#LifeKoKaroLift

Data Science Certification Program

Course : Data Science

Lecture On :Intro of NN

Instructor : Shivam Garg

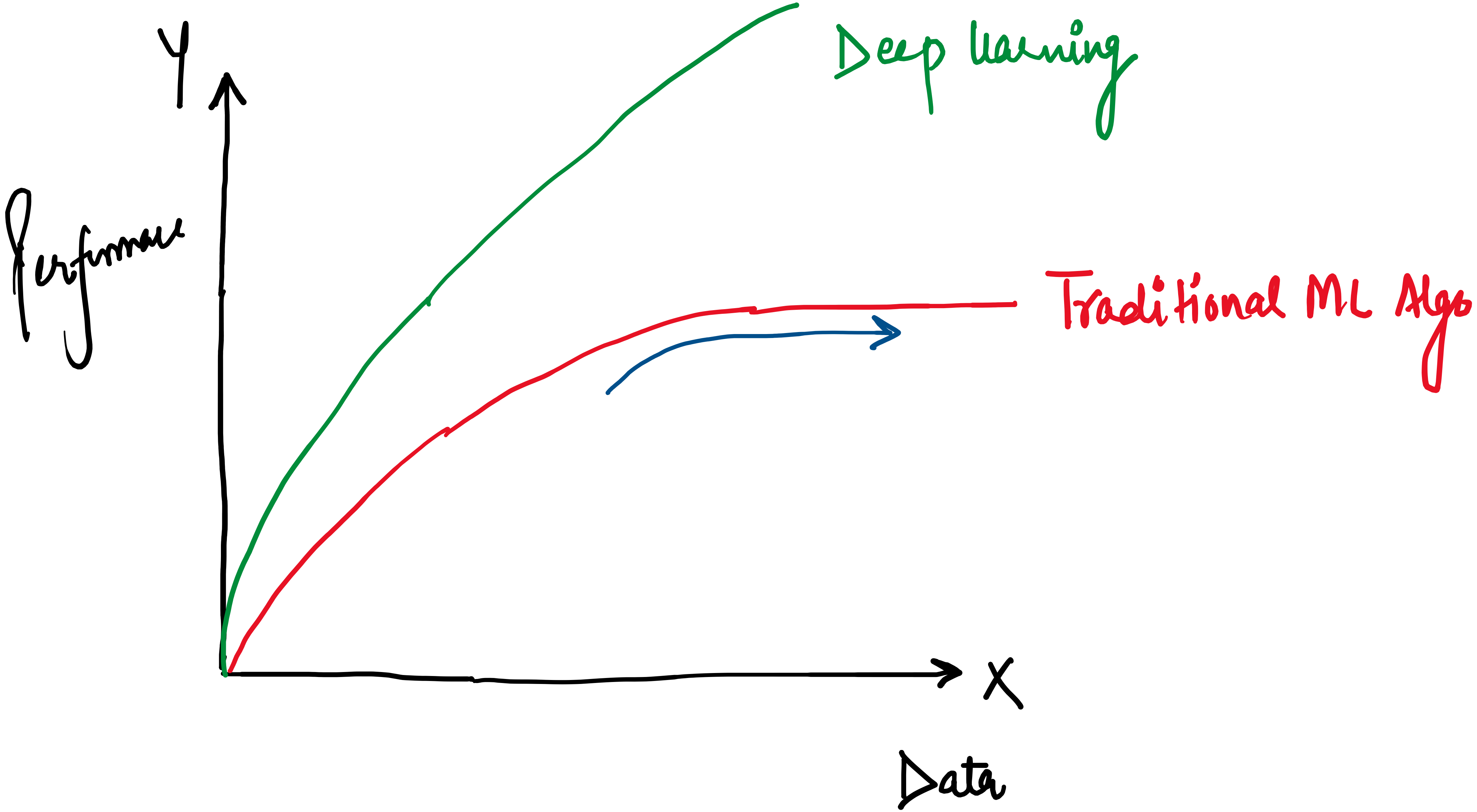
upGrad

Today's Agenda

Need of neural network

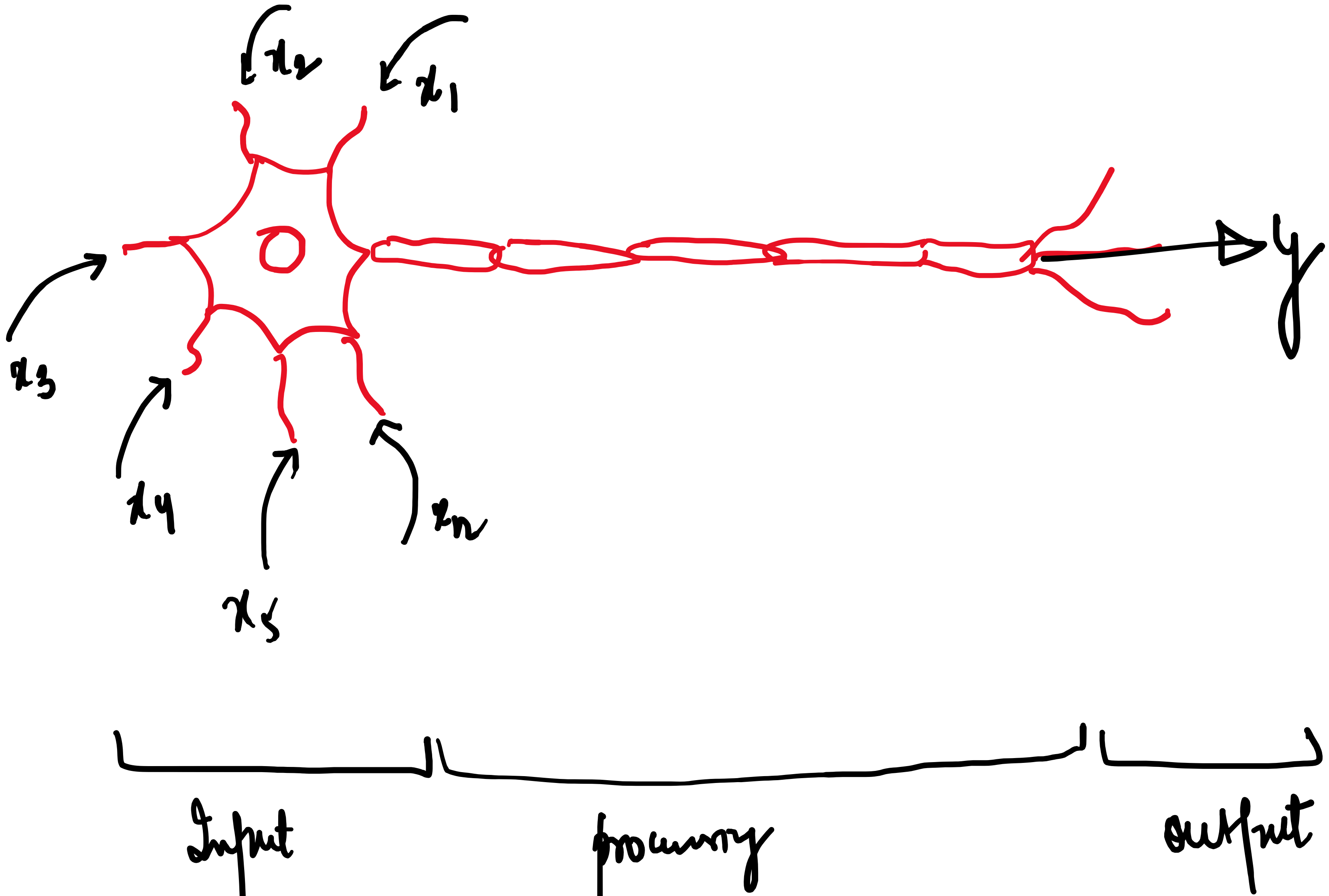
1. Basics of Neural Network
2. Activation Function
3. Forward Pass
4. Back Propagation

Training



Y
Performance
Deep learning
Traditional ML Algo
X
Data

→ Human Neuron :-



Neural networks

Connection (b/w neurons)

Neuron (Basic entity of NN)

