Data Science & AI

Machine Learning

Unsupervised Learning

Lecture No.- 02

By- Krish Naik Sir

Recap of Previous Lecture

Topic

SVM

Topic

PCA

Topic

Topic

Topic

Slide 2

Topics to be Covered

Topic — Deep Learning

Topic

Topic

Topic

Topic

Deep Learning

① Artificial Neural N/W [Perceptron, Multi Layered Neural N/w]
② Loss functions
③ Activation functions
④ Optimizers → Back Propogation. [Partial Derivatives].
⑤ Weight Initialization.

Why Deep Learning

2005 → Facebook → Social Media   Data ↑↑↑

↑↑↑ DATA → $\boxed{\text{ML}}$ → O/P → Acc

①

② Nvidia → GPU's → Research. → Training Huge Deep Learning Model.

③ Frameworks

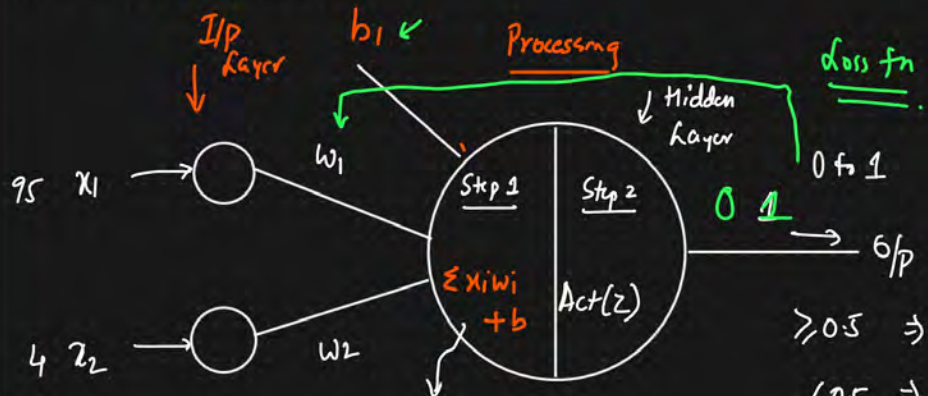
$\boxed{\text{Tensorflow}}$ → Google $\boxed{\text{PYTORCH}}$ → Facebook (Meta)

Perceptron [Artificial Neuron or Neural N/w Unit]

1. I/p Layer ✓
2. Hidden Layer ✓
3. Weights and bias ✓
4. Activation function. ✓

Dataset

x_1	x_2	O/p



0 to 1

$\geq 0.5 \Rightarrow 1$

$< 0.5 \Rightarrow 0$

DATASET

x_1 IQ	x_2 Study hours	O/p
95	4	$\boxed{1}$
100	5	1
95	2	0

w_1, w_2: [0.01, 0.02] bias: [0.001]

$$z = b + x_1 w_1 + x_2 w_2$$

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} [x_1 \; x_2] \Rightarrow \boxed{z = w^T x + b}$$

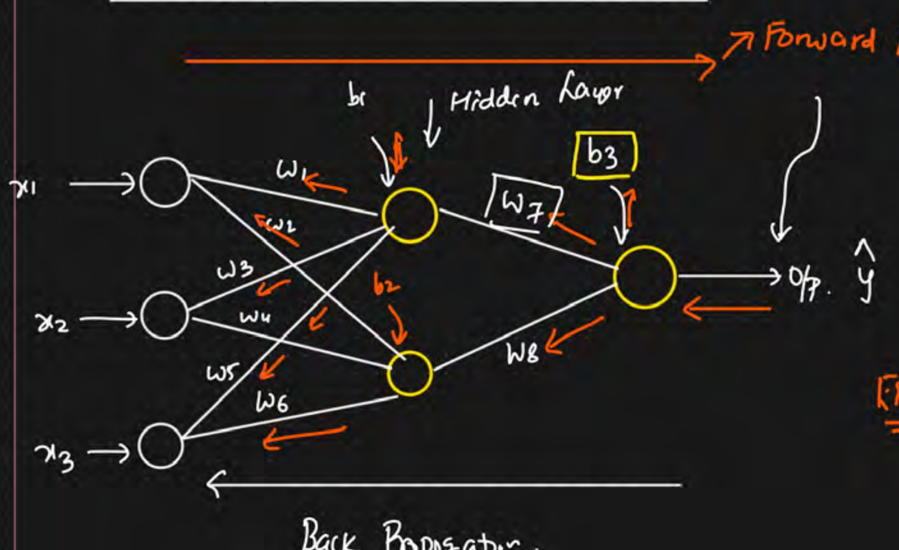
$$\text{Sigmoid} = \frac{1}{1+e^{-z}} \Rightarrow 0 \text{ to } 1$$

