Exp: 5

22.08.25

STUDY OF ACTIVATION FUNCTION AND ITS ROLE.

Aim: To study the activation function and its role.

Pseudo code:

1. Import necessary libraries.
   - (numpy, tensorflow/keras, matplotlib)

2. Define different activation functions.
   ↳ - Sigmoid
   - Tanh
   - ReLU
   - Leaky ReLU
   - Softmax.

3. Visualize each function & its derivatives over input range.
   - range of input values ($x = -10$ to $+10$)
   - plot graphs to analyze shape & behaviour.

4. Build a simple NN
   ↳ MNIST classification.

5. Train the network multiple times, each w/ different activation.
   - Use same dataset, optimizer, learning rate, & epochs.
   - Record training loss and accuracy.

6. Compare performances.
   - plot accuracy v/s activation func.
   - plot loss v/s activation func.

7. compare results & conclude the role of the activation function.

SMALL NOTE: (Reference purpose).

Sigmoid = output 0 to 1 ; formula: $f(x) = \frac{1}{1+e^{-x}}$

↳ Binary solve.

Tanh = output -1 to 1 ; $\tanh = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$

↳ Stronger than ~~[illegible]~~ gradient.

Softmax = converts outputs to probabilities. ; $\frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$

↳ Solves multi-class problems.

ReLU = most popular output → 0

↳ if -ve

But if +ve same value ; $f(x) = \max(0, x)$

fast & effective.

Observation:

Sample Activation Outputs:

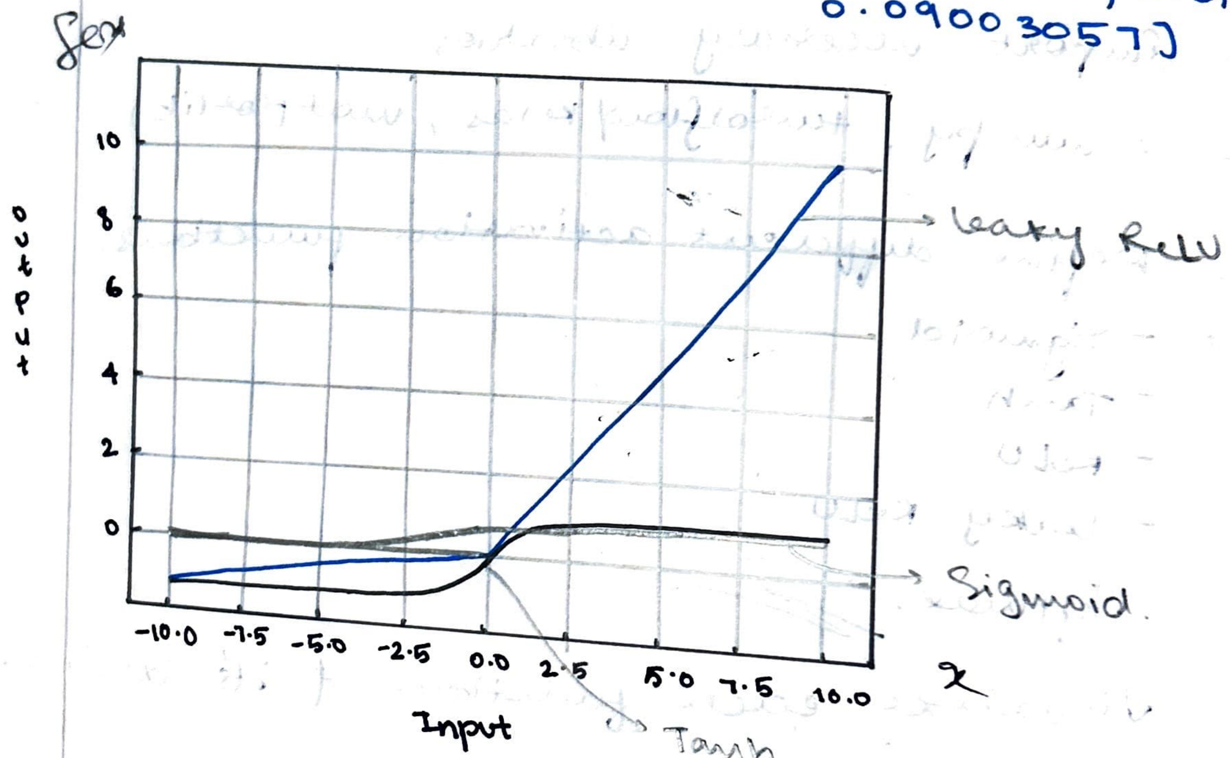
Sigmoid(2.0) = 0.8807971

Tanh(2.0) = 0.9640276

ReLU(-3.0) = 0.0

Leaky ReLU(-3.0) = -0.3

Softmax [[2.0, 1.0, 0.0]] = [0.66524094, 0.24472847, 0.09003057]]



Not secure | 10.1.38.19/user/ra2311047010012/lab/tree/Foundation%20of%20AI/SEM%205%20DLT%20LAB/week5.py

File Edit View Run Kernel Tabs Settings Help

Week2.py | Week3.py | Untitled.ipynb | week5.py | jupyter-ra23110 | week4.py | jupyter-ra23110

