: RA2311047010016

S.No.	Date	Title	Signature
1	07.08.2025	Exploring the Deep learning platforms	
2.	07.08.2025	Implement a classifier using open source dataset	7/8/25
3.	14.08.2025	Study of classifiers with respect to Statistical parameters	14/8/25
4.	22.08.2025	To build and train a simple feed forward Network (~~FFNN~~) on MNIST dataset.	23/8/25
5.	22.8.25	Study different activation function used in NN	6/9/25
6.	9.9.25	Implement Gradient Descent & Backpropagation in ~~ANN~~ NN.	
7.	16.09.2025	Build a CNN model to classify Cat and dog image	23/9/25

22.08.2025.

Lab-5:

Aim: To study different activation functions used in neural networks.

objective:

→ To explore commonly used activation functions.

→ To analyse their mathematical behaviour and impact on learning.

→ To understand the importance of non-linearity in deep neural networks.

pseudocode:

→ Define different activation functions: sigmoid, tanh, ReLU, Leaky ReLU, Softmax.

→ Replace activation layers with the current function.

→ Train the model on a dataset (e.g., MNIST)

→ Record training loss and test accuracy.

→ Compare results across activation functions.

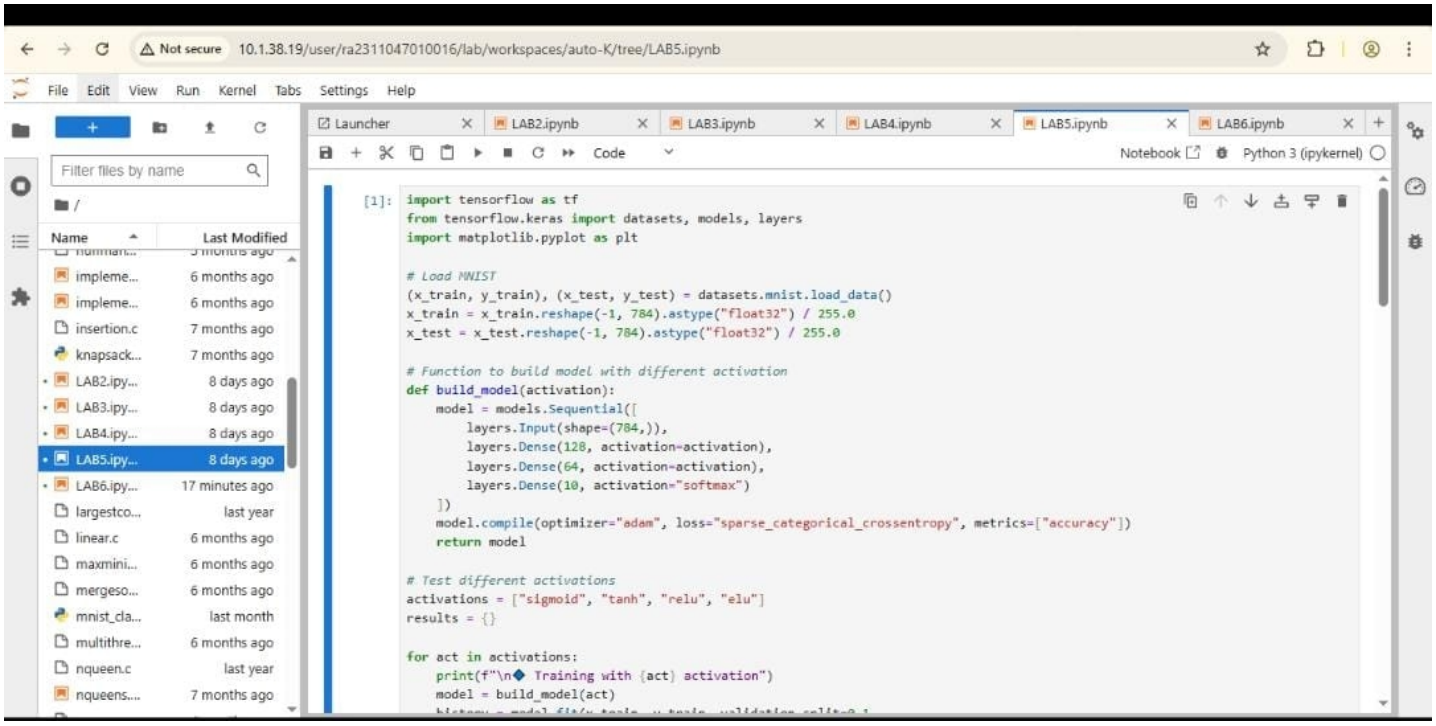
Observation:

