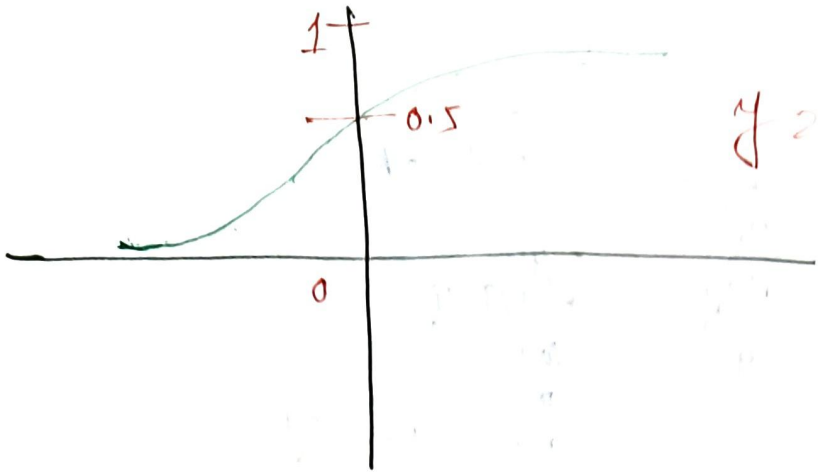
$$z = 95 * 0.01 + 4 * 0.02 + 4 * 0.03 + 0.01 * 1$$

$$= 1.151$$

Step2

Apply activation function.

Sigmoid Activation function – Transformed z into 0. to 1.



$$y = \frac{1}{1+e^{-1.151}} = 0.759$$

Step3

Let's consider $w_1 = [0.02]$ $b_2 = 0.03$

$$z = 0.759 * 0.02 + 1 * 0.03$$

(1 * constant)

$$= 0.04518$$

Step4

Apply Activation function

$$y = \frac{1}{1+e^{-z}}$$

$$= \frac{1}{1+e^{-0.04518}}$$

$$= 0.5429$$

Step5

$\hat{y} = 0.5429$

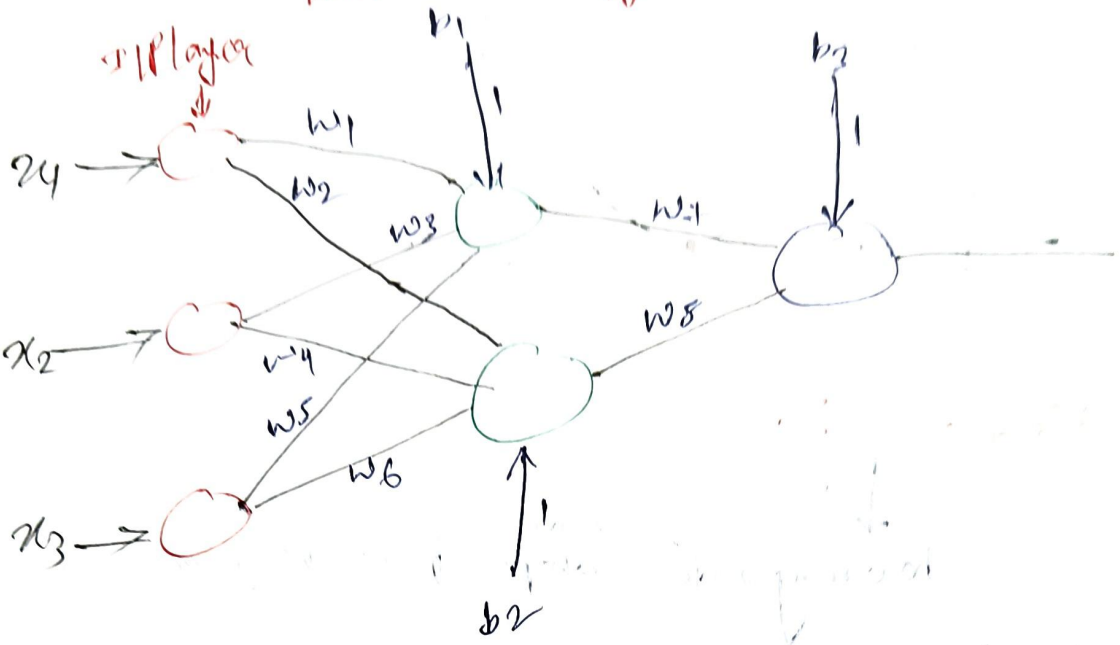
$y = 1$

Loss $= y - \hat{y} = 1 - 0.5429 = 0.49$ (It is on higher side)

> Since loss is high, we will perform backward propgation by changing all the weights