* Capability to learn non-linear patterns

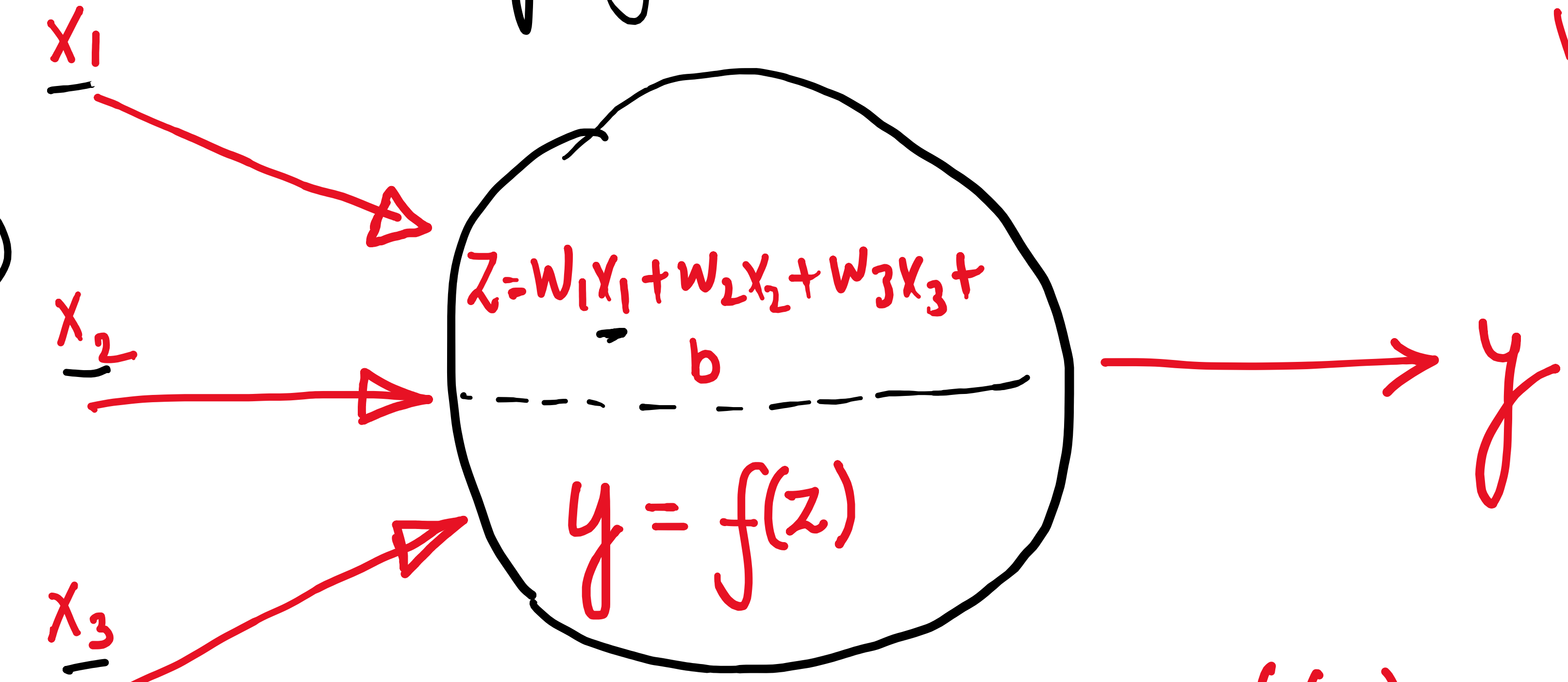
* Neural networks is "Universal function Approximator"