→ Sigmoid: Can cause vanishing gradients; Slow training.

→ Tanh: Zero-centered; better than sigmoid but still prone to vanishing gradients.

→ ReLU: Fast training, mitigates vanishing gradient; may cause "dead neurons".

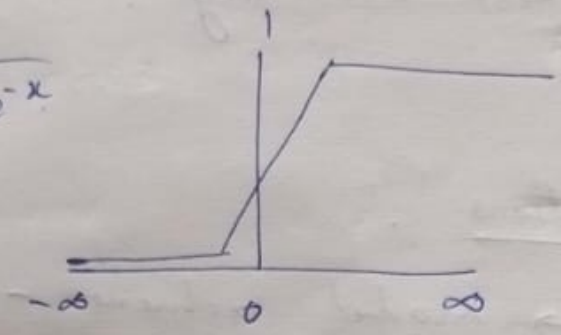
→ Leaky ReLU: fixes dead neuron problem by allowing small gradient when inactive


Not secure
10.1.38.19/user/ra2311047010016/lab/workspaces/auto-K/tree/LAB5.ipynb
File Edit View Run Kernel Tabs Settings Help
Launcher
LAB2.ipynb
LAB3.ipynb
LAB4.ipynb
LAB5.ipynb
LAB6.ipynb
Code
Notebook
Python 3 (ipykernel)
Filter files by name
Name
Last Modified
impleme... 6 months ago
impleme... 6 months ago
insertion.c 7 months ago
knapsack... 7 months ago
LAB2.ipy... 8 days ago
LAB3.ipy... 8 days ago
LAB4.ipy... 8 days ago
LAB5.ipy... 8 days ago
LAB6.ipy... 17 minutes ago
largestco... last year
linear.c 6 months ago
maxmini... 6 months ago
mergeso... 6 months ago
mnist_cla... last month
multithre... 6 months ago
nqueen.c last year
nqueens.... 7 months ago
[1]:
import tensorflow as tf
from tensorflow.keras import datasets, models, layers
import matplotlib.pyplot as plt
# Load MNIST
(x_train, y_train), (x_test, y_test) = datasets.mnist.load_data()
x_train = x_train.reshape(-1, 784).astype("float32") / 255.0
x_test = x_test.reshape(-1, 784).astype("float32") / 255.0
# Function to build model with different activation
def build_model(activation):
model = models.Sequential([
layers.Input(shape=(784,)),
layers.Dense(128, activation=activation),
layers.Dense(64, activation=activation),
layers.Dense(10, activation="softmax")
])
model.compile(optimizer="adam", loss="sparse_categorical_crossentropy", metrics=["accuracy"])
return model
# Test different activations
activations = ["sigmoid", "tanh", "relu", "elu"]
results = {}
for act in activations:
print(f"\n Training with {act} activation")
model = build_model(act)

