Network Login | exp5.ipynb (auto-D : | Inbox (3,125) - ds107 | ChatGPT | 21AIC301J- Lab AI-A | Welcome To Colab -

Not secure 10.1.38.19/user/ra2311047010033/lab/workspaces/auto-D/tree/DLT/exp5.ipynb

School

File Edit View Run Kernel Tabs Settings Help

exp2.ipynb | exp3.ipynb | exp4.ipynb | exp5.ipynb

Code

Notebook | Python 3 (ipykernel)

Filter files by name

/ DLT /

Name	Last Modified
exp2.ipynb	last month
exp3.ipynb	last month
exp4.ipynb	last month
exp5.ipynb	14 days ago
exp6.ipynb	13 days ago
exp7.ipynb	34 minutes ago
Untitled.ipy...	13 days ago

[1]:

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-10, 10, 400)

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def sigmoid_derivative(x):
    s = sigmoid(x)
    return s * (1 - s)

def tanh(x):
    return np.tanh(x)

def tanh_derivative(x):
    return 1 - np.tanh(x)**2

def relu(x):
    return np.maximum(0, x)

def relu_derivative(x):
    return np.where(x > 0, 1, 0)

def leaky_relu(x):
    return np.where(x > 0, x, 0.01 * x)

def leaky_relu_derivative(x):
    return np.where(x > 0, 1, 0.01)

plt.figure(figsize=(14, 10))
```

Simple 0 4 Python 3 (ipykernel) | Idle Mem: 281.54 MB Mode: Command Ln 1, Col 1 exp5.ipynb 0

Type here to search ENG US 10:34 23-09-2025

Network Login | exp5.ipynb (auto-D : | Inbox (3,125) - ds107 | ChatGPT | 21AIC301J- Lab AI-A | Welcome To Colab -

Not secure 10.1.38.19/user/ra2311047010033/lab/workspaces/auto-D/tree/DLT/exp5.ipynb

School

File Edit View Run Kernel Tabs Settings Help

exp2.ipynb | exp3.ipynb | exp4.ipynb | exp5.ipynb

Code

Notebook | Python 3 (ipykernel)

Filter files by name

/ DLT /

Name	Last Modified
exp2.ipynb	last month
exp3.ipynb	last month
exp4.ipynb	last month
exp5.ipynb	14 days ago
exp6.ipynb	13 days ago
exp7.ipynb	34 minutes ago
Untitled.ipy...	13 days ago

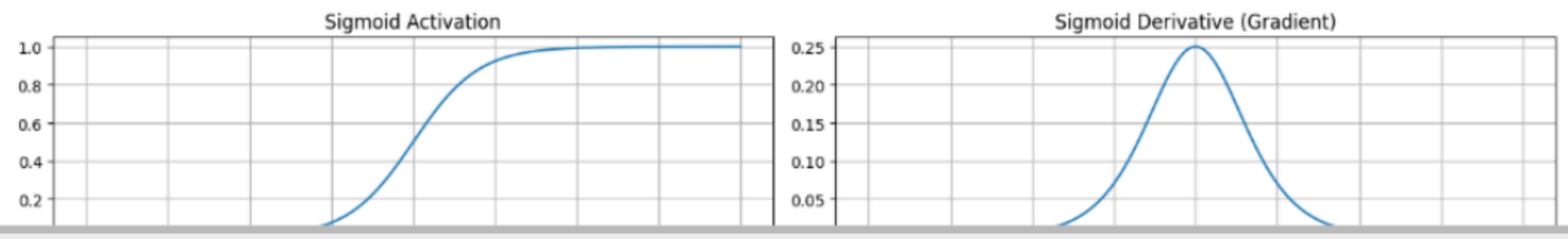
```
plt.figure(figsize=(14, 10))

activations = {
    'Sigmoid': (sigmoid, sigmoid_derivative),
    'Tanh': (tanh, tanh_derivative),
    'ReLU': (relu, relu_derivative),
    'Leaky ReLU': (leaky_relu, leaky_relu_derivative)
}