$$z = [95 * 0.01 + 4 \times 0.02] + 0.001$$

$$z = 1.031$$

$$\frac{1}{1+e^{-1.031}} = 0.737 \quad \geq 0.5 \Rightarrow 1$$

Multi-Layered NN And Back propogation



Loss function

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

ERROR

GRADIENT DESCENT

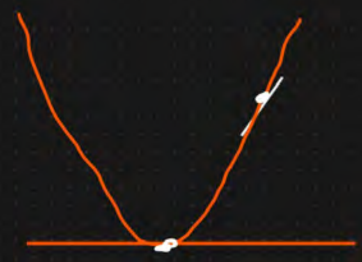
Minimize Optimizers

Weight updation formula

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

Learning

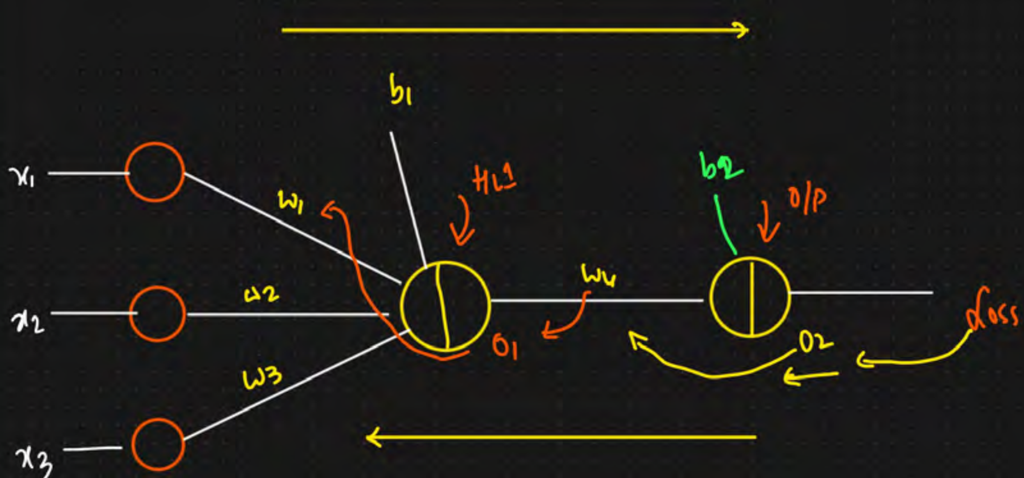
slope



$$W_{8new} = W_{8old} - \eta \boxed{\frac{\partial Loss}{\partial W_{8old}}}$$

Optimizer → Reduce Loss.

