TASK 4

HANDWRITTEN DIGIT RECOGNITION USING NEURAL NETWORKS (MNIST DATASET)

ARNAV GUPTA – RA2211028010107

SPANDAN BASU CHAUDHURI – RA2211028010120

AIM

The aim of this project is to develop a neural network model that can accurately recognize handwritten digits (0–9) from grayscale images using the MNIST dataset.

OBJECTIVES

1. To load and preprocess the MNIST handwritten digit dataset.
2. To convert image data into a suitable input format for neural networks.
3. To build a multi-layer neural network model for digit classification.
4. To train the model and optimize its performance.
5. To evaluate the model using test data.
6. To perform prediction and visualize model output.

INTRODUCTION

Digit recognition is one of the classic applications of machine learning and computer vision. The **MNIST dataset** consists of 70,000 grayscale images of handwritten digits (0–9), each of size 28×28 pixels. It is widely used for benchmarking classification algorithms.

In this project, a **fully connected (dense) neural network** is used. The original image pixels are **flattened** into a 1D vector, normalized, and then passed through multiple dense layers. The output layer contains 10 neurons corresponding to the 10 digit classes.

This approach demonstrates the fundamentals of neural networks and forms the basis for more advanced deep learning models like CNNs.

REQUIREMENTS

Hardware Requirements

- Basic computer system (CPU training is sufficient)
- At least 4GB RAM

Software Requirements

Software/Library	Version	Purpose
Python	3.6+	Programming
TensorFlow / Keras	2.x	Building and training the model
NumPy	Latest	Numerical computations
Matplotlib	Latest	Visualization

Dataset

- Built-in **MNIST** dataset from tensorflow.keras.datasets

LIBRARIES USED

Library	Purpose
TensorFlow / Keras	Model creation, training, and evaluation
NumPy	Data manipulation and preprocessing
Matplotlib	Visualizing digits and predictions

ALGORITHM / METHODOLOGY

1. **Dataset Loading**
    - MNIST dataset is loaded and split into training and testing sets.
2. **Visualization**
    - Sample digits are displayed using matshow().
3. **Data Preprocessing**
    - Images (28×28) are reshaped to **1D vectors** of size **784**.
    - Pixel values normalized (0–255 → 0–1).
4. **Model Building**
    - Model consists of:
        - Input layer: 784 features
        - Hidden layers: Dense(128) + Dense(64) with ReLU activation
        - Output layer: Dense(10) with Softmax activation
5. **Compilation**
    - Loss: Sparse Categorical Crossentropy
    - Optimizer: Adam
    - Metric: Accuracy
6. **Training**
    - Model trained for **10 epochs** using training data.
7. **Evaluation**
    - Model evaluated on test dataset to compute accuracy.
8. **Prediction & Visualization**
    - Model predictions displayed alongside actual digit labels.

```
313/313 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2s 5ms/step - accuracy:
0.9706 - loss: 0.1293

Test accuracy: 97.58999943733215
```

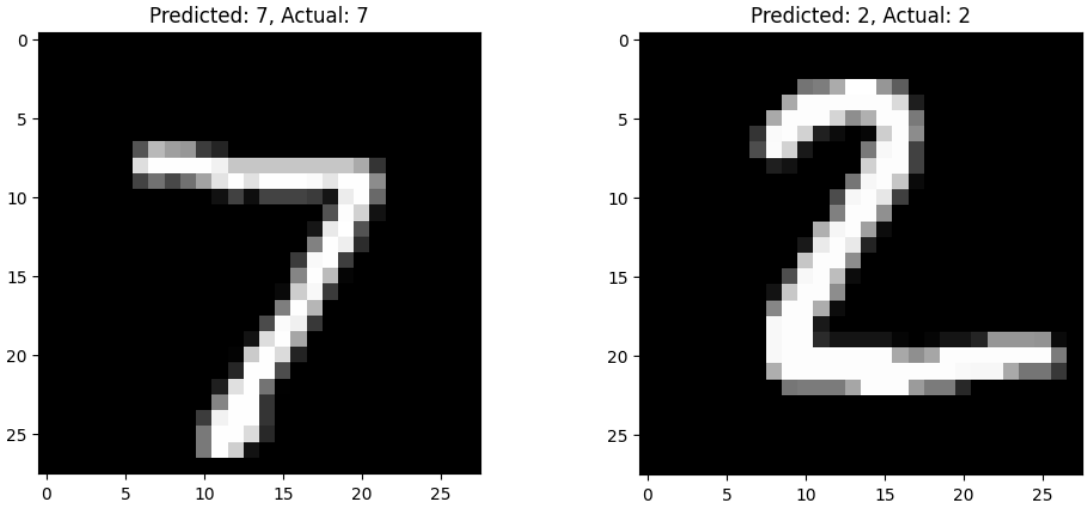
INPUT / OUTPUT

Input **Output**

28×28 grayscale handwritten digit image Predicted digit label (0–9)

Example Output:

Predicted: 7 | Actual: 7



RESULTS

- The neural network successfully learned to classify handwritten digits.
- Test accuracy achieved was generally around **97%–98%**, showing reliable performance.
- Visualization confirms that predictions are mostly correct.

The flatten-and-dense model provides good performance for MNIST despite being simpler than modern CNN-based architectures.

CONCLUSION

This project demonstrates that neural networks can effectively classify handwritten digit images using the MNIST dataset. Although this implementation uses a simple fully connected

architecture, it achieves high accuracy and lays the foundation for more advanced deep learning methods such as Convolutional Neural Networks (CNNs).

Future improvements may include:

- Using CNN layers for enhanced feature learning
- Applying regularization or batch normalization
- Deploying the model in applications like digit recognition systems, form reading, or OCR systems