Filter files by name

/ Foundation of AI / SEM 5 DLT LAB /

Name	Last Modified
mnist_ffnn_...	33 minutes ago
Untitled.ipy...	51 minutes ago
Week2.py	24 days ago
Week3.py	17 days ago
week4.py	22 minutes ago
Week4.py	58 minutes ago
week5.py	1 minute ago

```python
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
x = np.linspace(-10, 10, 400, dtype=np.float32)
y_sigmoid = tf.keras.activations.sigmoid(x).numpy()
y_tanh = tf.keras.activations.tanh(x).numpy()
y_relu = tf.keras.activations.relu(x).numpy()
y_leaky = tf.nn.leaky_relu(x, alpha=0.1).numpy()
x_multi = np.vstack([x, x*0.5, x*0.2]).astype(np.float32)
y_softmax = tf.nn.softmax(x_multi, axis=0).numpy()
plt.figure(figsize=(10, 6))
plt.plot(x, y_sigmoid, label="Sigmoid")
plt.plot(x, y_tanh, label="Tanh")
plt.plot(x, y_relu, label="ReLU")
plt.plot(x, y_leaky, label="Leaky ReLU")
plt.title("Activation Functions")
plt.xlabel("Input")
plt.ylabel("Output")
plt.grid(True)
plt.legend()
plt.show()
plt.figure(figsize=(10, 6))
plt.plot(x, y_softmax[0], label="Class 1")
plt.plot(x, y_softmax[1], label="Class 2")
plt.plot(x, y_softmax[2], label="Class 3")
plt.title("Softmax Activation (Multiple Classes)")
plt.xlabel("Input")
plt.ylabel("Probability")
plt.grid(True)
plt.legend()
```

Simple 3 8 Python Mem: 1.76 GB Ln 37, Col 72 Spaces: 4 week5.py 1

Type here to search 09:31 01-09-2025

Not secure 10.1.38.19/user/ra2311047010012/lab/tree/Foundation%20of%20AI/SEM%205%20DLT%20LAB/week5.py

File Edit View Run Kernel Tabs Settings Help

Week2.py | Week3.py | Untitled.ipynb | week5.py | jupyter-ra23110 | week4.py | jupyter-ra23110

Filter files by name

/ Foundation of AI / SEM 5 DLT LAB /

Name	Last Modified
mnist_ffnn_...	34 minutes ago
Untitled.ipy...	52 minutes ago
Week2.py	24 days ago
Week3.py	17 days ago
week4.py	23 minutes ago
Week4.py	59 minutes ago
week5.py	1 minute ago

```python
y_leaky = tf.nn.leaky_relu(x, alpha=0.1).numpy()
x_multi = np.vstack([x, x*0.5, x*0.2]).astype(np.float32)
y_softmax = tf.nn.softmax(x_multi, axis=0).numpy()
plt.figure(figsize=(10, 6))
plt.plot(x, y_sigmoid, label="Sigmoid")
plt.plot(x, y_tanh, label="Tanh")
plt.plot(x, y_relu, label="ReLU")
plt.plot(x, y_leaky, label="Leaky ReLU")
plt.title("Activation Functions")
plt.xlabel("Input")
plt.ylabel("Output")
plt.grid(True)
plt.legend()
plt.show()
plt.figure(figsize=(10, 6))
plt.plot(x, y_softmax[0], label="Class 1")
plt.plot(x, y_softmax[1], label="Class 2")
plt.plot(x, y_softmax[2], label="Class 3")
plt.title("Softmax Activation (Multiple Classes)")
plt.xlabel("Input")
plt.ylabel("Probability")
plt.grid(True)
plt.legend()
plt.show()
print("Sample Activation Outputs:")
print("Sigmoid(2.0) =", tf.keras.activations.sigmoid(2.0).numpy())
print("Tanh(2.0) =", tf.keras.activations.tanh(2.0).numpy())
print("ReLU(-3.0) =", tf.keras.activations.relu(-3.0).numpy())
print("Leaky ReLU(-3.0) =", tf.nn.leaky_relu(-3.0, alpha=0.1).numpy())
print("Softmax([2.0,1.0,0.0]) =", tf.nn.softmax([2.0,1.0,0.0]).numpy())
```

Simple 3 8 Python Mem: 1.76 GB Ln 37, Col 72 Spaces: 4 week5.py 1

Type here to search 09:31 01-09-2025

```
_COMPLEX64, DT_COMPLEX128]> [Op:Sigmoid] name:
jupyter-ra2311047010012@cintel:~/Foundation of AI/SEM 5 DLT LAB$ python week5.py
2025-09-01 09:28:07.638684: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU inst
ructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
Sample Activation Outputs:
Sigmoid(2.0) = 0.8807971
Tanh(2.0) = 0.9640276
ReLU(-3.0) = 0.0
Leaky ReLU(-3.0) = -0.3
Softmax([2.0,1.0,0.0]) = [0.66524094 0.24472848 0.09003057]
jupyter-ra2311047010012@cintel:~/Foundation of AI/SEM 5 DLT LAB$
```

GB jupyter-ra2311047010012@cintel: ~/Foundation of AI/SEM 5 DLT LAB 1

09:32
01-09-2025