7. Chain Rule of Derivative

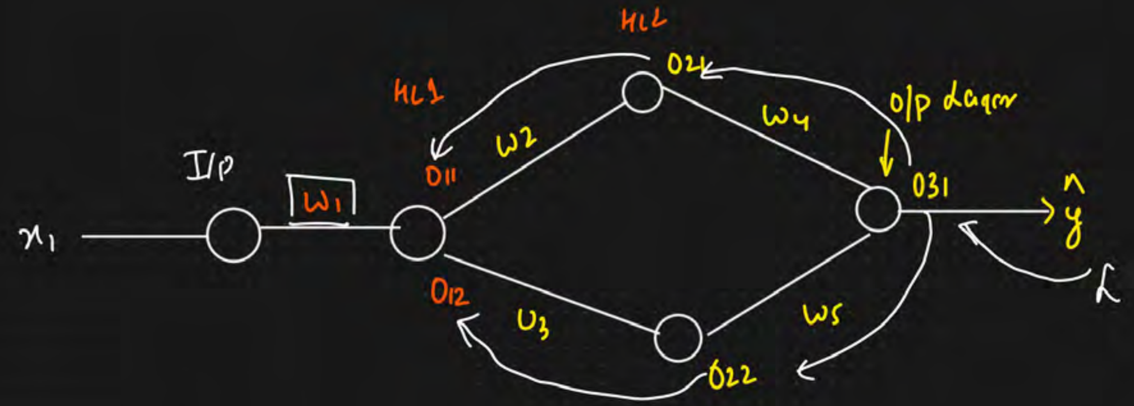


$$\boxed{W_{4new} = W_{4old} - \eta \frac{\partial L}{\partial W_{4old}}}$$

$$\frac{\partial L}{\partial W_{4old}} = \frac{\partial L}{\partial \cancel{O_2}} * \frac{\partial \cancel{O_2}}{\partial W_{4old}}$$ ⇒ Chain Rule of derivative.

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

$$\frac{\partial L}{\partial W_{1old}} = \frac{\partial L}{\partial O_2} * \frac{\partial O_2}{\partial O_1} * \frac{\partial O_1}{\partial W_{1old}}$$ ⇒ Chain Rule of derivative

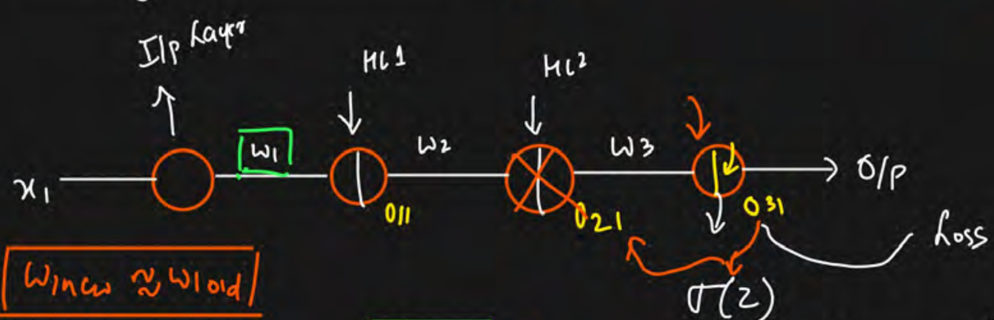


$$W_{1new} = W_{1old} - \eta \boxed{\frac{\partial L}{\partial W_{1old}}}$$

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

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

⑧ Vanishing Gradient Problem [Sigmoid Activation Function].



$\boxed{W_{1new} \approx W_{1old}}$

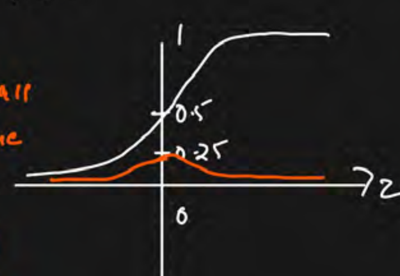
$$W_{1new} = W_{1old} - \eta \boxed{\frac{\partial L}{\partial W_{1old}}} \approx 0$$

Activation function: Sigmoid

$$= \frac{1}{1+e^{-z}} = 0 \text{ to } 1$$

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

⇒ Small Value



Activation Sigmoid

$$O_{31} = \sigma(\underbrace{W_3 * O_{21} + b}_{z})$$

≈ 0

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

$\Rightarrow$ Derivative of Sigmoid.

