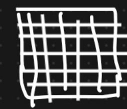
Deep Learning



{ Multi Layered Neural N/W ← DL → Lot of Research

DL ⇓ Mimicing the human brain

ML → Tabular dataset → Classification, Regression, Clustering Algo

Deep Learning

Tabular

① ANN → Artificial Neural N/W → CLASSIFICATION, Regression

② CNN → Convolutional Neural N/W → I/p = Images, Video frames $\boxed{\text{VGG 16}}$, Resnet

⇓ Computer Vision

Eg: RCNN, Masked RCNN, Detectron, Yolo V6

Yolov8

③ RNN → Recurrent Neural N/W → I/p ⇒ Text Data, Time Series Data.

↳ LSTM RNN, RNN GRU, Bidirectional LSTM RNN, Encoder Decoder, Transformers, BERT, Attention Models

Keras ⇓ ⇓ $\boxed{\text{Tensorflow, PyTorch}}$

② Why Deep Learning Is Becoming popular?

2005 → Facebook, Orkut } → Social Media platform

Images, Text, Documents Videos

⇓

DATA WAS GETTING GENERATED EXPONENTIALLY. ⇒ HADOOP ⇒ unstructured data, structured data

⇓ 2011 ⇒ BIG DATA

⇓

STORING THESE DATA EFFICIENTLY

⇓

S3 Buckets

2011 - 2011 ⇒ Cloudera, Hortonworks

⇓

2015

⇓

Can we use the data to make the product better

⇓

DATA SCIENCE

⇓

Deep Learning

NVIDIA

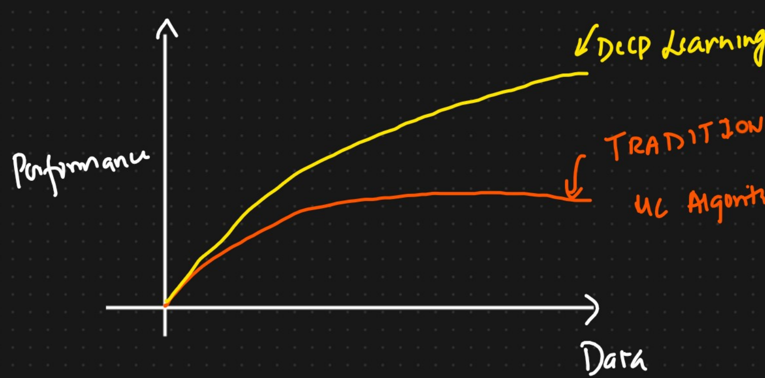
⇓

TITAN RTX

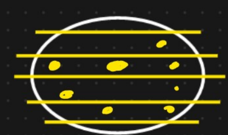
RTX 3080

① Hardware Requirement ⇒ GPU's ⇒ Nvidia GPU's Price ⇓⇓

② Huge amount of data is generated ⇒ Deep Learning Model performs well



Tumor



Brain Scan

③ Deep Learning is been used in Many Domains

3D Image

① Medical → Prediction of diseases, X rays, Bone crack, lungs disease.

⇓

DATA

MRI Scan

② Ecommerce

③ Retail

④ Logistic

① Perceptron

[Artificial Neuron or Neural Network Unit]

