Experiment-5 - Study of Activation Functions and its role

9/9/25

Aim: To study and analyze different activation functions used in deep learning and neutral networks, understand their mathematical formulation, visualize their graphs, and observe their effect on model training.

Description: Activation functions introduce non-linearity into ~~neutral~~ neural network, enabling them to learn complex patterns. without activation layer, the network would get like a linear regression model regardless of the number of layers.

Sigmoid function

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

range: $(0,1)$

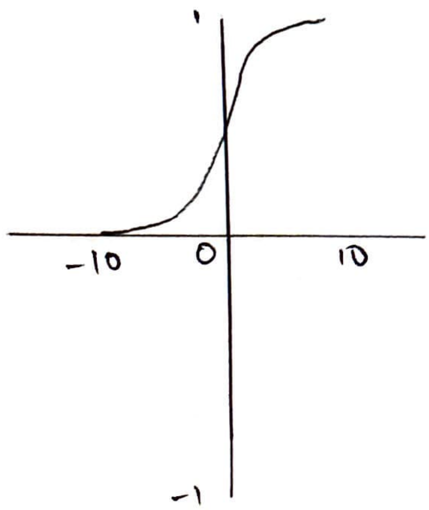
:) Suffers from vanishing gradient

Hyperbolic tangent (tanh)

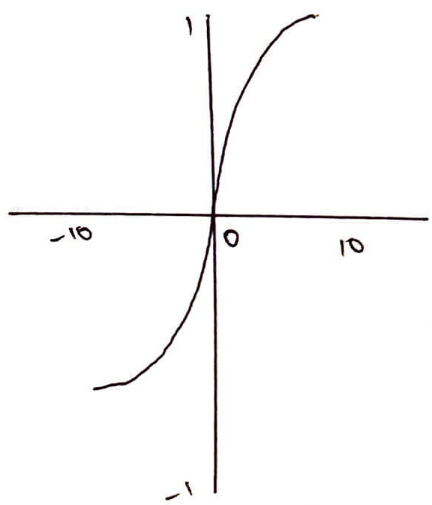
$$f(x) = \tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}.$$

range: $(-1,1)$

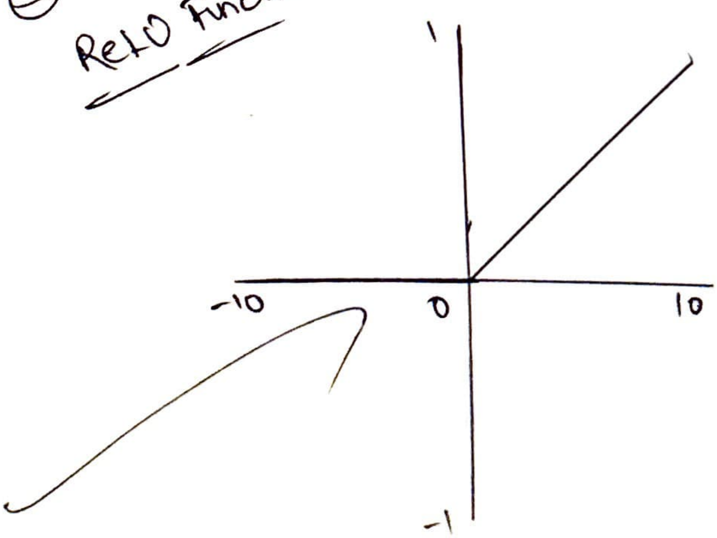
① Sigmoid Function



② Tanh Function



③ ReLU Function



centered at zero
also suffers from vanishing gradient.

iii) Rectified Linear Unit (ReLU):

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

range : $[0, x]$

avoids vanishing gradients

iv) Leaky ReLU:

$$f(x) = \begin{cases} x, & x > 0 \\ x, & x \le 0. \end{cases}$$

allows small negative slope.

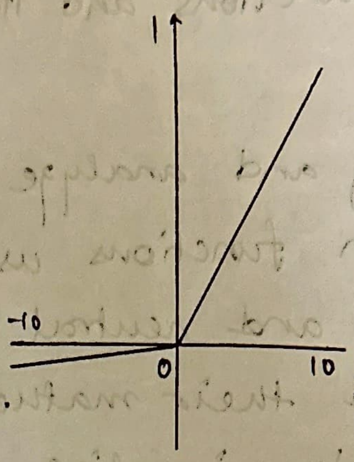
(v) Softmax Function

$$f(x) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

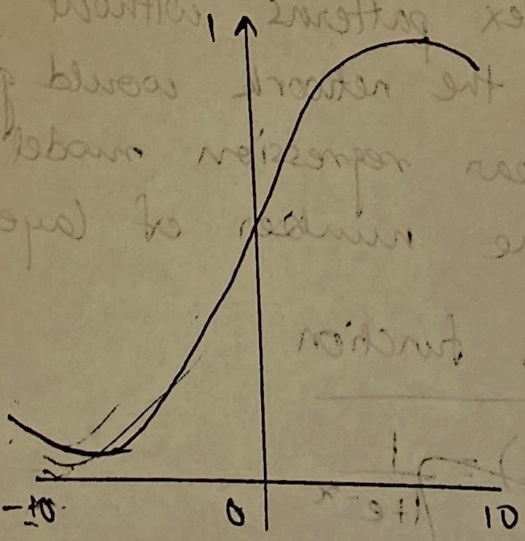
Procedure:

1. Import necessary libraries (numpy, matplotlib)
2. Define each activation function.
3. Generate input values and compute output.
4. Plot graphs of each activation function.
5. Observe the properties.

iv) Leaky ReLU:



v) Softmax Function:



Observation:

i) Sigmoid: Smooth shaped, compresses values between (0, 1), but gradients vanish

ii) Tanh: Similar to sigmoid; outputs values between -1 and 1

iii) ReLU: Outputs 0 for negative inputs and linear for positive.

iv) Leaky ReLU: Similar to ReLU but allows a small, non-zero gradient.

v) Softmax: Converts raw score into probabilities across multiple classes.

Result:

Implementation of activation functions commonly used in deep learning was successfully done.