Deep Learning

① ANN → Artificial Neural N/w → Classification / Regression

② CNN → Convolutional Neural N/w → Images, Videos frame → Object Detection

{ RCNN, MASKED RCNN, DETECTRON }

③ RNN → Recurrent Neural N/w → NLP, Time Series

⇓

{ Word2vec, Embedding Layer, doc2vec, Transformers, RNN, LSTM RNN, Bidirectional LSTM RNN, Encoder Decoder, BERT, }

Tensorflow, pytorch

Solving usecases ⇐

④ Why Deep Learning Is Becoming Very Popular?

2004-2005 → ORKUT, FACEBOOK, Instagram, WhatsApp, LinkedIn, Twitter

⇓

{ Images, Text, Documents }. Social Media Platform.

⇓ ⇓

Team in quick go

BIG DATA ⇐ STORE THIS DATA

HADOOP ← Solved ← $$

2011

⇓ 2006-2007

2012-2013 : Company → Data → Improve their product

⇓

Netflix : 2010 → Recommendation System (1 Million $$)

↳ AI Module

Rule Based ⇒ Not Efficient

⇓

AI → popular → Deep Learning

⇓

2005-2006

① → [Huge Data]

① huge available data

② Hardwarde Advancement (NVIDIA) → GPU's → RTX → parallel Training

GPU's ↓↓
↳ integrated → Laptop

libraries

③ ML → F.E → Reduce the FE

$\boxed{\text{Tensorflow}}$, PyTorch
⇓ ⇓
Google Fb

FS → DROP OUT'S,

$\boxed{\text{tf.Keras}}$

>2.0 ⇒ Keras

(.)

$\boxed{\text{KERAS} \rightarrow \text{Tensorflow}}$ ← Tensorflow < 2.0
↓ ⇓
Wrapper Keras → Separate

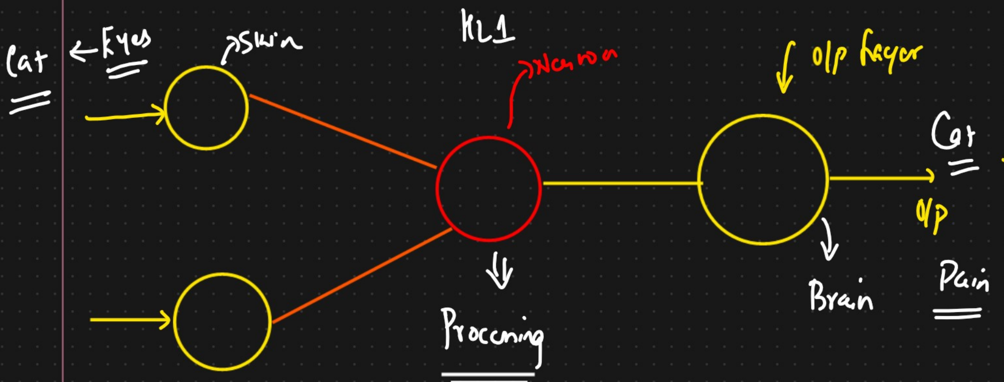
⇑ 2 mentors

[Weekdays → Doubt clearing → Doubts ← 2 mentors ⇒ $\boxed{\text{7-11pm}}$
⇓
ineuron → $\boxed{15}$ → Top → Avnish, Sunny, Sunny Bhaveen, Chandra, Paul, }