$$\frac{\partial O_{31}}{\partial O_{21}} = \frac{\partial(\sigma(z))}{\partial(z)} * \frac{\partial z}{\partial O_{21}}$$

$\Downarrow$

$\boxed{0 - 0.25}$

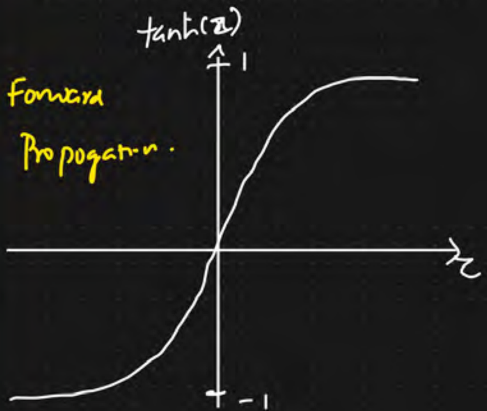
To solve this Change Activation function

② Tanh Activation function

$$\boxed{\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}}$$

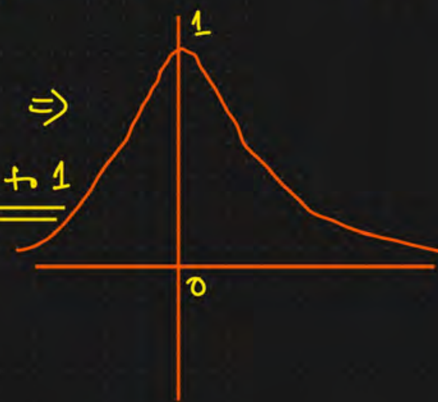
$\Rightarrow -1$ to 1 0 to 0.25

Forward Propogation.



$\Rightarrow$

$\frac{\partial(\tanh(x))}{\partial x} = 0$ to 1



③ Relu Activation [Rectified Linear Unit]

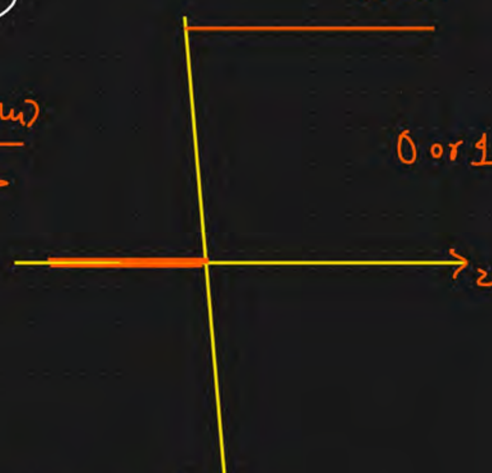
$Relu = \max(0, z)$

It will be either 0 or 1.

Forward Propagation

$\frac{\partial(Relu)}{\partial z}$

0 or 1



⊗

If Derivative of Relu is 1 $\Rightarrow$ Weight updation will happen

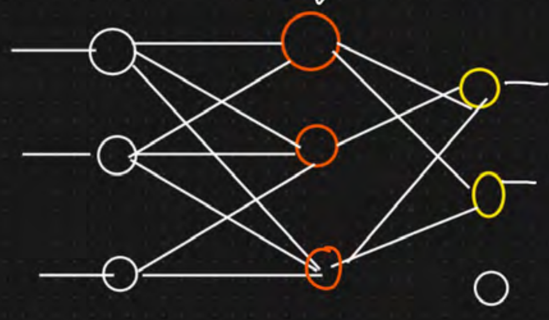
If " " " is 0 $\Rightarrow$ Dead Neuron

Leaky Relu And Paragmetric Relu

Parametric Hyperparameter

$f(x) = \max(0.01x, x)$ 0.01, 0.02, 0.03, 0.04, 0.05

$\frac{\partial(f(x))}{x}$



Relu, Variants of Relu

→ Binary classification

Sigmoid, Softmax ⇒ Multiclass classification

Assignment

Explodient Gradient Descent ⟺ [Weights ⟶ Huge Number].

THANK - YOU