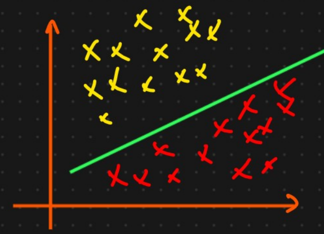
⇓

Single Layered Neural N/W

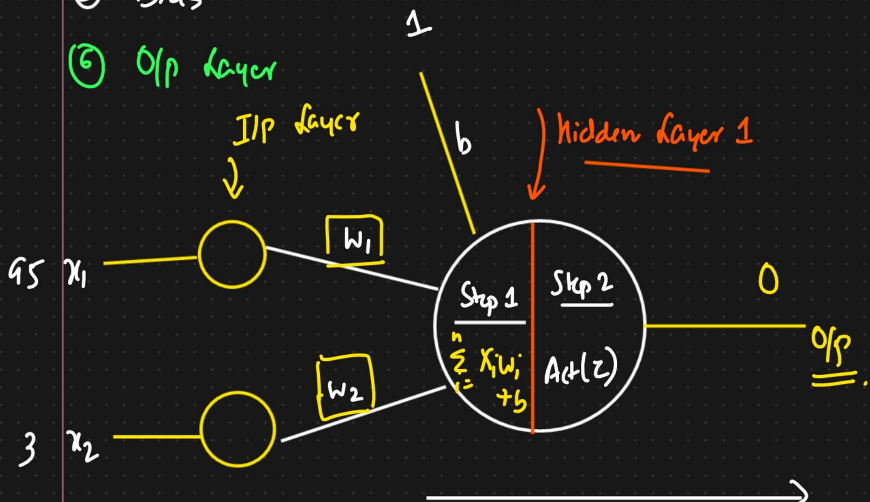
[Binary classification]

⇓

Linear Seperable



① I/p Layer
② Hidden Layer
③ Weights
④ Activation function
⑤ Bias
⑥ O/p Layer



DATASET

x_1 IQ	x_2 No. of Study hours	y O/p PASS/FAIL
95	3	0
110	4	1
100	5	1

Step 1 — Feed Forward Neural N/W ⟶ Forward Propogation

↓

$$z = x_1 w_1 + x_2 w_2 + 1 \times b$$ ← Intercept

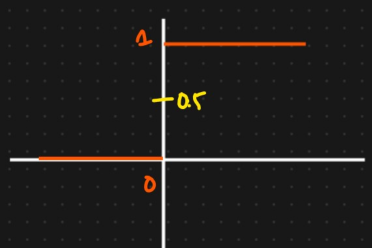
$$z = \sum_{i=1}^{n} x_i w_i + b$$

$\boxed{z = -2}$

$\boxed{z = 5} \Rightarrow 1$

$\boxed{5 > 0}$

Step Function

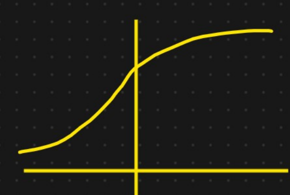


$\boxed{\text{Threshold} = 0}$

$\boxed{0.5}$

$\boxed{0 \text{ or } 1}$

Sigmoid



Hyperparameter

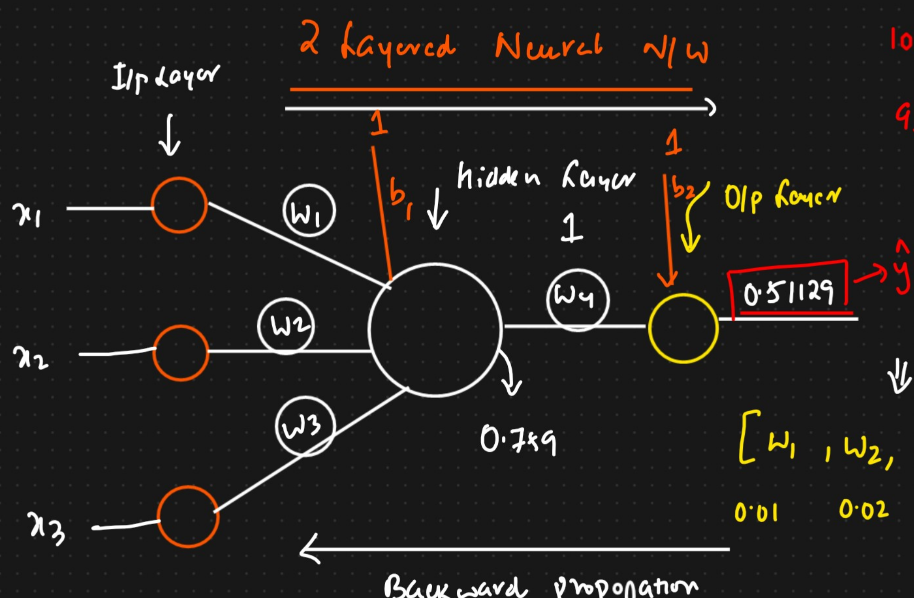
$\boxed{0.1}$ $\boxed{0.2}$ 0.3, 0.4, 0.5

Multi Layered Perceptron Model [ANN]

① Forward Propogation
② Backward Propogation
③ Loss function
④ Activation function
⑤ Optimizers.