x	Sigmoid	tanh	ReLU	LiR
5	0.9933	0.999	5	5
7	0.999	0.999	7	7
-1	0.268	-0.761	0	0

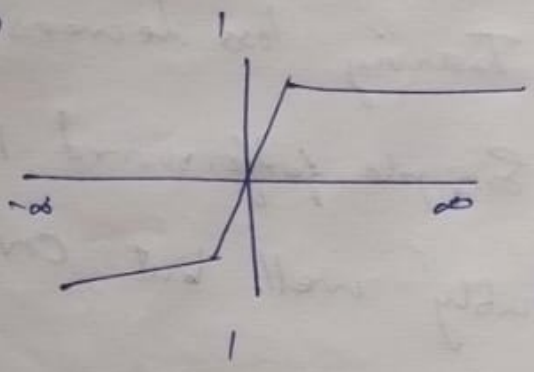
1.) Sigmoid

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



2. tanh

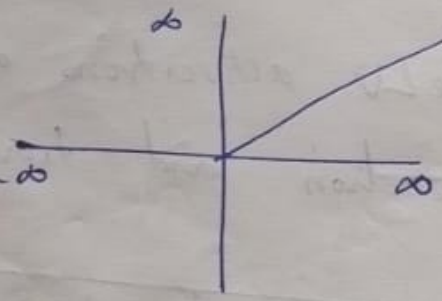
tanh



5. Maxout

$\max(w_1^T x + b_1, w_2^T x$

3.) ReLu

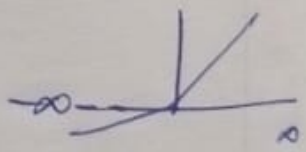


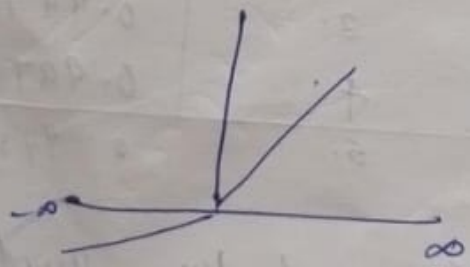
6. ELU

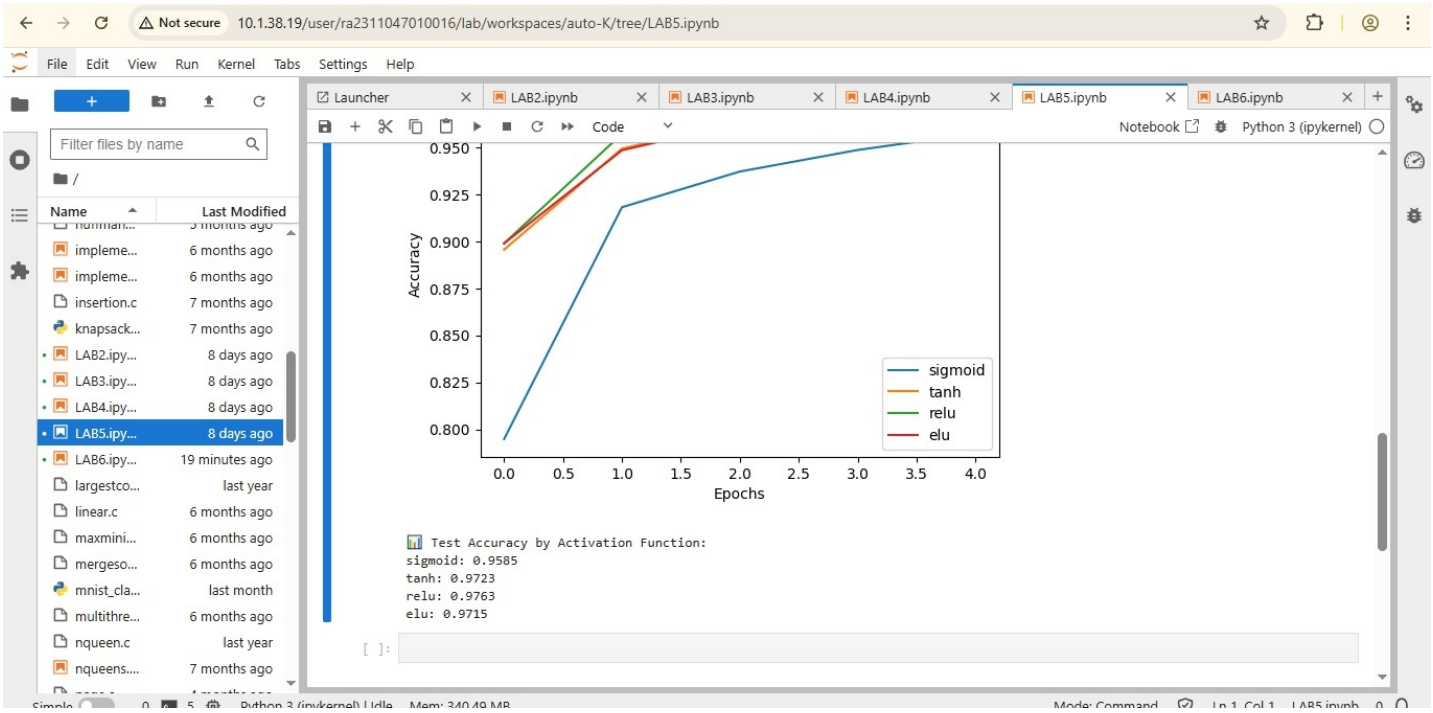
$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$

