**Assignment 1**

**1. Introduction**

In this assignment, we implement a **deep learning model** to classify handwritten digits from the **MNIST dataset** using **TensorFlow and Keras**. The MNIST dataset contains **28×28 grayscale images** of digits ranging from 0 to 9. Our task is to train a neural network that can accurately identify these digits.

This problem demonstrates the practical application of **neural networks** in computer vision and serves as a foundational project for understanding deep learning concepts.

**2. Dataset**

* **Dataset Used**: MNIST (Modified National Institute of Standards and Technology)
* **Training Set**: 60,000 images
* **Test Set**: 10,000 images
* **Image Size**: 28×28 pixels, grayscale
* **Classes**: 10 (digits 0–9)

Before training, the dataset is normalized by dividing all pixel values by 255.0 to bring them into the range **[0, 1]**, which improves the efficiency of the learning process.

**3. Model Architecture**

We use a **Sequential Neural Network** consisting of the following layers:

1. **Flatten Layer**: Converts the 2D 28×28 pixel matrix into a 1D vector of size 784.
2. **Dense Layer (128 units, ReLU activation)**: Fully connected layer that learns hidden representations of the input.
3. **Dropout Layer (0.2)**: Randomly sets 20% of neurons to zero during training to prevent **overfitting**.
4. **Dense Output Layer (10 units, Softmax activation)**: Produces probability distribution over 10 classes (digits 0–9).

**4. Compilation**

The model is compiled using:

* **Optimizer**: Adam → an adaptive optimizer for efficient gradient descent.
* **Loss Function**: SparseCategoricalCrossentropy → suitable for multi-class classification with integer labels.
* **Metrics**: Accuracy → evaluates performance of the model.

**5. Training**

The model is trained on the training dataset for **5 epochs**. Each epoch represents one full pass through the entire dataset. The training process adjusts model weights to minimize the loss function.

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

**6. Evaluation**

The trained model is tested on unseen test data using:

model.evaluate(x\_test, y\_test)

* **Evaluation Metric**: Accuracy
* On MNIST, the model typically achieves an accuracy of around **97–98%** with this architecture.

**7. Results & Discussion**

* The model successfully classifies digits with high accuracy, demonstrating that even a simple feedforward neural network is effective for image classification tasks.
* The inclusion of **Dropout** improves generalization by preventing overfitting.
* The model could be further improved using **Convolutional Neural Networks (CNNs)**, which are specifically designed for image data.

**8. Conclusion**

This assignment demonstrated the implementation of a neural network using **TensorFlow/Keras** for digit recognition on the **MNIST dataset**. The results confirm that deep learning models can achieve high accuracy on image classification tasks with minimal preprocessing.