Binary

x_1 IQ	x_2 Study hours	x_3 play hours	O/p
95	4	4	1
100	5	2	1
95	2	7	0



Loss $= 1 - 0.51129 = 0.49$ ↑↑

$(y - \hat{y})$

Initialized

$[w_1, w_2, w_3]$ $[0.01]$

0.01 0.02 0.03

HL 1

Step 1 $= 95 * 0.01 + 4 \times 0.02 + 4 * 0.03 + 0.01$

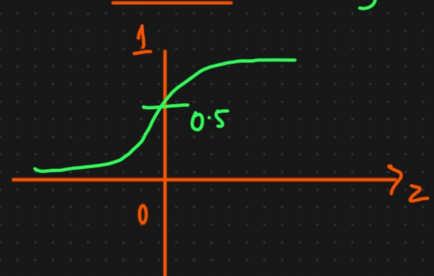
$z = 1.151$

Step 2 $=$ Activation (z)

Sigmoid

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

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



HL 2 :

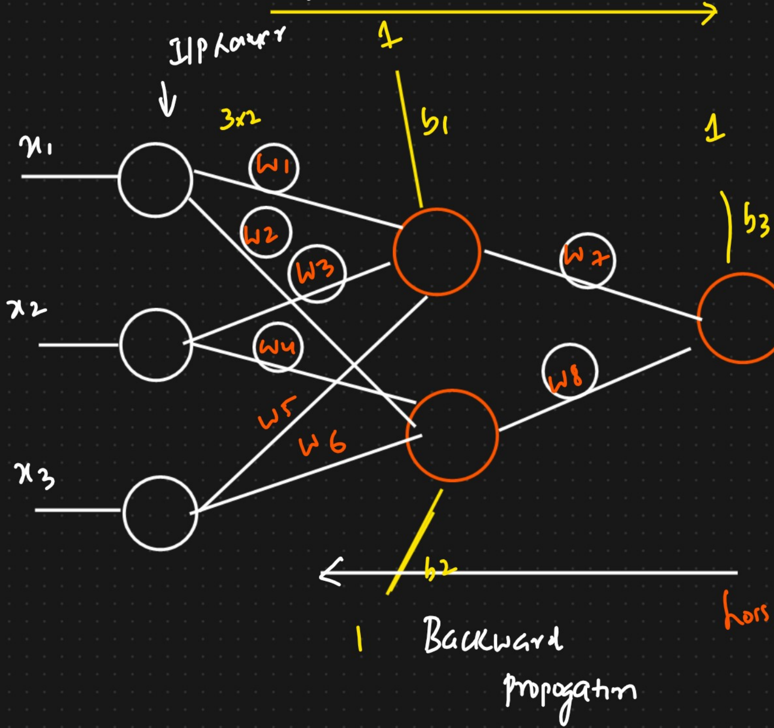
$w_4 = [0.02]$ $b_2 = 0.03$

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

$= 0.04518$

Step 2 : $O_2 = \dfrac{1}{1+e^{-0.04518}} = 0.51129 \Rightarrow$ O/P Layer

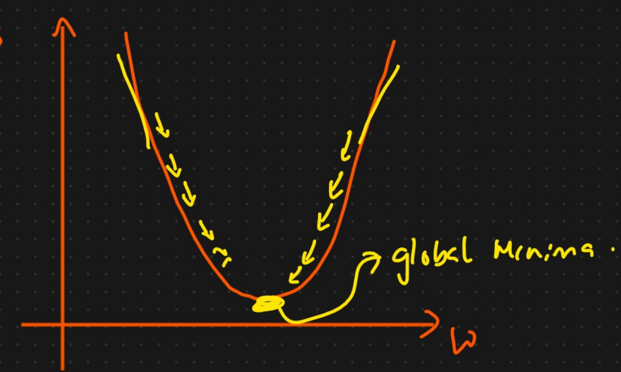
Backward Propogation And Weight Updation Formula