4.) leaky ReLu

$\max(0, 1x, 2)$



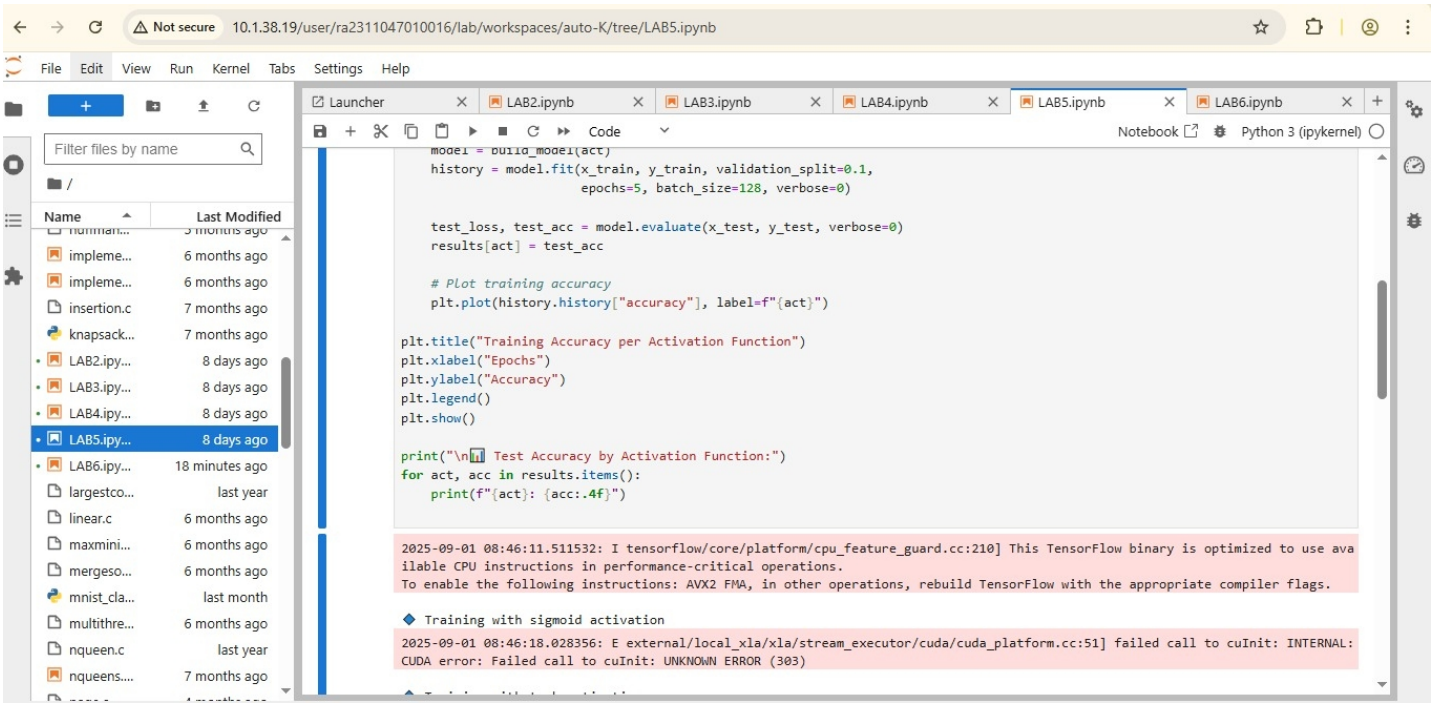



Not secure 10.1.38.19/user/ra2311047010016/lab/workspaces/auto-K/tree/LAB5.ipynb
File Edit View Run Kernel Tabs Settings Help
Launcher
LAB2.ipynb
LAB3.ipynb
LAB4.ipynb
LAB5.ipynb
LAB6.ipynb
Code
Notebook
Python 3 (ipykernel)
Filter files by name
Name
Last Modified
impleme... 6 months ago
impleme... 6 months ago
insertion.c 7 months ago
knapsack... 7 months ago
LAB2.ipy... 8 days ago
LAB3.ipy... 8 days ago
LAB4.ipy... 8 days ago
LAB5.ipy... 8 days ago
LAB6.ipy... 19 minutes ago
largestco... last year
linear.c 6 months ago
maxmini... 6 months ago
mergeso... 6 months ago
mnist_cla... last month
multithre... 6 months ago
nqueen.c last year
nqueens.... 7 months ago
Accuracy
Epochs
sigmoid
tanh
relu
elu
Test Accuracy by Activation Function:
sigmoid: 0.9585
tanh: 0.9723
relu: 0.9763
elu: 0.9715

→ Activation choice significantly affects convergents speed and final accuracy.

Result:

Successfully completed the different activation functions used in neural networks.


Not secure
10.1.38.19/user/ra2311047010016/lab/workspaces/auto-K/tree/LAB5.ipynb
File Edit View Run Kernel Tabs Settings Help
Launcher
LAB2.ipynb
LAB3.ipynb
LAB4.ipynb
LAB5.ipynb
LAB6.ipynb
Code
Notebook