Backward Propgation and updation Formula

Forward Propgation



← Backward Propgation

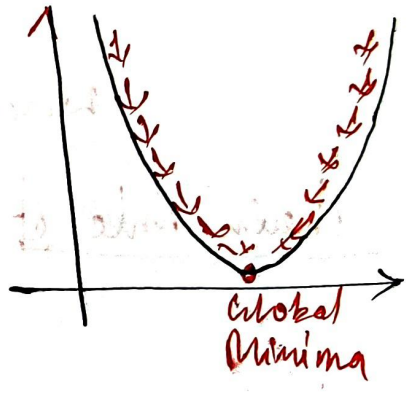
Loss function = $(y - \hat{y})^2$

weight updation Formula

$$w_{new} = w_{old} - \eta \left[\frac{\partial L}{\partial w_{old}}\right]$$

$$W_{new} = W_{old} - \eta \left(\frac{\partial loss}{\partial W_{old}}\right)$$

learning rate → derivative

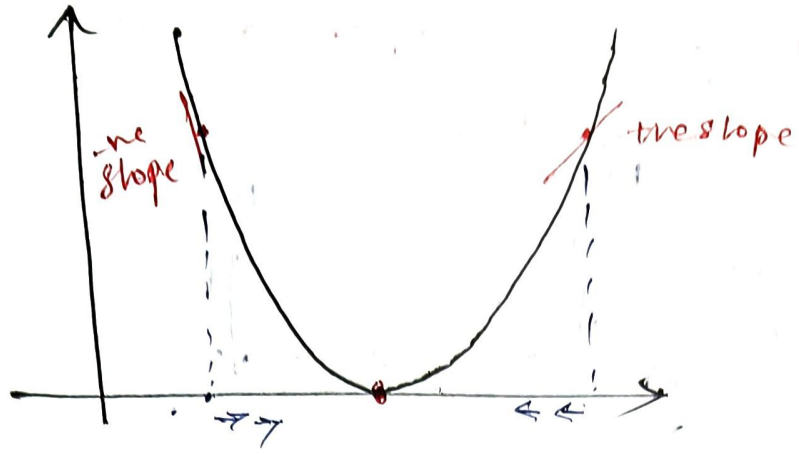


$$b_{new} = b_{old} - \eta \left(\frac{\partial loss}{\partial b_{old}}\right)$$ Bias updation formula.

> Weight updation can be done with the help of optimizers like Gradient Descent.

Optimizers

> To reduce the loss value.



-ve slope

$$W_{new} = W_{old} - \eta \frac{\partial loss}{\partial W_{old}}$$

learning rate → Rate of convergence.

~~-ve slope~~

$W_{new} > W_{old}$

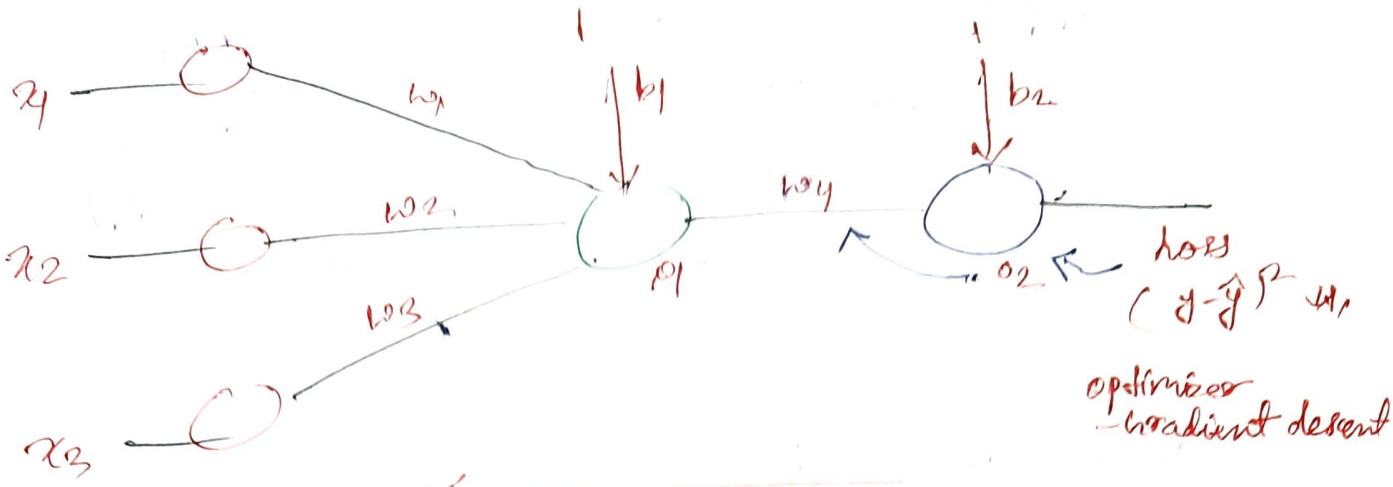
+ve slope

$$W_{new} = W_{old} - \eta \frac{\partial loss}{\partial W_{old}}$$

$W_{new} << W_{old}$

~~Chain Rule of Derivative~~

chain Rule of Derivative



Backward Propgation

$$w_{4\,new} = w_{old} - \gamma \left[\frac{\partial loss}{\partial w_{old}} \right] \rightarrow \text{slope}$$

> loss is dependent on o_1, o_2 is dependent on w_4 now.