L Loss function

$(y-\hat{y})^2 \downarrow\downarrow\downarrow\downarrow$

Optimizers $\Rightarrow$ Gradient Descent



Weight updation Formula

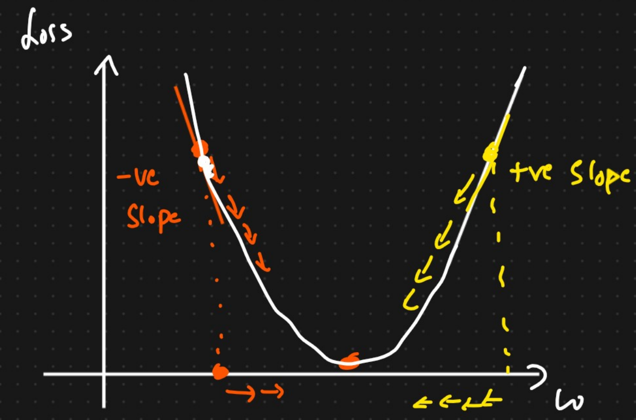
$$W_{7new} = W_{7old} - \eta \boxed{\frac{\partial L}{\partial W_{7old}}}$$

$\hookrightarrow$ slope

$$\boxed{W_{new} = W_{old} - \eta \frac{\partial L}{\partial W_{old}}} \Rightarrow \text{Weight Updation Formula.}$$

$$\boxed{b_{new} = b_{old} - \eta \frac{\partial L}{\partial b_{old}}} \Rightarrow \text{bias Updation Formula.}$$

Optimizers : To reduce the Loss value



$W_{new} = W_{old} - \eta\,(-ve)$

η = Learning Rate

$W_{new} = W_{old} + (+ve)$

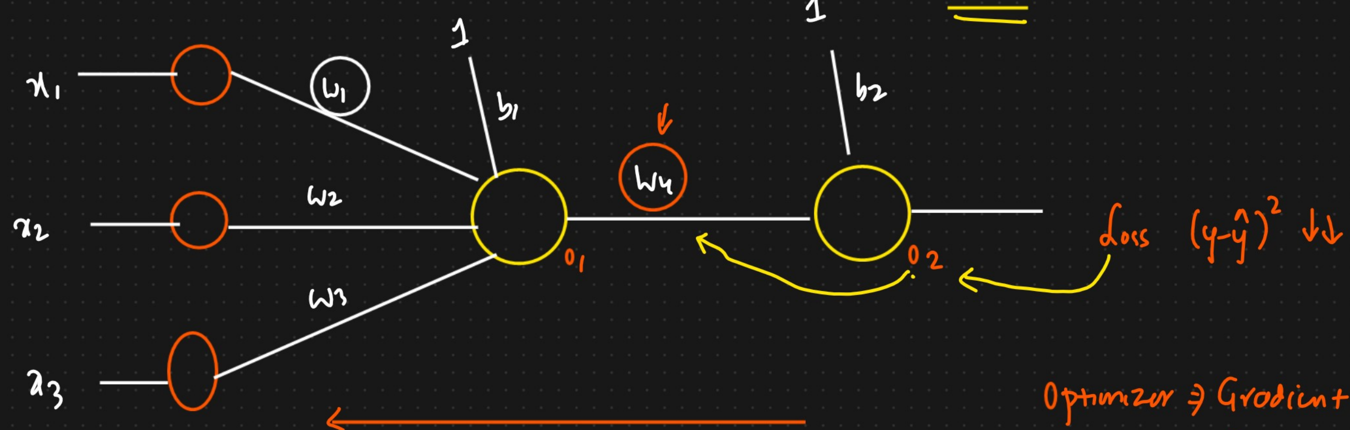
$\boxed{W_{new} >> W_{old}}$

$W_{new} = W_{old} - \eta\,(+ve)$

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

(*) Chain Rule Of Derivative

$\sigma(o_1 * w_4 + b_2)$



Optimizer ⇒ Gradient Descent

$$w_{4_{new}} = w_{4_{old}} - \eta \boxed{\frac{\partial L}{\partial w_{4_{old}}}} \Rightarrow \text{Slope}$$

