D. Khaja Nawaz

RA2311047010037

Deep Learning Techniques (LAB)

Date	Title	Sign
24/07/2025	1. Exploring the deep learning platform	
31/07/25	2. Implement a classifier using open-source dataset	
31/07/25	3. Study of the classifiers with respect to statistical parameters	
14/08/25	4. Build a simple feed forward neural network to recognize handwritten character	
22/08/25	5. Study of Activation functions and their role	
09/09/25	6. Implement gradient descent and backpropagation in deep neural network	

22/08/25

LAB-5

Study of Activation function and their role

* **Aim :**

To study the different activation functions used in neural networks and analyze their role in introducing non-linearity, enabling the network to learn complex patterns

* **Objectives :**

1. To understand the mathematical behaviour of commonly used activation functions
2. To implement sigmoid, Tanh, Relu, and softmax functions
3. To compare the effect of different activation functions on model performance
4. To observe the importance of non-linearity in neural network

* **Pseudocode :**

START

Import necessary Libraries
Define activation functions
sigmoid(x) = 1/(1+exp(-x))
Tanh(x) = (exp(x)-exp(-x))
Relu(x) = max(0,x)
Softmax(x) = exp(x)/sum(exp(x

*Sigmoid :-

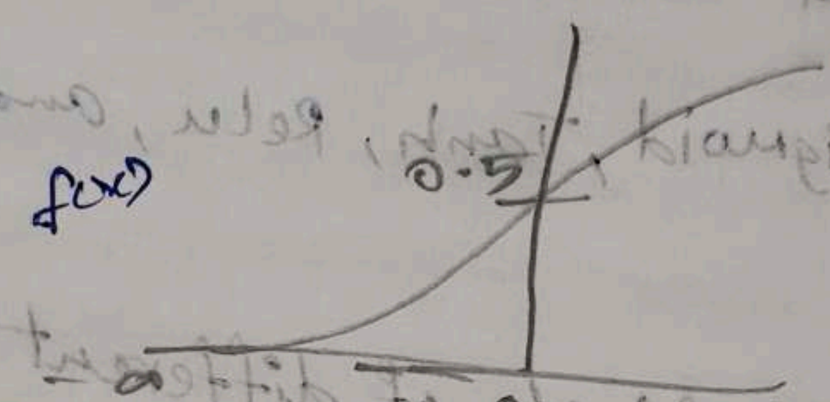
→ Graph has a characteristics S-shaped or sigmoid curve

Formula

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

S(x) → Sigmoid function

e → Euler's Number



Tanh :-

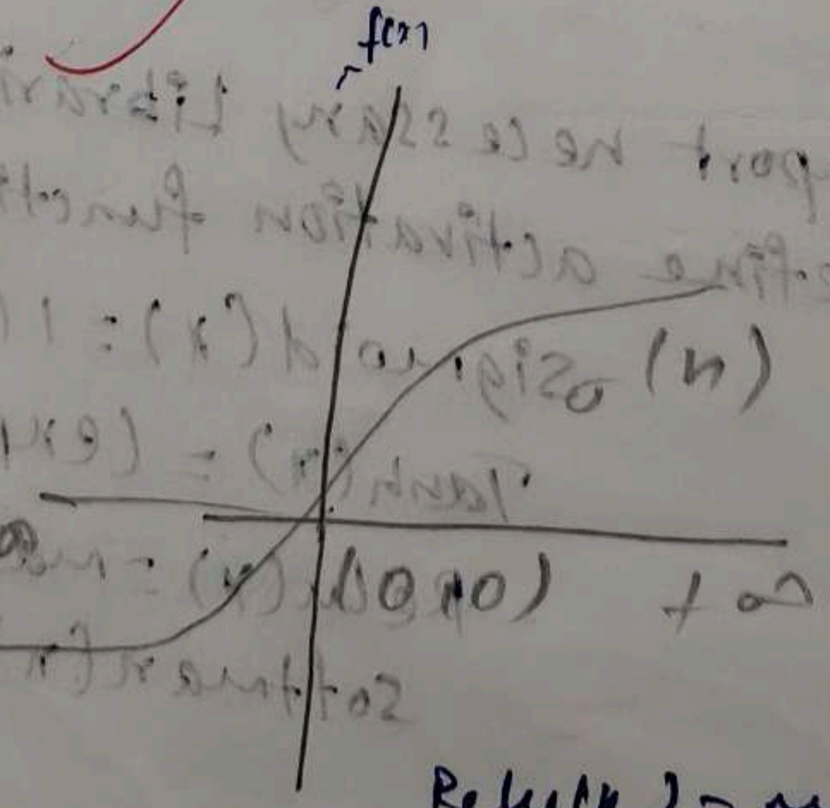
→ Hyperbolic tangent

→ It in the hidden layer of inculding recurrent neural net.

→ Zero-centered output

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)}$$

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



*Relu :-

$$Relu(x) = \max(0, x)$$

→ Defines the Relu function

→ Returns x if it's positive, otherwise returns

Load dataset(MNIST)

For each activation function in [sigmoid, Tanh, Relu],

Build a neural network with that activation

Train the network on training data

Evaluate on test data

store accuracy

compare accuracy results

END

* Observations :-

1. Sigmoid :- smooth, but differs from vanishing gradient for large positive/negative inputs

2. Tanh :- Better than sigmoid, outputs range (-1, 1) but still faces vanishing gradient

3. Relu :- most effective in deep networks, avoids vanishing gradient, fast convergence

4. softmax :- used in output layer for multi-classificati problems

* conclusion :-

Activation functions introduce non-linearity into neural networks, enabling them to approximate complex mapping between inputs and outputs.