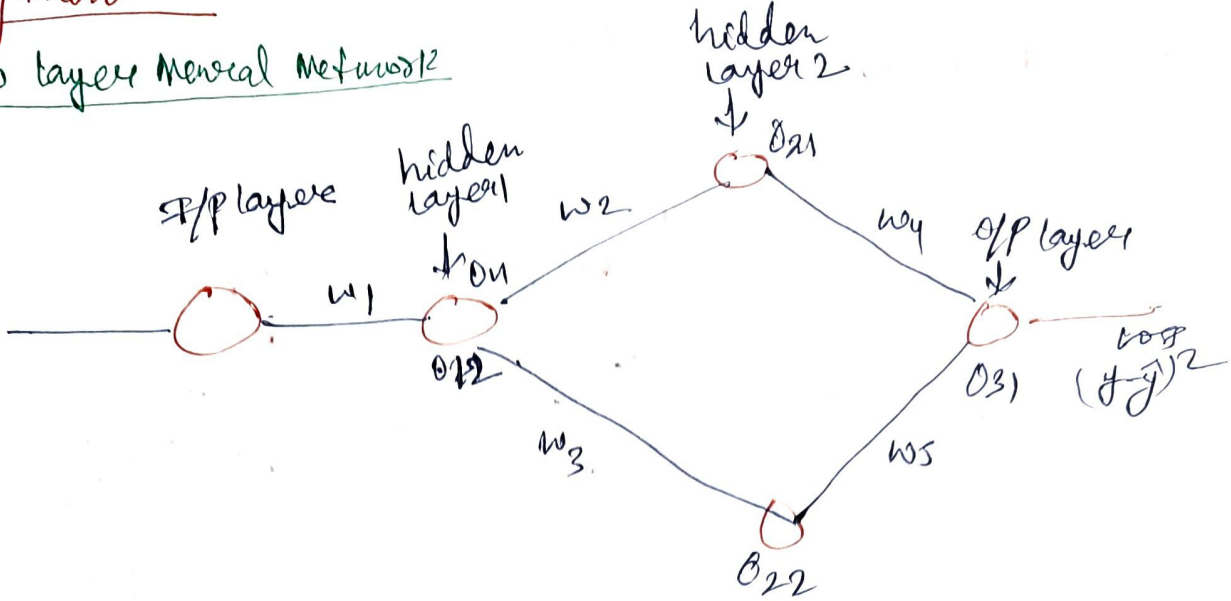
$$\boxed{\frac{\partial loss}{\partial w_{old}} = \frac{\partial loss}{\partial o_2} * \frac{\partial o_2}{\partial w_{4\,old}}}$$ > chain Rule of Derivative

$$w_{new} = w_{old} - \gamma \frac{\partial loss}{\partial w_{old}}$$

$$\boxed{w_{new} = w_{old} - \gamma \left(\frac{\partial loss}{\partial o_2} * \frac{\partial o_2}{\partial [redacted]} * \frac{\partial o_1}{\partial w_{old}} \right.}$$

Assignment

Two layer Neural Network



$$W_{new} = W_{old} - \alpha \left(\frac{\partial loss}{\partial W_{old}}\right)$$

> there will be two path

1st Path

$$\left[\frac{\partial loss}{\partial w_{old}}\right]_1 = \left[\frac{\partial loss}{\partial o_{31}} * \frac{\partial o_{31}}{\partial o_{21}} * \frac{\partial o_{21}}{\partial o_{11}} * \frac{\partial o_{11}}{\partial w_1}\right]$$