for i, (name, (act_func, der_func)) in enumerate(activations.items(), 1):
    plt.subplot(4, 2, 2*i - 1)
    plt.plot(x, act_func(x))
    plt.title(f'{name} Activation')
    plt.grid(True)

    plt.subplot(4, 2, 2*i)
    plt.plot(x, der_func(x))
    plt.title(f'{name} Derivative (Gradient)')
    plt.grid(True)

plt.tight_layout()
plt.show()
```



Simple 0 4 Python 3 (ipykernel) | Idle Mem: 281.54 MB Mode: Command Ln 1, Col 1 exp5.ipynb 0

Type here to search ENG US 10:34 23-09-2025

Network Login | exp5.ipynb (auto-D : | Inbox (3,125) - ds107 | ChatGPT | 21AIC301J- Lab AI-A | Welcome To Colab -

Not secure 10.1.38.19/user/ra2311047010033/lab/workspaces/auto-D/tree/DLT/exp5.ipynb

School

File Edit View Run Kernel Tabs Settings Help

exp2.ipynb | exp3.ipynb | exp4.ipynb | exp5.ipynb

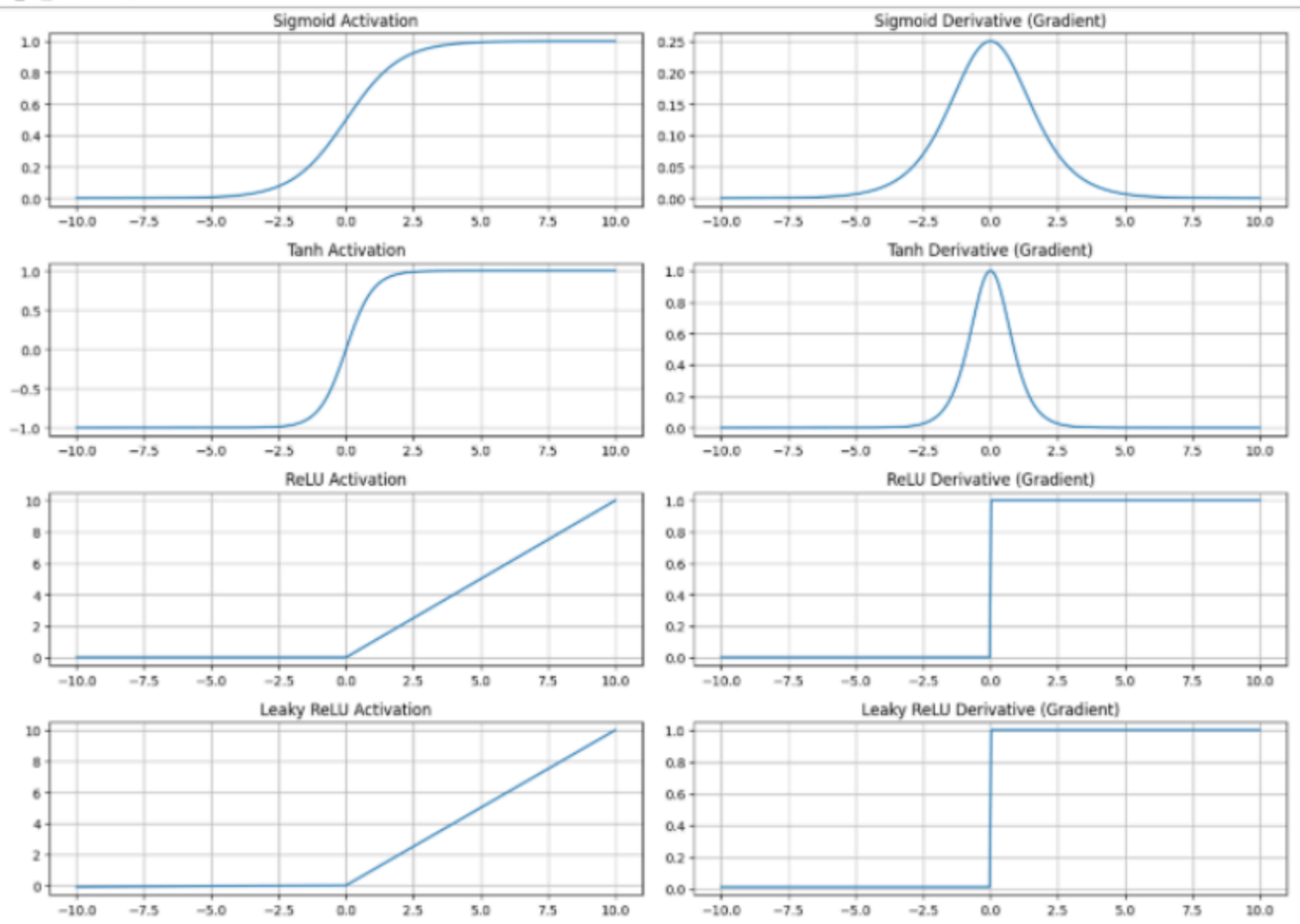
Code