→ Working of Neuron :-

$y = m_1x_1 + m_2x_2 + m_3x_3 + \dots + c$

$w_i \rightarrow$ Weights

$b_i \rightarrow$ b



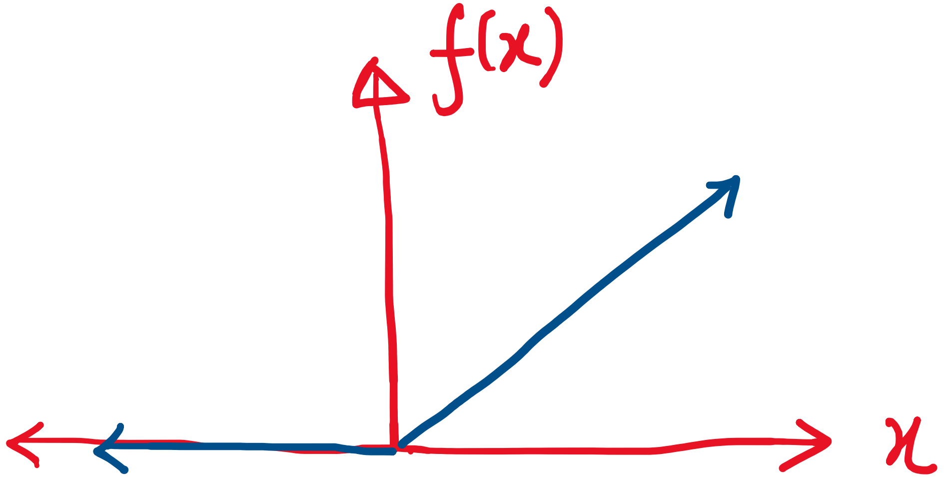
$f(z) \rightarrow$ Activation function

(Non-linear transformation)

Activation function :- These are non-linear mathematical function which are responsible in NN to learn non-linear patterns.

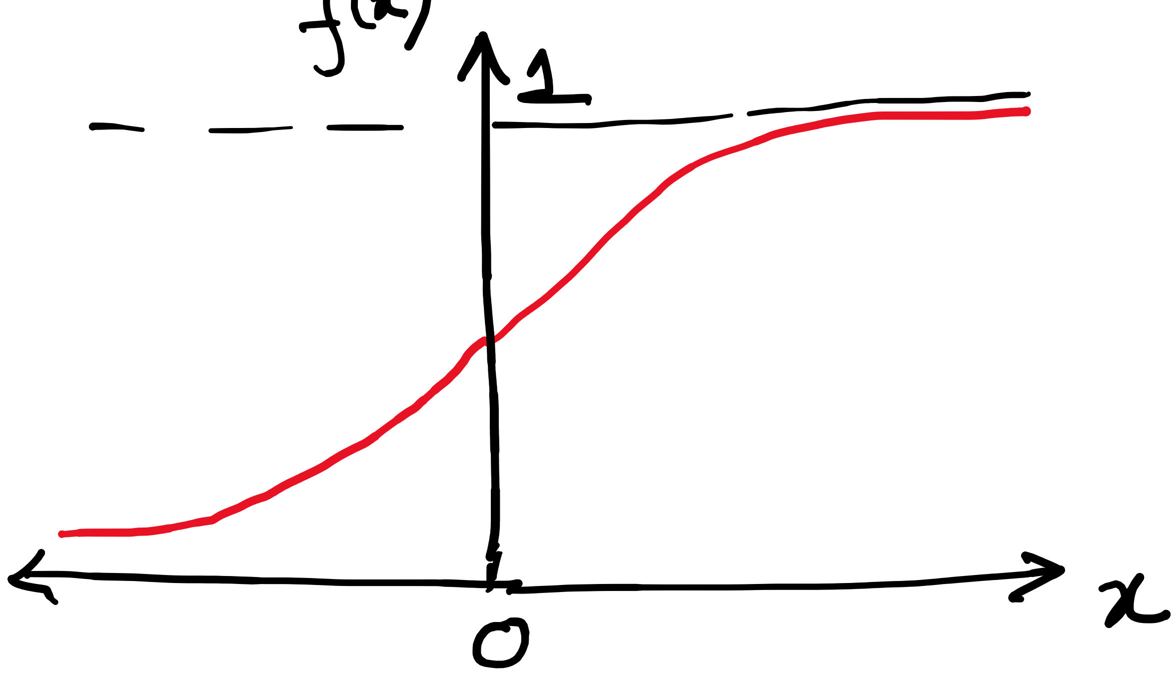
* **ReLU f^n :-** (Rectified Linear Unit)

