**Loads the MNIST dataset** (handwritten digits 0-9).  
✅ **Preprocesses the data** (normalization).  
✅ **Builds an ANN with 1 hidden layer (128 neurons, ReLU activation)**.  
✅ **Compiles using Adam optimizer & categorical cross-entropy loss**.  
✅ **Trains for 5 epochs on the training dataset**.  
✅ **Evaluates performance on the test dataset**.  
✅ **Makes predictions and visualizes one sample**.

# Step 1: Install TensorFlow (Uncomment if not installed)

# !pip install tensorflow

# Step 2: Import required libraries

import tensorflow as tf

from tensorflow import keras

import numpy as np

import matplotlib.pyplot as plt

# Step 3: Load the MNIST dataset (Handwritten Digits)

(X\_train, y\_train), (X\_test, y\_test) = keras.datasets.mnist.load\_data()

# Step 4: Normalize the data (scale pixel values between 0 and 1)

X\_train, X\_test = X\_train / 255.0, X\_test / 255.0

# Step 5: Build a Simple ANN Model

model = keras.Sequential([

keras.layers.Flatten(input\_shape=(28, 28)), # Input Layer (Flatten 28x28 images)

keras.layers.Dense(128, activation='relu'), # Hidden Layer with 128 neurons & ReLU

keras.layers.Dense(10, activation='softmax') # Output Layer (10 classes, Softmax)

])

# Step 6: Compile the Model

model.compile(optimizer='adam',

loss='sparse\_categorical\_crossentropy',

metrics=['accuracy'])

# Step 7: Train the Model

model.fit(X\_train, y\_train, epochs=5, validation\_data=(X\_test, y\_test))

# Step 8: Test the Model

test\_loss, test\_acc = model.evaluate(X\_test, y\_test)

print(f"Test Accuracy: {test\_acc:.4f}")

# Step 9: Make Predictions

predictions = model.predict(X\_test)

# Step 10: Display a Sample Prediction

plt.imshow(X\_test[0], cmap="gray")

plt.title(f"Predicted: {np.argmax(predictions[0])}, Actual: {y\_test[0]}")

plt.show()