Notebook | Python 3 (ipykernel)

Filter files by name

/ DLT /

Name	Last Modified
exp2.ipynb	last month
exp3.ipynb	last month
exp4.ipynb	last month
exp5.ipynb	14 days ago
exp6.ipynb	13 days ago
exp7.ipynb	35 minutes ago
Untitled.ipy...	13 days ago



Simple | 0 4 | Python 3 (ipykernel) | Idle | Mem: 281.58 MB

Mode: Command | Ln 1, Col 1 | exp5.ipynb | 0

Type here to search

ENG US 10:35 23-09-2025

ex:-5. Study of activation functions and its role

22/08/25

Aim:-

To study the differencte activation fonctions used in neoral networks and analyze their role 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 fonctions
3) To compare the effect of different activation fonctions on model performance
4) To observe the importance of nonlinearity in neural network

* Pseodo code:-

Import necessary Libraries
Define activation fonctions

Sigmoid (x) = 1/(1+exp(-x))
Tanh (x) = (exp(x) - exp(-x))
Relu (x) = max(0,x)
Softmax (x) = exp(x) / (sum(exp(x))

load data set (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 Inegative inputs

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

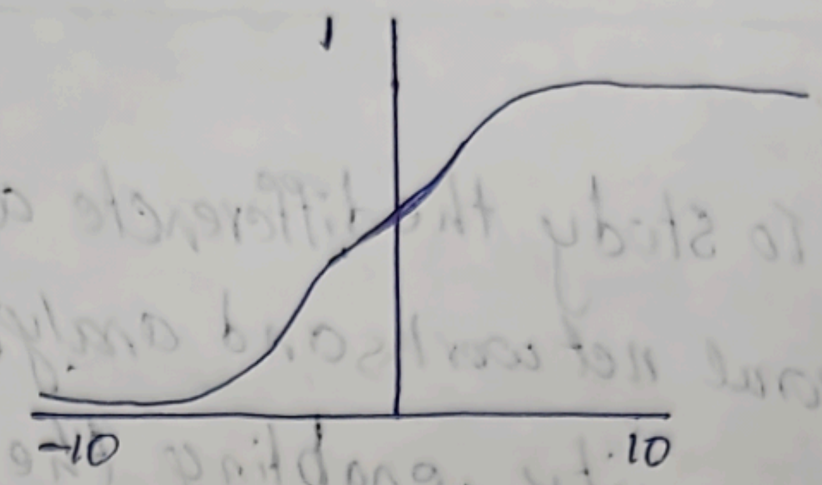
3). ReLu :- most effective in deep net works, avoids vanishing gradient, fast convergence.

4) Softmax :- used in output layer for multi-classification problems

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

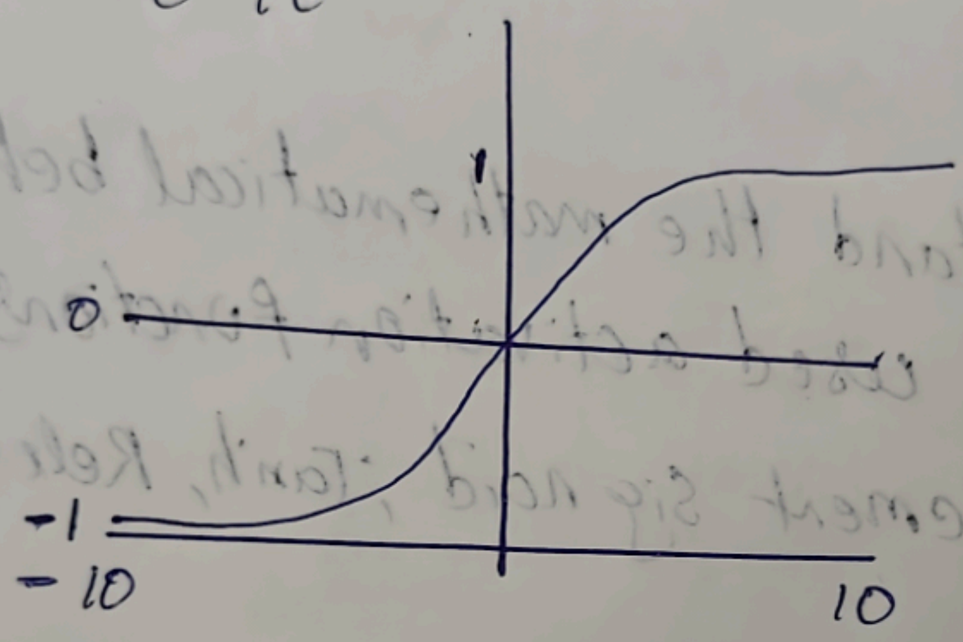
Sigmoid:

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



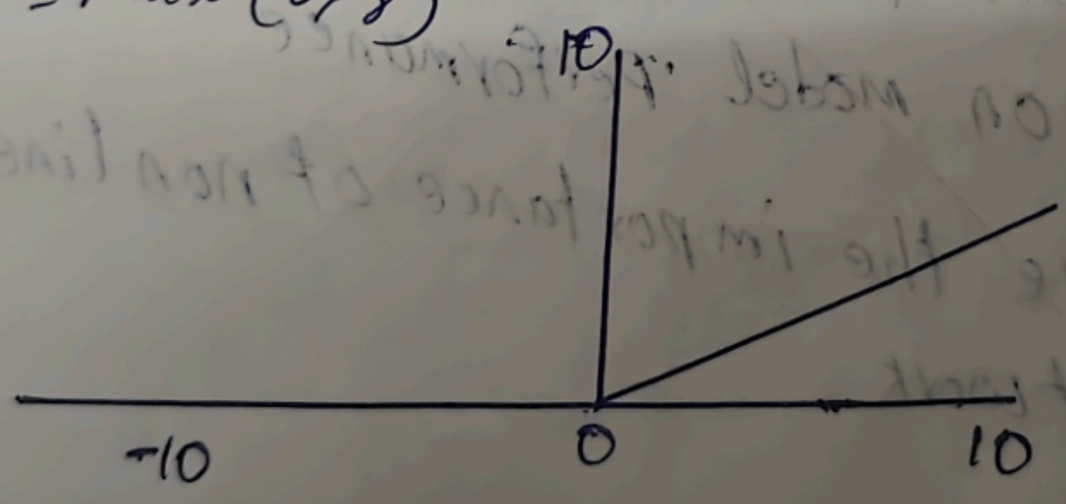
tanh:-

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



Relu:-

$$g(z) = \max(0, z)$$



Result:-

successfully implemented Activation functions and its different roles [Sigmoid; Tanh, Relu]