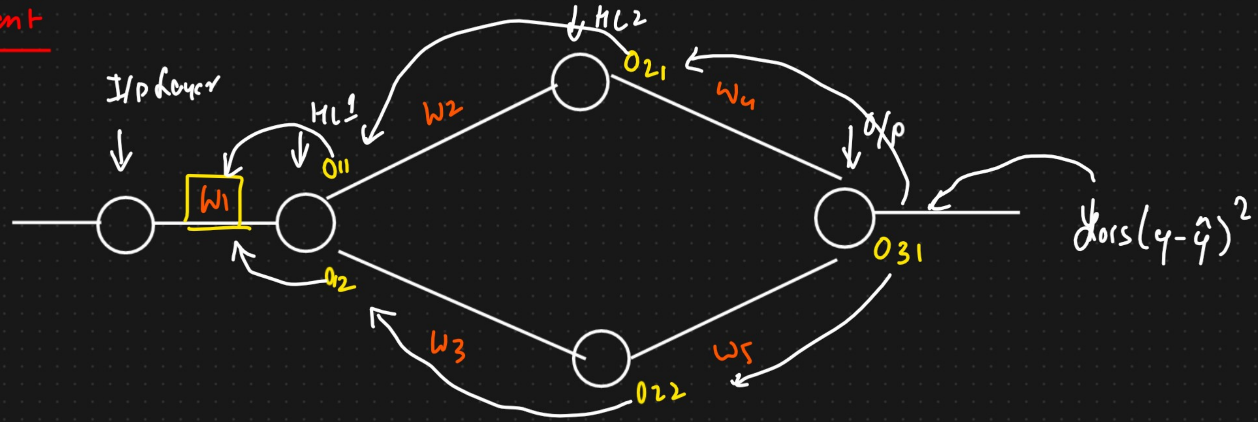
$$\frac{\partial L}{\partial w_{4old}} = \frac{\partial L}{\partial o_2} * \frac{\partial o_2}{\partial w_{4old}}$$

{Chain Rule} Of Derivation

$$W_{1new} = W_{1old} - \eta \boxed{\frac{\partial L}{\partial W_{1old}}}$$

$$\frac{\partial L}{\partial w_{1\,old}} = \frac{\partial L}{\cancel{\partial o_2}} * \frac{\cancel{\partial o_2}}{\cancel{\partial o_1}} * \frac{\cancel{\partial o_1}}{\partial w_{1\,old}}$$

Assignment



$$W_{1new} = W_{1old} - \eta \boxed{\frac{\partial L}{\partial W_{1old}}}$$

$$\frac{\partial L}{\partial W_{1old}} = \left[\frac{\partial L}{\partial O_{31}} * \frac{\partial O_{31}}{\partial O_{21}} * \frac{\partial O_{21}}{\partial O_{11}} * \frac{\partial O_{11}}{\partial W_{1old}}\right]$$

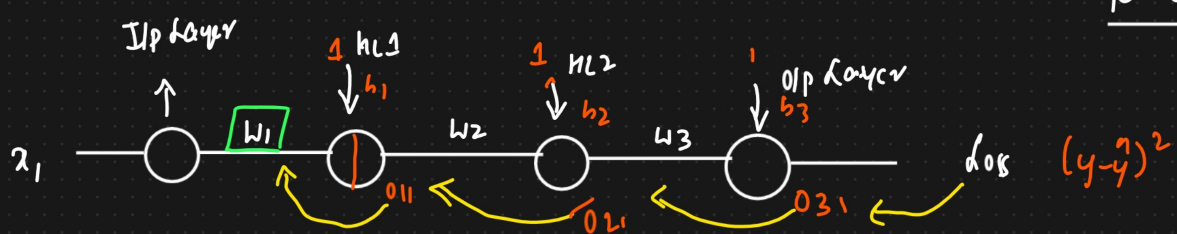
$$+$$

$$\left[\frac{\partial L}{\partial O_{31}} * \frac{\partial O_{31}}{\partial O_{22}} * \frac{\partial O_{22}}{\partial O_{12}} * \frac{\partial O_{21}}{\partial W_{1old}}\right]$$

Dep Layered NN $\Rightarrow$ Sigmoid Activation will not work

8 Vanishing Gradient Problem And Activation functions

$\boxed{0 - 0.25}$ ✓.