Python 3 (ipykernel)
Filter files by name
Name
Last Modified
impleme... 6 months ago
impleme... 6 months ago
insertion.c 7 months ago
knapsack... 7 months ago
LAB2.ipy... 8 days ago
LAB3.ipy... 8 days ago
LAB4.ipy... 8 days ago
LAB5.ipy... 8 days ago
LAB6.ipy... 18 minutes ago
largestco... last year
linear.c 6 months ago
maxmini... 6 months ago
mergeso... 6 months ago
mnist_cla... last month
multithre... 6 months ago
nqueen.c last year
nqueens.... 7 months ago
model = build_model(act)
history = model.fit(x_train, y_train, validation_split=0.1,
epochs=5, batch_size=128, verbose=0)
test_loss, test_acc = model.evaluate(x_test, y_test, verbose=0)
results[act] = test_acc
# Plot training accuracy
plt.plot(history.history["accuracy"], label=f"{act}")
plt.title("Training Accuracy per Activation Function")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()
print("\n Test Accuracy by Activation Function:")
for act, acc in results.items():
print(f"{act}: {acc:.4f}")
2025-09-01 08:46:11.511532: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
Training with sigmoid activation
2025-09-01 08:46:18.028356: E external/local_xla/xla/stream_executor/cuda/cuda_platform.cc:51] failed call to cuInit: INTERNAL: CUDA error: Failed call to cuInit: UNKNOWN ERROR (303)