$$f(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$$



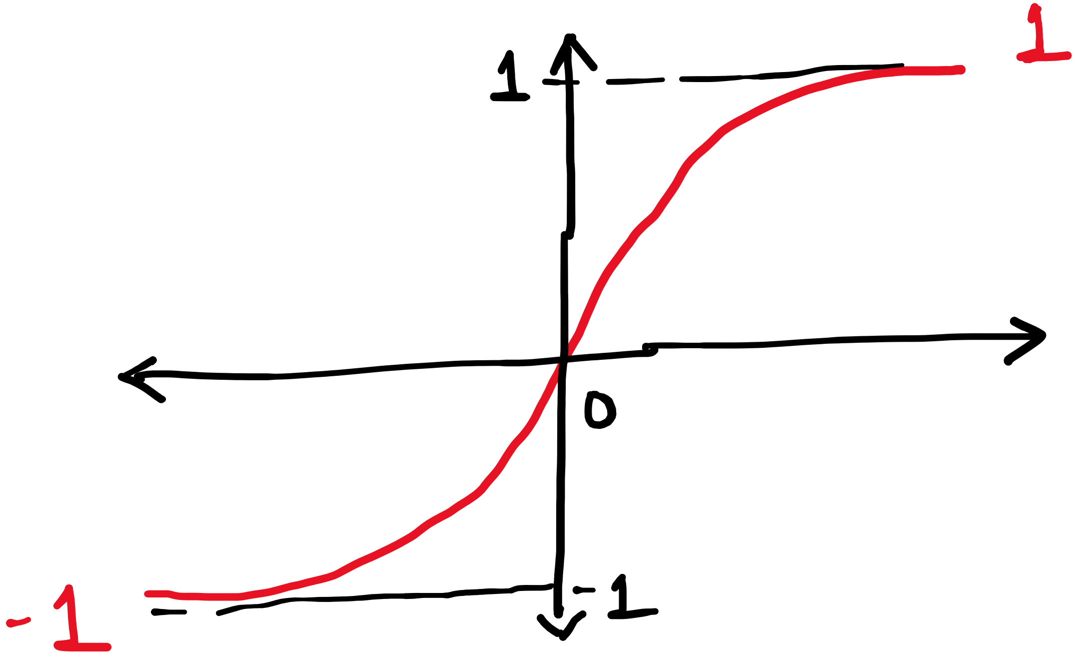
* Sigmoid function :-

$$f(x) = \frac{1}{1+e^{-x}}$$



* Tanh function :-

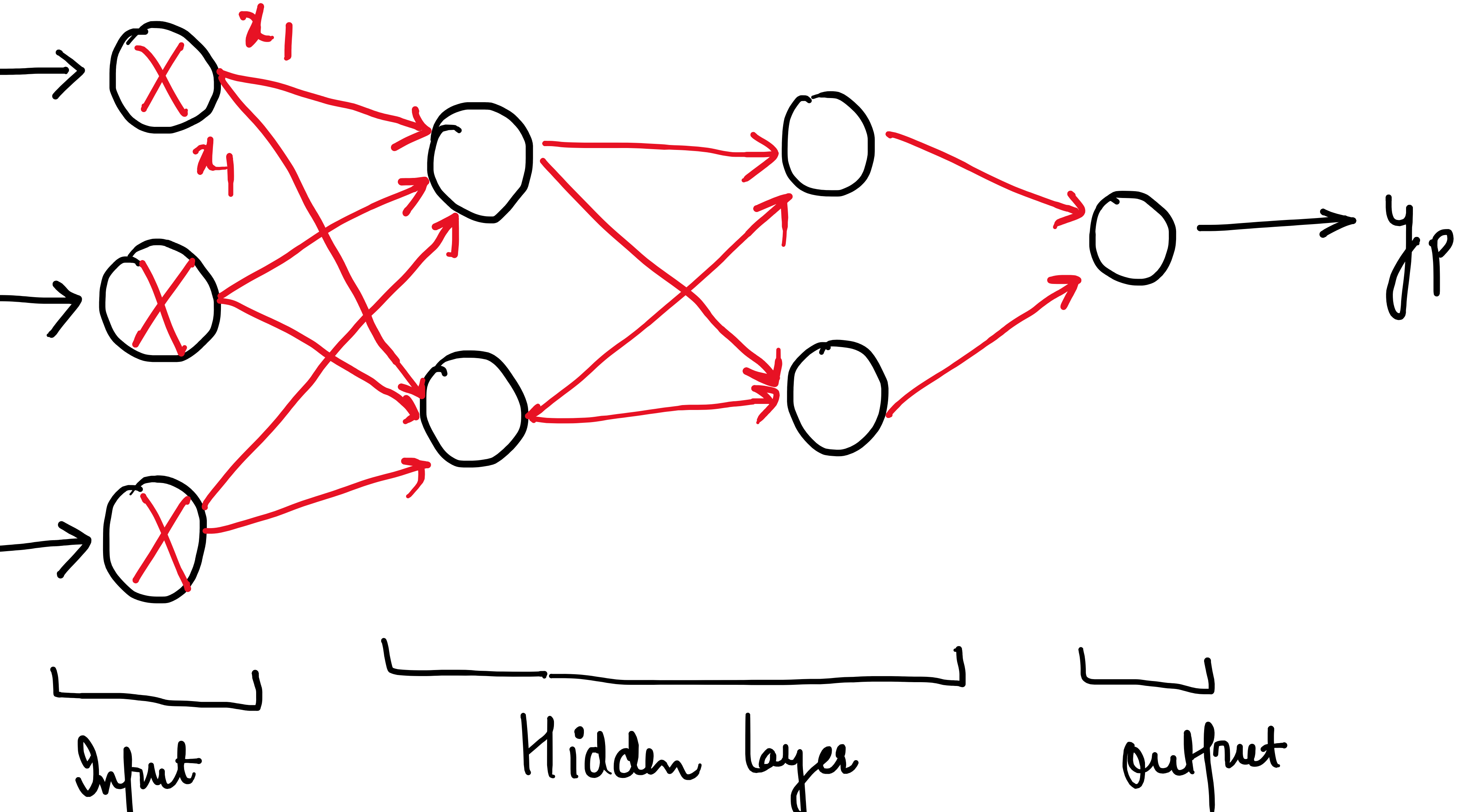
$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$