Sigmoid Activation

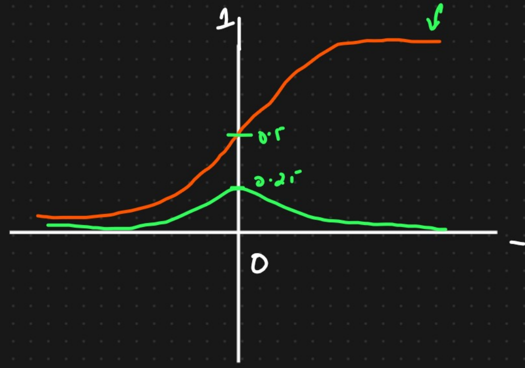
$$\sigma(z) = \boxed{\frac{1}{1+e^{-z}}} \rightarrow 0 - 0.25$$

[0 to 1]

$$\boxed{0 \leq \sigma(z) \leq 1}$$

Derivation of $\sigma(z)$

$$\boxed{0 \leq \sigma(z) \leq 0.25}$$



0.01 ↑ ↓ very small

$$W_{1new} = W_{1old} - \eta \boxed{\frac{\partial L}{\partial W_{1old}}}$$

$\left[W_{new} \approx W_{old} \right]$ Small value

$[0-0.25]$ $[0 \leftarrow$

small value

$$\frac{\partial L}{\partial W_{1old}} = \frac{\partial L}{\partial O_{31}} * \boxed{\frac{\partial O_{31}}{\partial O_{21}}} * \frac{\partial O_{21}}{\partial O_{11}} * \frac{\partial O_{11}}{\partial W_{11}}$$

$$O_{31} = \sigma(W_3 * O_{21} + b_3)$$

$$\boxed{Z = W_3 * O_{12} + b_3}$$

$$O_{31} = \sigma(Z)$$

$$\frac{\partial O_{31}}{\partial O_{21}} = \frac{\partial \sigma(Z)}{\partial O_{21}} = \frac{\partial \sigma(Z)}{\partial Z} * \frac{\partial Z}{\partial O_{21}}$$ {Chain Rule}.

0-0.25

$$\frac{\partial (W_3 * O_{21} + b_3)}{\partial O_{21}} \Rightarrow W_3$$

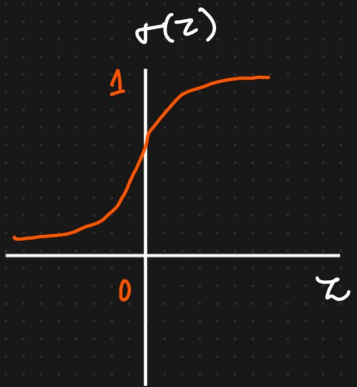
$$\frac{\partial O_{31}}{\partial O_{21}} = [0-0.25] * W_3.$$ ⇒ small value

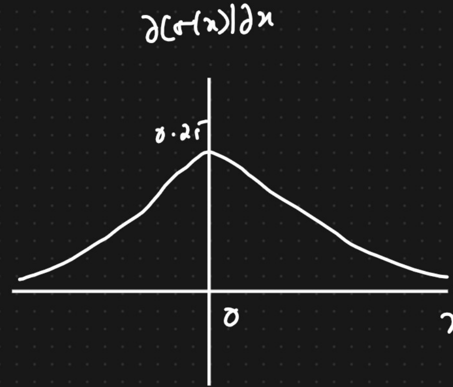
(*) To fix this problem we use other Activation function

(1) Tanh (2) Relu (3) Prelu (4) Elu (5) Swiss.