④ Perceptron → { Single Layered Neural N/w }

Binary Classification

↓ I/p layer ↓ Hidden layer ⇗ Human Brain



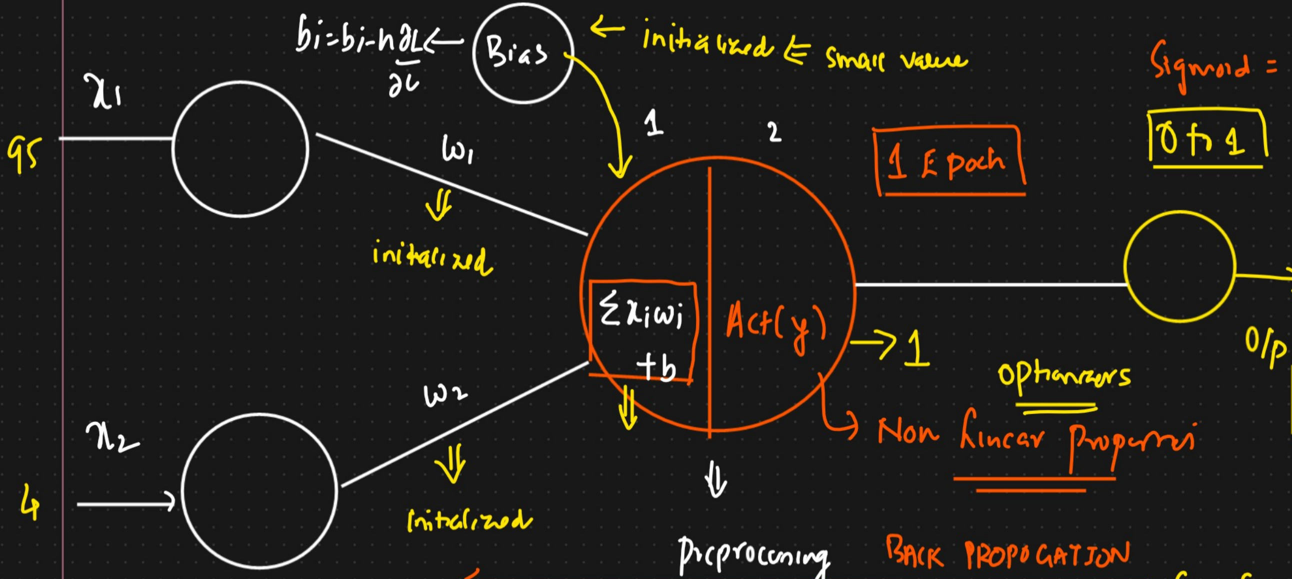
2 Layered NN

x_1 IQ	x_2 No. of hours	Pass/Fail
95	4	0
110	4	1
100	5	1

Dataset

O/p feature

→ Forward Propagation



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

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

$\boxed{1 \text{ Epoch}}$

optimizers

O/p $\boxed{\hat{y} = 1}$ $\hat{y} = 0.8$

$\boxed{y = 0}$ $y = 0$

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

$=$

$(1-0)^2 = \boxed{1}$

$\boxed{x_1 w_1 + x_2 w_2} \Rightarrow \sum x_i w_i \Rightarrow \boxed{w^T x} \Rightarrow$ Linear Regression

Logistic

SVM

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_n x_n$$

Intercept

$$= w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots + w_n x_n$$

$$= \boxed{w^T x} + w_0.$$

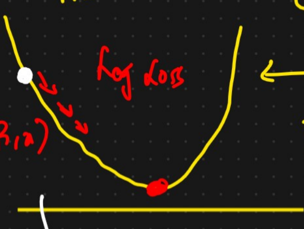
⇓

Linear Regression → straight line

$w_1, w_2, w_3 \cdots \Rightarrow$ coefficients

Learning Rate

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



← GRADIENT DESCENT

⇓

Optimizers

Loss function And Cost function

Loss function → $\rightleftarrows$ Record by Record

Cost function → $\rightleftarrows$ All the Record $= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