2nd Path

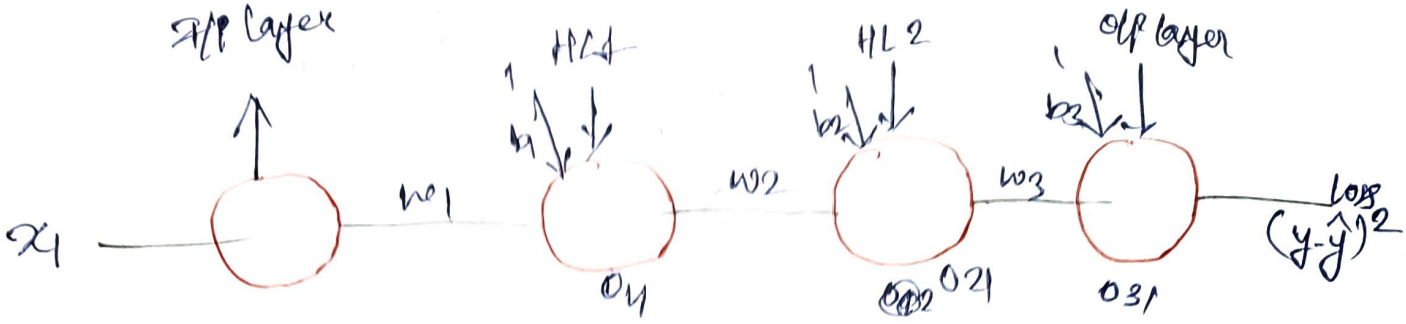
$$\left[\frac{\partial loss}{\partial w_{old}}\right]_2 = \left[\frac{\partial loss}{\partial o_{31}} * \frac{\partial o_{31}}{\partial o_{22}} * \frac{\partial o_{22}}{\partial o_{12}} * \frac{\partial o_{12}}{\partial w_1}\right]$$

$$\left[\frac{\partial loss}{\partial w_{old}}\right] = \left[\frac{\partial loss}{\partial w_{old}}\right]_1 + \left[\frac{\partial loss}{\partial w_{old}}\right]_2$$

> w_1 is dependent on both the path. So they have been added.

Vanishing Gradient Problem and Activation functions

Three layer neural network



O_{ij} → Outputs

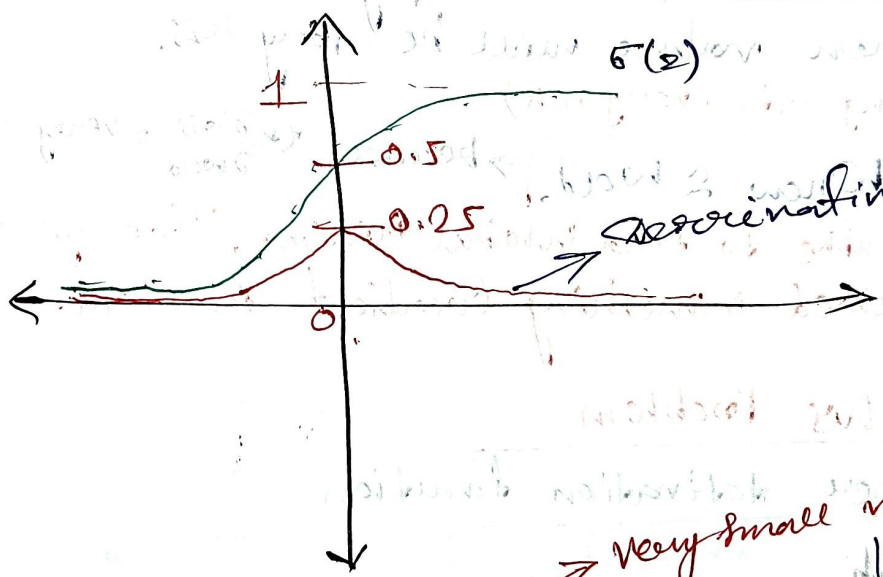
Sigmoid Activation

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad [0 \text{ to } 1]$$

$0 \le \sigma(z) \le 1$

Derivation of $\sigma(z)$

$0 \le \sigma(z) \le 0.25$



$$W_{new} = W_{old} - \eta \frac{\partial loss}{\partial w_{old}}$$

very small value

(η * ∂loss/∂wold = more small)

small value

$$\frac{\partial loss}{\partial w_{old}} = \frac{\partial loss}{\partial O_{31}} * \frac{\partial O_{31}}{\partial O_{21}} * \frac{\partial O_{21}}{\partial O_{11}} * \frac{\partial O_{11}}{\partial w_1}$$

$$O_{31} = \sigma(w_3 * O_{21} + b_3) \qquad z = w_3 * O_{21} + b_3$$

small value

$$\frac{\partial O_{31}}{\partial O_{21}} = \frac{\partial(\sigma(z))}{\partial(z)} * \frac{\partial(z)}{\partial O_{21}} \quad \{\text{chain rule}\}$$

$\{0-0.25\}$

$$\frac{\partial(w_3 * O_{21} + b_3)}{\partial O_{21}} = w_3$$

$$\rightarrow \frac{\partial w_3 O_{21}}{\partial O_{21}} + \frac{\partial b_3}{\partial O_{21}}$$

$$\rightarrow w_3 + 0$$

$$= w_3$$

$$\frac{\partial O_{31}}{\partial O_{21}} \rightarrow [0-0.25] * w_3$$

> As the chain rule is very big, derivative value becomes zero. If we multiply all the derivative value then value will be very less.