* Architecture of Neural network :-

○ → Neuron

X_i → Input features



→ <u>Input layer</u> :- (1)

* pass through layer. (no calculation)
* No activation f^n
* No. of neurons is equal to no. of inputs (features)

* Output layer :- (1)

* No. of neurons & Activation f^n depends on use case

Use Case	No. of neurons	Activation f^n
Regression	1	ReLU
Binary Classif^n	1	Sigmoid | tanh
Multiclass classif^n (n)	n (no. of classes)	Softmax

→ Hidden layers :- (n)

* Activation f^n : ReLU

No. of hidden layers
No. of neurons in each layer } hyperparameter

→ **Disadvantages of NN :-**

1. Black box (Non-Interpretable) } *
2. Need to huge amount of data.
3. Need of good Infra Structure }

→ **Advantages of NN :-**

* Very good on Unstructured data

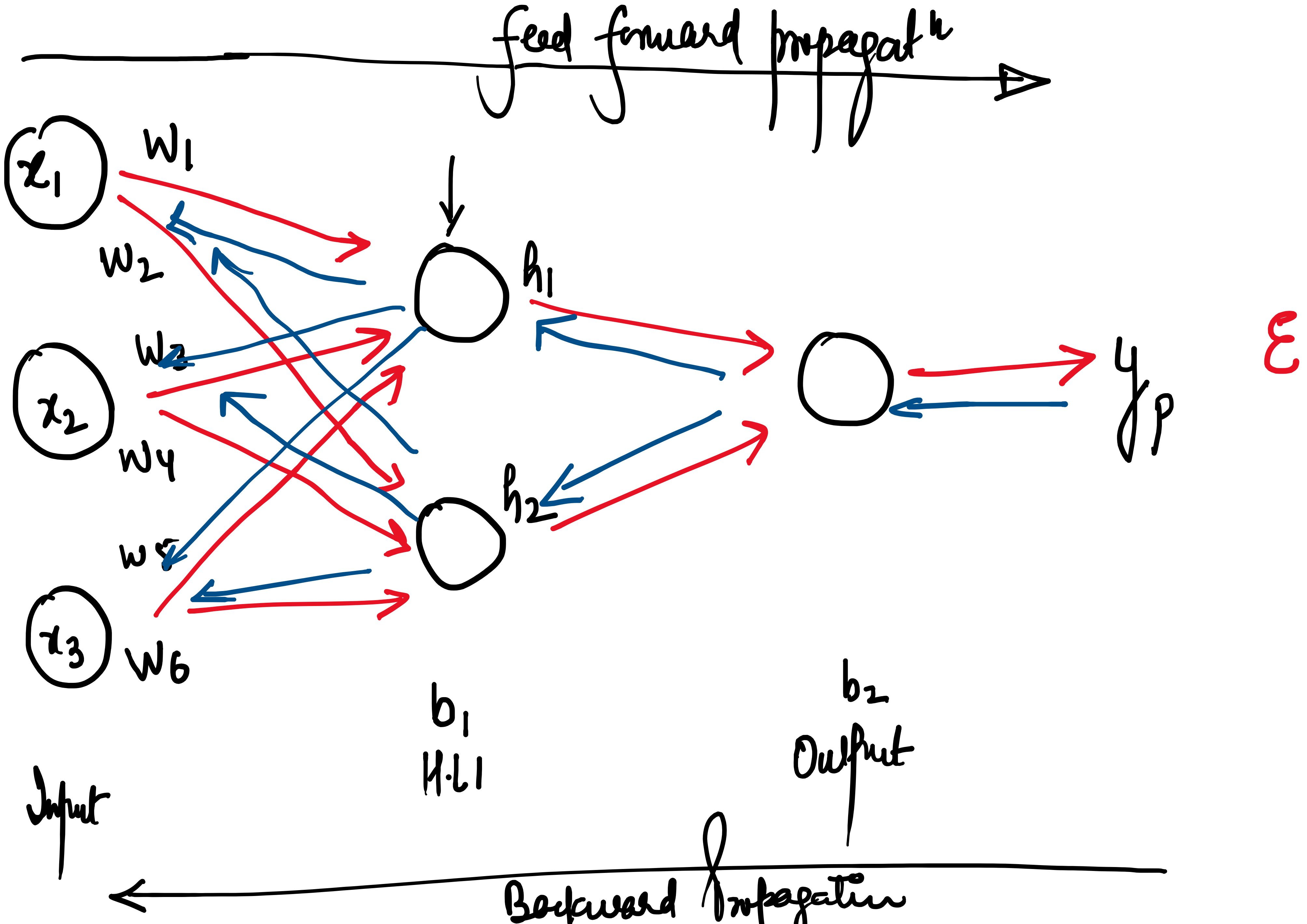
Training

- feed forward
  (to generate the output from input to output layer)
- Back propagation
  (to update weights & bias in order to reduce the error)

1. Random Weights & Bias Initialization.
2. feed forward propagation (y_p)