1. Perceptron
2. I/p Layers
3. Weights
4. Bias

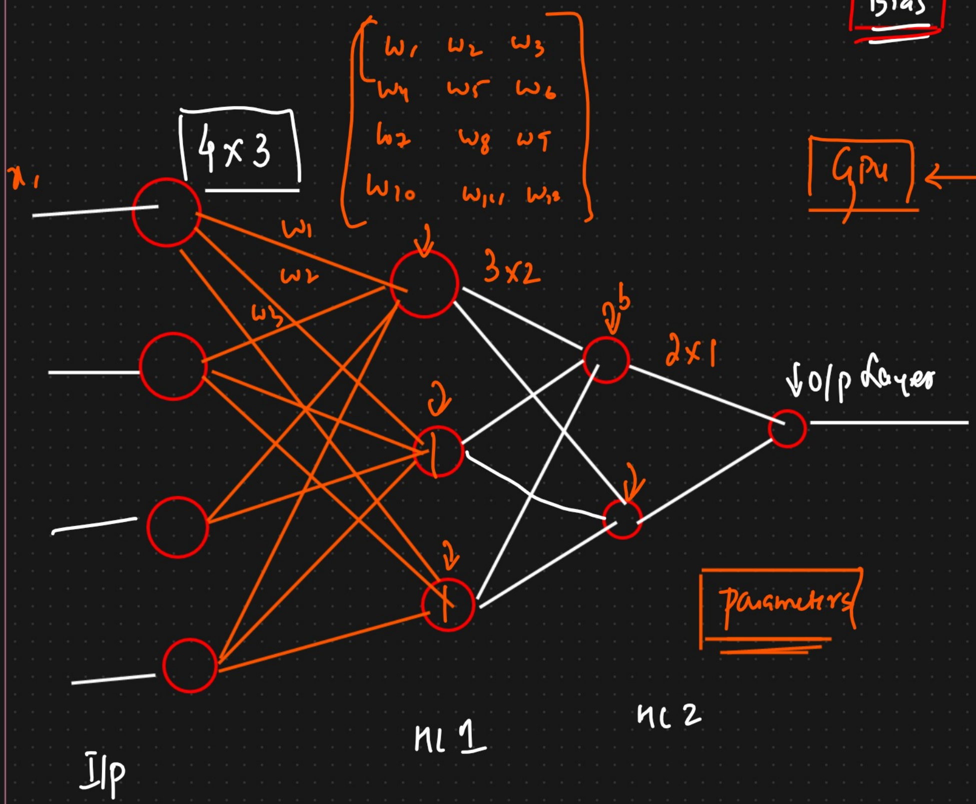
5. Activation function
6. Loss function, Cost function
7. Optimizers
8. Update the weights

$y = mx + \boxed{c}$

$y = \boxed{\beta_0} + \beta_1 x_1$

$\Downarrow$

$\boxed{\text{Bias}}$



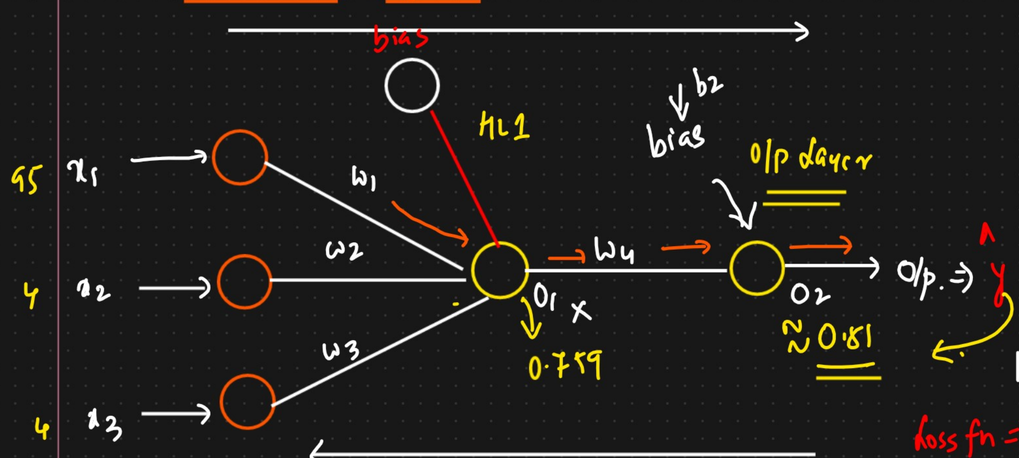
$\boxed{\text{GPU}} \leftarrow$ TRAIN

GPT3

3 Layer Neural N/w

$\boxed{\text{parameters}}$

Forward Propogation



			PASS/FAIL
	No of study hours	play	↓
x_1	x_2	x_3	o/p
95	4	4	1
100	5	2	1
85	2	7	0

$\text{Loss fn} = (y - \hat{y})^2 \downarrow\downarrow\downarrow$

$$\begin{array}{ccc} w_1 & w_2 & w_3 \end{array}$$
$$[0.01, 0.02, 0.03] + [0.001]$$
bias