η = learning rate (very less)

$w_{new} \simeq w_{old}$ → because $\eta * \frac{\partial loss}{\partial w_{old}}$ = very small

> Not able to find global minima, This problem is called Vanishing Gradient Problem.

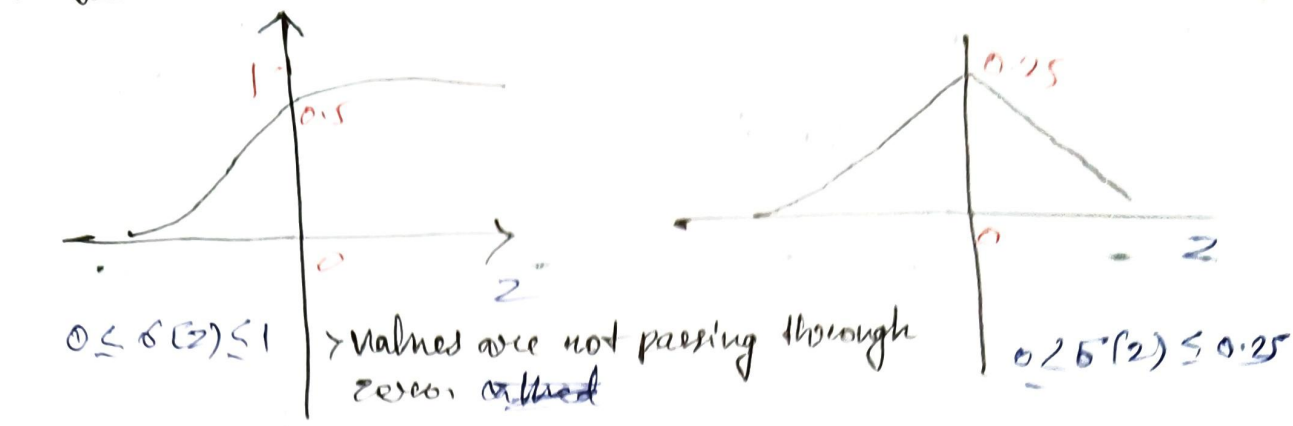
To fix this Problem

> Use other Activation function

1. Tanh
2. Relu
3. Prelu
4. Elu
5. Swish

Activation Function

① Sigmoid Activation Function



$0 \leq \sigma(z) \leq 1$ > values are not passing through zero.

$0 \leq \sigma'(z) \leq 0.25$

Advantages

① It gives clear prediction (0 to 1).

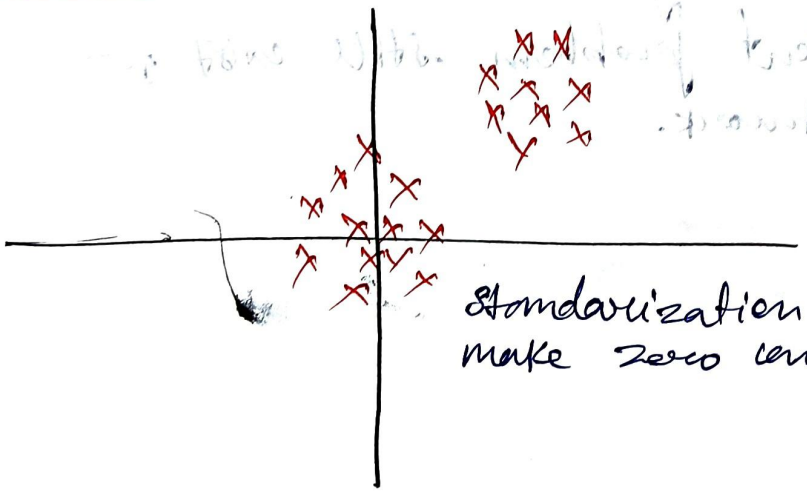
Disadvantages

① It is prone to vanishing gradient problem.

② Function output is not zero centered.

Efficient weight updates.

Zero centred



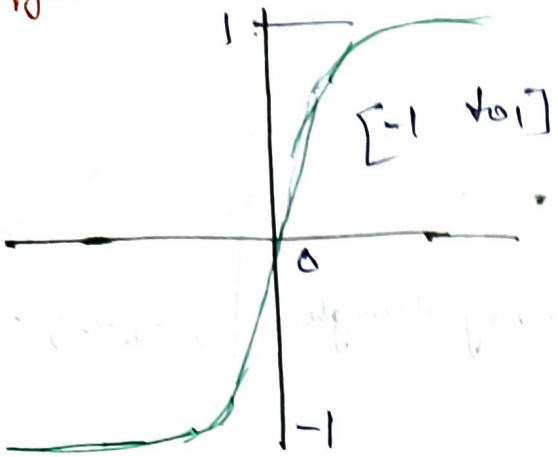
Standardization to make zero centred.