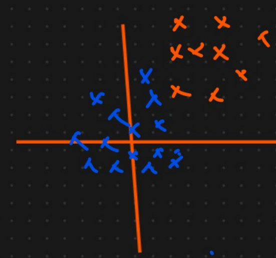
Activation function

① Sigmoid Activation function





Zero centred



Standardization

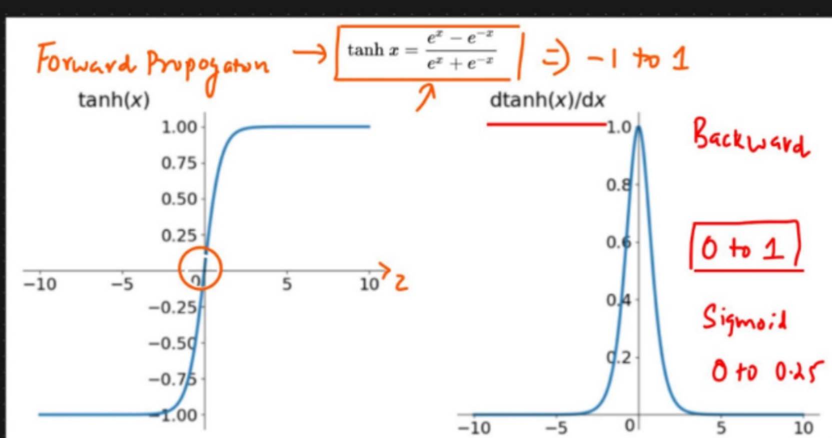
Advantages

① Clear prediction
1 or 0

Disadvantages

① Prone to vanishing gradient problem
② Function output is not zero centred
⇓
Efficient weight updation

② Tanh Activation function



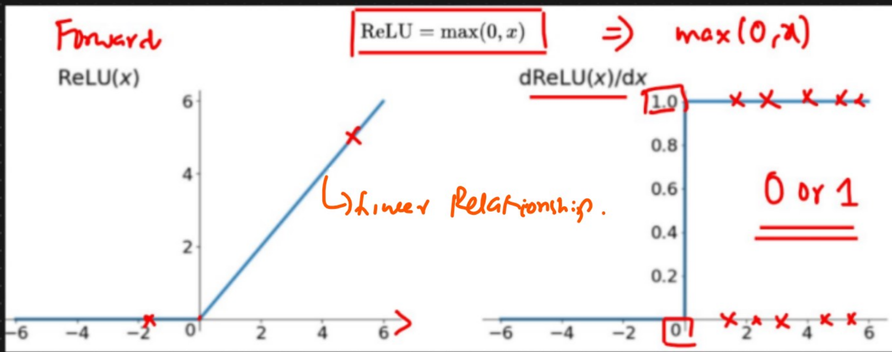
Advantage

① Zero centered

Disadvantage

① Time Complexity
② Vanishing gradient Problem still exist.