$[0.02] \leftarrow w_4$

$[0.03] \leftarrow b_2$

Forward propogation

1. $y = 95 * 0.01 + 4 * 0.02 + 4 * 0.03 = 1.15 + 0.001 = \underline{1.151}$

$$L = Act(y) = \frac{1}{1+e^{-(1.151)}} = \underline{\underline{0.759}}$$

② $[0.759 * 0.02] + 0.03 = \underline{\underline{0.04518}}$

Sigmoid

$$Act(0.04518) = \frac{1}{1+e^{-(0.04518)}} = \underline{\underline{0.51129}}$$ ⇒ Binary Classification

③ $Loss = (\overset{y}{1} - \overset{\hat{y}}{0.51}) \Leftarrow \boxed{0.49}$ ↓↓↓↓

④ Backward propagation $\{\hat{y}$ is dependent on weight and bias$\}$

We need to update weights And bias

Optimizers {Gradient Descent}.

Updating weights

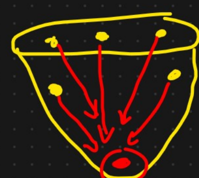
Weight updation formula.

↓ Linear Regression

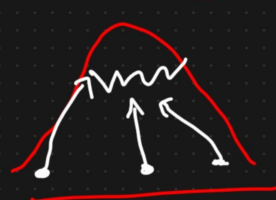
$$W_{4new} = W_{4old} - \eta \boxed{\frac{\partial L}{\partial W_{4old}}}$$ → Slope

{Convergence Formula}

$$\boxed{b_{2new} = b_{2old} - \eta \frac{\partial L}{\partial b_{2old}}}$$



$$\boxed{W_{3new} = W_{3old} - \eta \frac{\partial L}{\partial W_{3old}}}$$



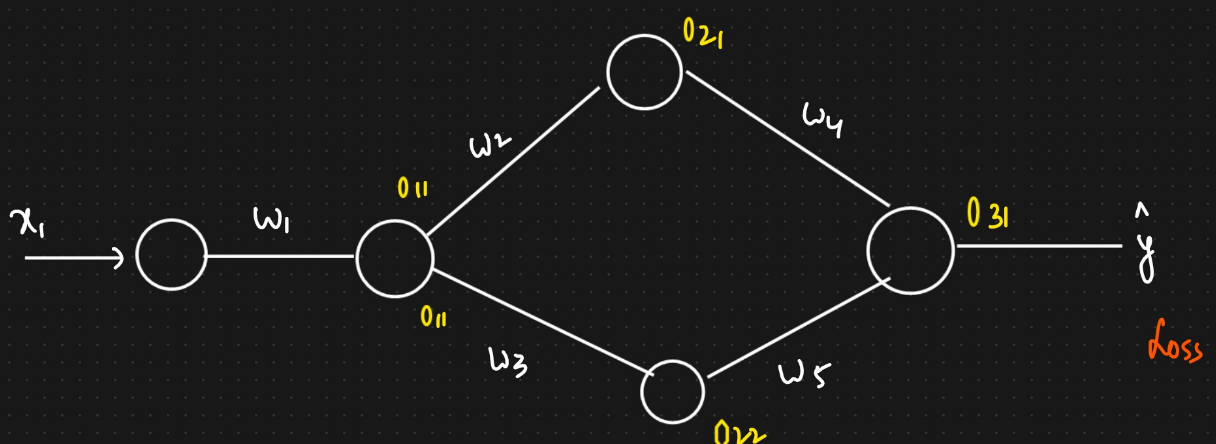
$$\boxed{W_{2new} = W_{2old} - \eta \frac{\partial L}{\partial W_{2old}}}$$

Chain Rule derivative

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

$$\frac{\partial L}{\partial W_{1old}} = \frac{\partial L}{\partial O_2} * \frac{\partial O_2}{\partial O_1} * \frac{\partial O_1}{\partial W_{1old}} \Rightarrow \text{Chain Rule of Derivative}$$

$$\frac{\partial L}{\partial W_{2old}} = \frac{\partial L}{\partial O_2} * \frac{\partial O_2}{\partial O_1} * \frac{\partial O_1}{\partial W_{2old}}$$



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

↓ slope

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

$$+$$

$$\left[\frac{\partial L}{\partial O_{31}} * \frac{\partial O_{31}}{\partial O_{22}} * \frac{\partial O_{22}}{\partial O_{11}} * \frac{\partial O_{11}}{\partial W_{1old}} \right]$$

Chain Rule of Derivative