

Experiment - 5 - Study of Activation Functions and its role

Aim:

To study and analyse different activation functions used in deep learning and neural 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 act like a linear regression model regardless of the number of layers.



Sigmoid function

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

range: $(0, 1)$

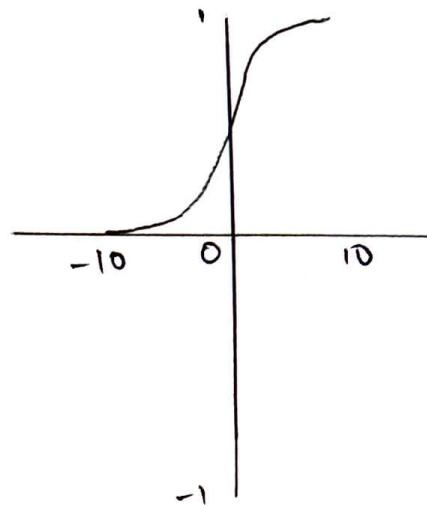
→ Suffers from vanishing gradient

Hyperbolic tangent (tanh)

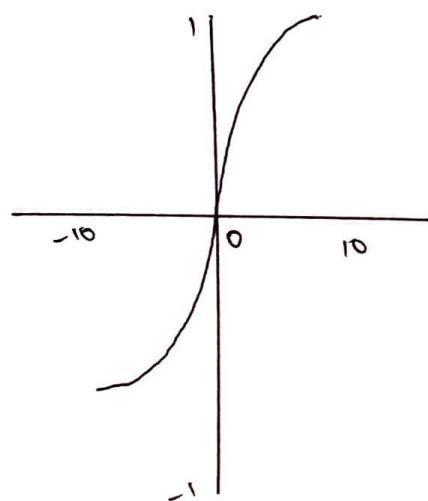
$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

range: $(-1, 1)$

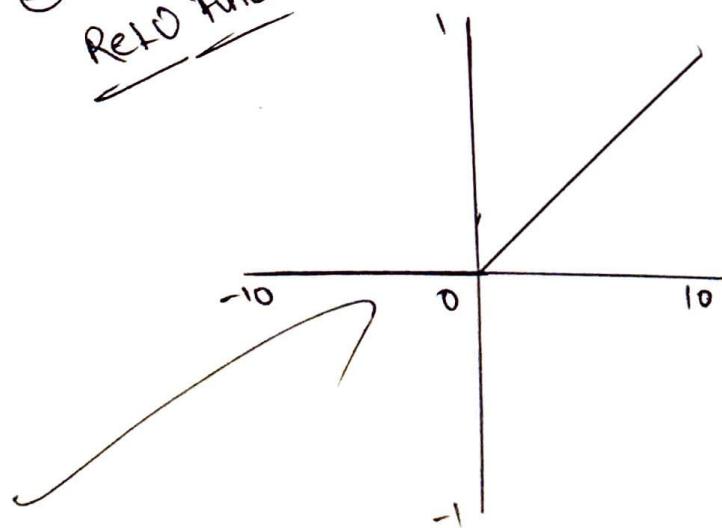
① Sigmoid Function



② Tanh Function



③ ReLU Function



centered at zero
also suffers from vanishing gradient.

iii) Rectified Linear Unit (ReLU):

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

$$\text{range} = [0, \infty]$$

avoids vanishing gradients

iv) Leaky ReLU:

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

allows small negative slope.

v) Softmax Function

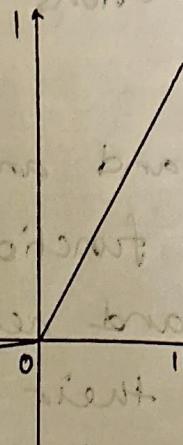
$$f(x) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

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.

iii) Leaky ReLU

graph: $y = \max(0.01x, 0)$



Leaky ReLU approximates the behavior of

gradient $\begin{cases} 1 & \text{if } x > 0 \\ 0.01 & \text{if } x \leq 0 \end{cases}$ (primes)

Leaky ReLU has the following properties:
- always non-zero gradient, no vanishing gradient
- learning rate depends on the slope of the function

- non-saturating gradient (not flat)

v) Softmax function: ~~choose one~~ ~~softmax~~

most of next problems concern

softmax function (softmax = exp(x) / sum(exp(x)))

will top layer neuron act, if output

softmax layer receives signals from all other neurons

(e.g.) to predict cat to

non-dominant category

$$z = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}$$

$(z, 0)$: sparse

~~softmax~~ ~~function~~ most common:

(softmax) function softmax

$$\frac{e^{z_i}}{\sum e^{z_j}} = \frac{e^{z_i}}{e^{z_1} + e^{z_2} + \dots + e^{z_n}}$$

$(z, 1)$: sparse

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.