> Zero centred smoothen the activation function

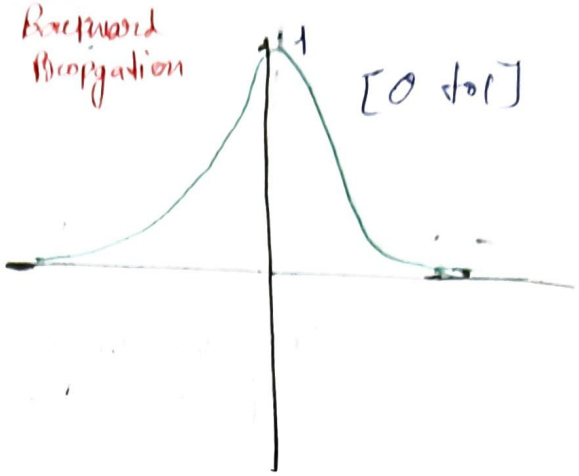
③ Tanh Activation function

$$\tanh z = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \quad (-1 \text{ to } 1)$$

Forward Propagation



Backward Propagation



Advantages

① It is zero centred.

Disadvantages

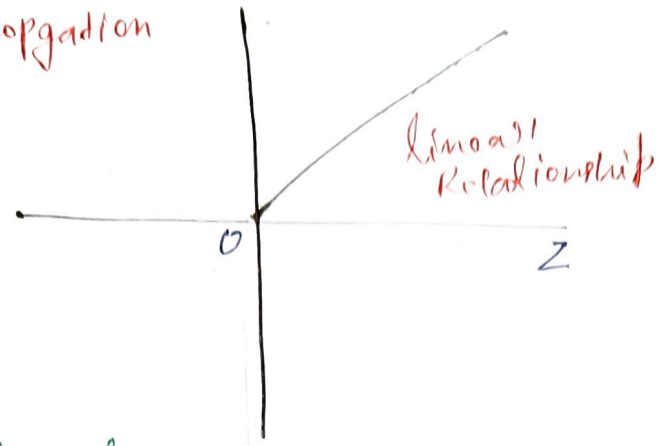
② More time complexity due to derivative.

③ Vanishing gradient problem still exist for deep neural network.

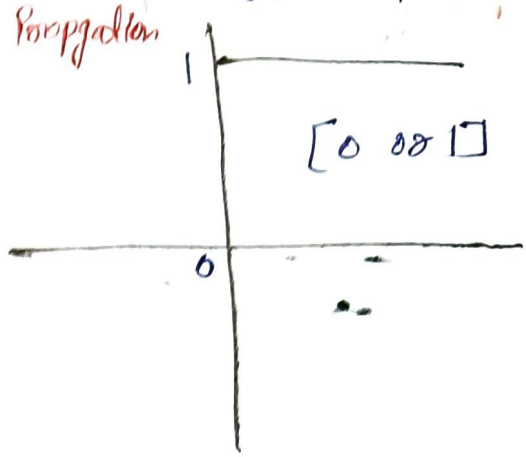
③ Relu Activation Function

$R(z) = \max(0, z)$

Forward Propagation



Backward Propagation $d\text{Relu}(x)/dx$



Advantages

① solved vanishing gradient problem.
② It is faster than tanh and sigmoid.