③ Relu Activation Function

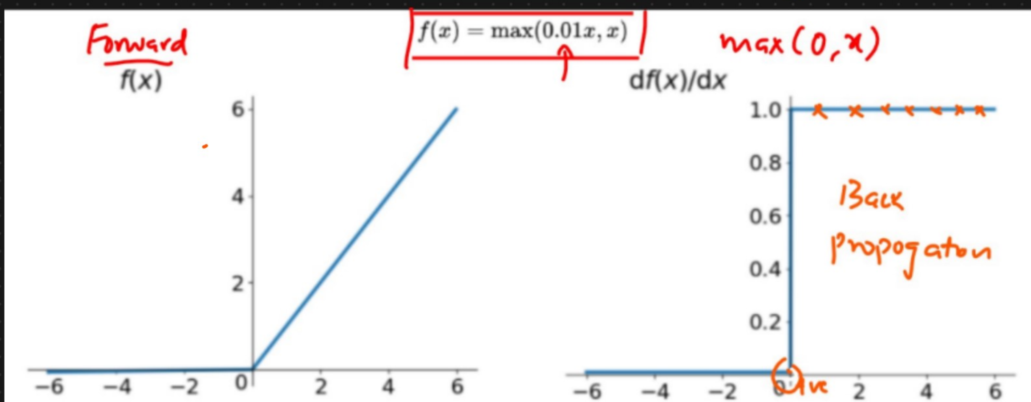


Dead Neuron

④ Leaky Relu or Parametric Relu

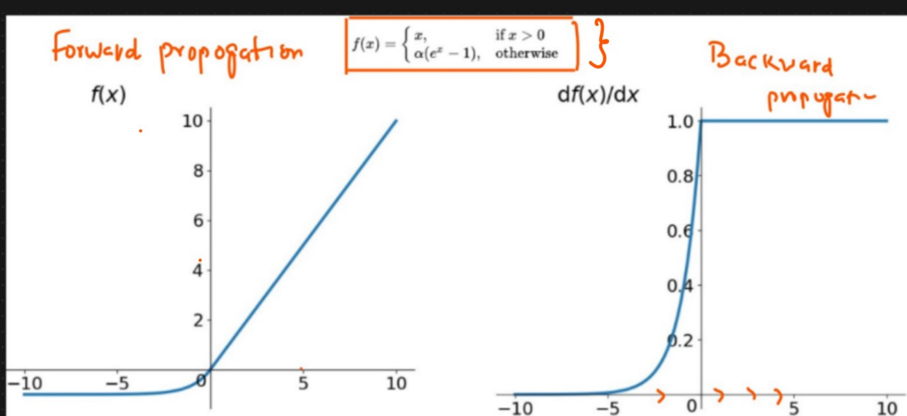
Hyperparameter

max (αx, x)



Prevents Dead Neuron

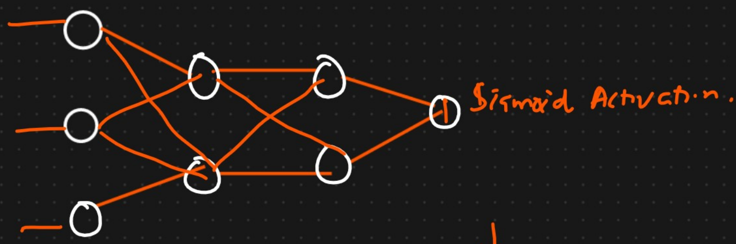
⑤ ELU (Exponential Linear Units)



Disadvantage

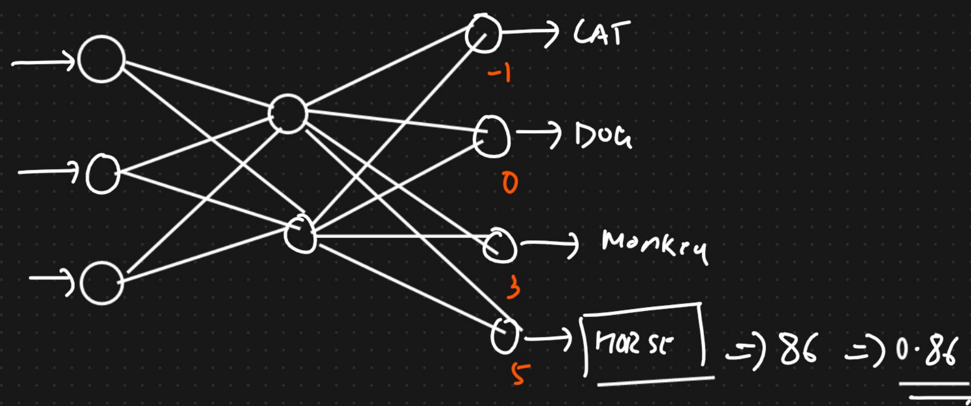
Time consuming

⑥ Softmax Activation Function



Softmax Activation

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



Softmax Activation ⇐

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

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

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

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

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

$\approx$ 86%

Regression ⇒ Linear

Sigmoid → Binary Classif

Softmax ⇒ Multiclass Classification

⑦ Which Activation function To Use When