# Step 1: Import necessary libraries

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Flatten

from tensorflow.keras.datasets import mnist

from tensorflow.keras.utils import to\_categorical

# Step 2: Load and preprocess the dataset (MNIST)

(x\_train, y\_train), (x\_test, y\_test) = mnist.load\_data()

x\_train, x\_test = x\_train / 255.0, x\_test / 255.0 # Normalize pixel values

y\_train, y\_test = to\_categorical(y\_train), to\_categorical(y\_test) # One-hot encode labels

# Step 3: Build the feedforward neural network model

model = Sequential([

Flatten(input\_shape=(28, 28)), # Input layer (flatten 28x28 images)

Dense(128, activation='relu'), # Hidden layer with 128 neurons

Dense(64, activation='relu'), # Another hidden layer

Dense(10, activation='softmax') # Output layer (10 classes)

])

# Step 4: Compile the model

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

# Step 5: Train the model

model.fit(x\_train, y\_train, epochs=5, batch\_size=32)

# Step 6: Evaluate the model

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

print("Test Accuracy:", test\_acc)

# Step 7: Make a prediction (optional example)

import numpy as np

pred = model.predict(np.expand\_dims(x\_test[0], axis=0))

print("Predicted class:", np.argmax(pred))

Epoch 1/5

1875/1875 [==============================] - 5s 3ms/step - loss: 0.2623 - accuracy: 0.9235

Epoch 2/5

1875/1875 [==============================] - 5s 3ms/step - loss: 0.1144 - accuracy: 0.9657

Epoch 3/5

1875/1875 [==============================] - 5s 3ms/step - loss: 0.0804 - accuracy: 0.9751

Epoch 4/5

1875/1875 [==============================] - 5s 3ms/step - loss: 0.0600 - accuracy: 0.9813

Epoch 5/5

1875/1875 [==============================] - 5s 3ms/step - loss: 0.0483 - accuracy: 0.9854

313/313 [==============================] - 1s 2ms/step - loss: 0.0843 - accuracy: 0.9766

Test Accuracy: 0.9766001100540161

Predicted class: 7