3. Error f^n (E)
4. Backward propagation (w_i & b_i get updated)
5. Repeat 2, 3 & 4 untill convergence.

y_a / y_p

$\rightarrow$ Error



feed forward propagat^n
x_1
W_1
W_2
h_1
W_3
x_2
W_4
h_2
y_p
$\mathcal{E}$
W_5
x_3
W_6
b_1
H.L1
b_2
Output
Input
Backward Propagation

$$m_i' = m_i - \alpha \underbrace{\frac{\partial E}{\partial m_i}}$$ } Gradient Descent

$\alpha \longrightarrow$ learning rate

(Hyperparameter)

$$W_i' = W_i - \alpha \frac{\partial E}{\partial W_i}$$

$$b_i' = b_i - \alpha \frac{\partial E}{\partial b_i}$$

→ Batch Training:-

Batch size = 100

1000 } → all in

→ in batches

→ RAM

iteration

No. of Batches = 10

R

1B W 2B 3B 10B

100 B 100 100 100

Epoch { complete data is exposed to NN

Total Data points = 1,00,000

Epochs = $\underline{\underline{10}}$ (1 time only)

Batch size = 50

No. of Batches = $\dfrac{\overset{2000}{\cancel{100000}}}{\cancel{50}} = 2000$

No. of iterat^n (1 Epoch) = No. of batches = 2000

Total no. of iterations = $2000 \times 10 = 20000$ } $\underline{\underline{\text{Ans}}}$