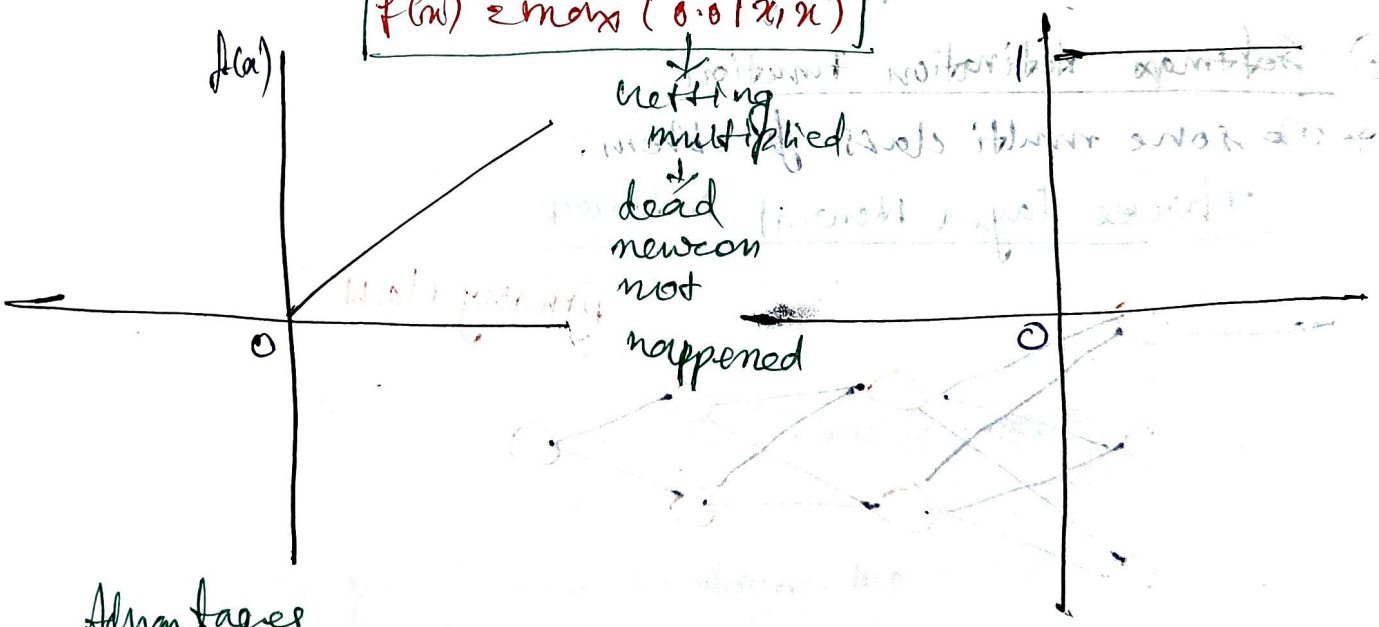
Disadvantages

① If derivative is zero, then it become dead neuron which will skip the layer.
② It is not zero centric.

④ Leaky Relu or Parametric Relu

$f(x) = \max(0.01x, x)$

$d f(x)/dx$



Advantages

① Dead neuron problem is solved.

Disadvantages

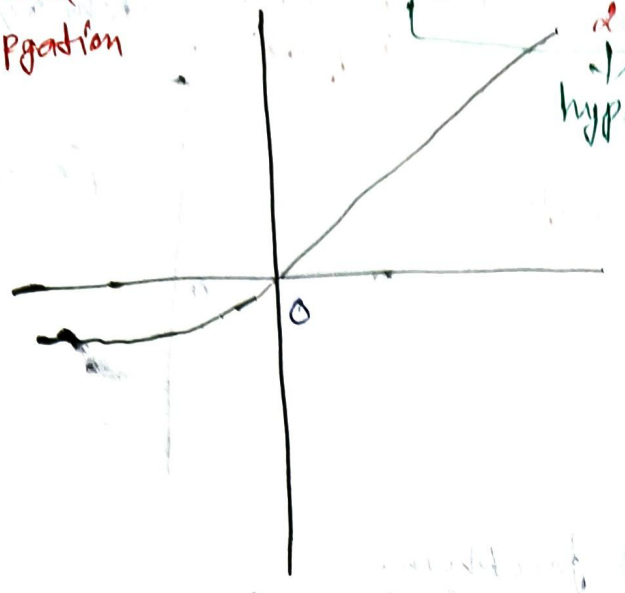
① It is not zero centric

⑤ ELU (Exponential linear Unit)

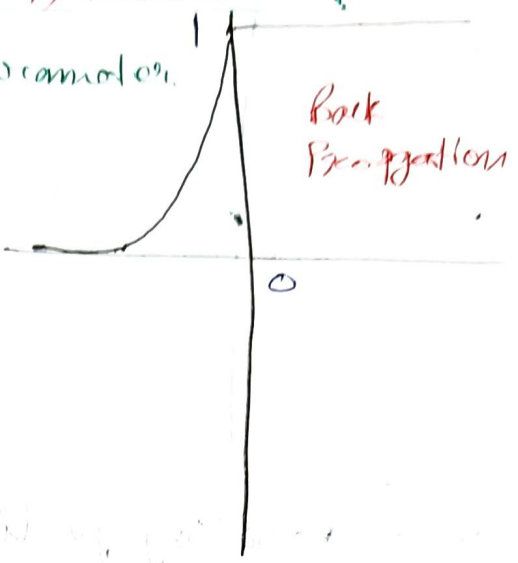
Forward & Bpgation

$f(x) = x$, if $x > 0$
$\alpha(e^x - 1)$, otherwise

↓
hyperparameter



Back Propagation



Advantages

① It is zero centered
② Vanishing gradient problem get solved.

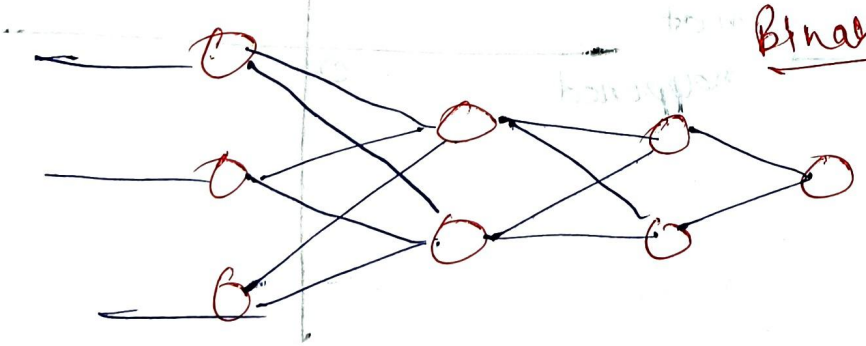
Disadvantages

① More time complexity due to complex derivative.

⑥ Softmax Activation Function

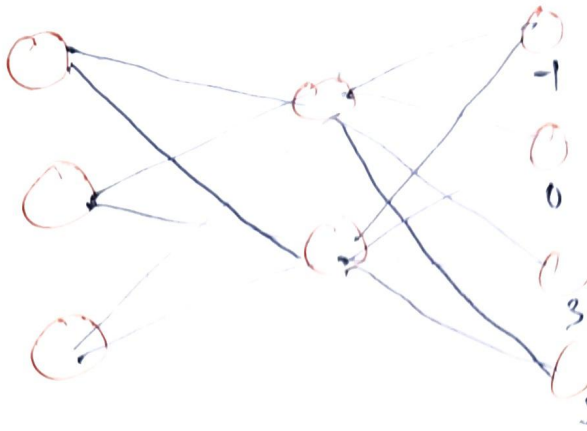
→ To solve multi class problem.

Three layer Neural Network



Binary class

Multi class Problem



Horse → 0.86 = 86% probability.

$$\text{Softmax} = \frac{e^{y_i}}{\sum_{k=0}^{n} e^{y_k}}$$

$$\text{cat} = \frac{e^{-1}}{e^{-1}+e^{0}+e^{3}+e^{5}}$$

output after applying softmax function.

$$\text{cat} = \frac{e^{-1}}{e^{-1+0+3+5}} = 0.00033$$

$$\text{Dog} = \frac{e^{0}}{e^{-1+0+3+5}} = 0.0024$$

$$\text{Monkey} = \frac{e^{3}}{e^{-1+0+3+5}} = 0.0183$$

$$\text{Horse} = \frac{e^{5}}{e^{-1+0+3+5}} = 0.1353 \quad \text{(Highest)}$$

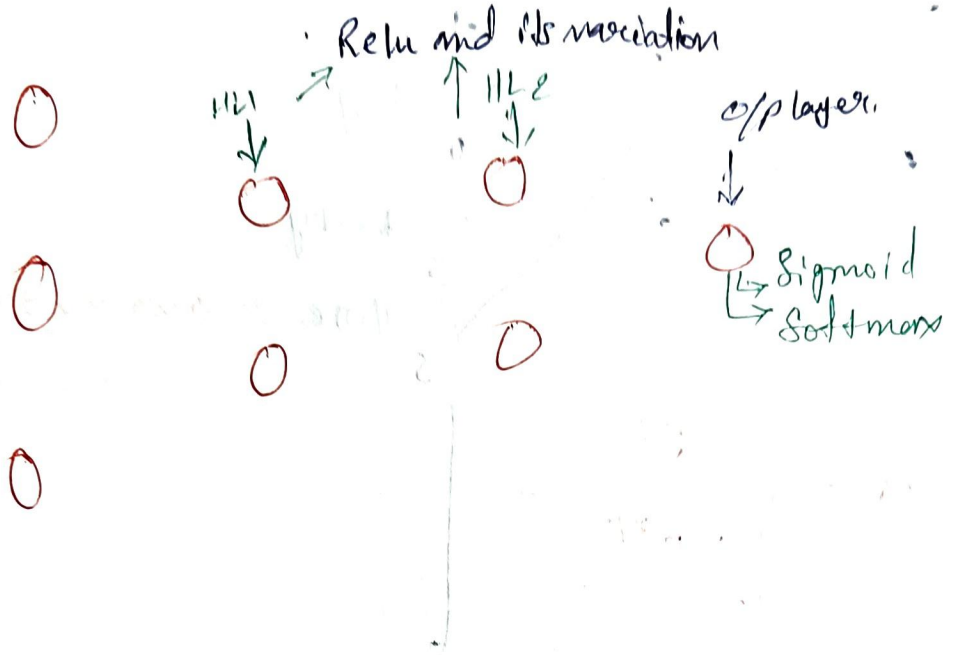
Probability

$$P(\text{Horse}) = \frac{0.1353}{0.00033+0.0024+0.0183+0.1353} = 0.86$$

> In case of multi class problem, we must use softmax activation function.

Q — Which activation function to use when?

Ans



Sigmoid → Binary classification
Softmax → multiclass classification

Regression

① Linear Activation function