# Recognition of Handwritten Digits by MLP

## Introduction

The objective of this project is to implement a multi-layer perceptron (MLP) for the classification of handwritten digits from the widely recognized MNIST dataset. The MNIST dataset, which consists of 70,000 grayscale images (60,000 for training and 10,000 for testing) of handwritten digits (0-9), serves as a standard benchmark for evaluating machine learning models in the domain of image classification.  
  
The choice of an MLP for this project is driven by its simplicity and effectiveness as a baseline model. Although more complex models such as convolutional neural networks (CNNs) typically achieve higher accuracies on image-based tasks, the MLP provides a straightforward framework for understanding key concepts such as data preprocessing, model architecture, training dynamics, and performance evaluation.

## Data Preprocessing and Methodology

Data preprocessing is a critical step in ensuring that the raw MNIST images are formatted appropriately for input into the neural network. The following preprocessing steps were implemented:  
  
- **Data Loading:**  The MNIST dataset is loaded directly from the TensorFlow/Keras library.  
- **Normalization:** Pixel values, originally ranging from 0 to 255, are scaled to the [0, 1] range. This normalization facilitates more stable and faster training.  
- **Reshaping:** Each 28×28 image is flattened into a 784-dimensional vector, which is necessary for feeding into the dense layers of the MLP.  
- **One-Hot Encoding:** Class labels (digits 0–9) are converted into one-hot encoded vectors, a common practice in multi-class classification problems.  
  
This preprocessing pipeline ensures that the input data is consistent and well-suited for the learning process, reducing the risk of issues such as vanishing gradients and numerical instabilities.

## Model Architecture and Training

The core of this project is the implementation of an MLP model using TensorFlow/Keras. The model architecture is defined as follows:  
  
**Input Layer:** Accepts a 784-dimensional vector corresponding to the flattened input image.  
**Hidden Layers:**   
 - The first hidden layer consists of 512 neurons with the ReLU activation function, which introduces non-linearity to help the network learn complex patterns.  
 - The second hidden layer has 256 neurons, also using the ReLU activation function.  
**Output Layer:** A softmax-activated layer with 10 neurons is used to output the probability distribution over the 10 classes (digits 0–9).  
  
The model is compiled using the Adam optimizer and categorical crossentropy as the loss function. The choice of the Adam optimizer is justified by its adaptive learning rate capabilities, which enhance convergence during training.  
  
Training is carried out over 10 epochs with a batch size of 128. A validation split of 20% of the training data is employed to monitor the model's performance and help detect any signs of overfitting. During the training process, both accuracy and loss metrics are recorded for subsequent visualization.

## Results and Discussion

Upon completion of training, the model is evaluated on the test set. The evaluation yields a test accuracy that serves as an indicator of the model’s generalization performance. In addition to numerical evaluation, several visualizations are generated:  
  
**Training History Plot:** This graph depicts the evolution of training and validation accuracy and loss over the epochs. It provides insights into the learning dynamics and whether the model is overfitting or underfitting.  
**Prediction Visualizations:** A subset of test images is displayed alongside their predicted and true labels, offering a qualitative assessment of model performance.  
  
The observed test accuracy reflects the model’s capability to correctly classify handwritten digits, although there is room for improvement. The performance of this MLP can be further enhanced by:  
  
**Tuning Hyperparameters**: Adjusting the learning rate, batch size, and number of epochs.  
**Model Architecture Enhancements:** Experimenting with different numbers of layers and neurons, or integrating dropout layers to mitigate overfitting.  
**Alternative Models:** Considering more advanced architectures like convolutional neural networks (CNNs), which are better suited for image classification tasks.

## Conclusion and Future Work

In summary, this project successfully demonstrates the implementation of an MLP for handwritten digit recognition using the MNIST dataset. The project encompasses key stages of a machine learning pipeline: data preprocessing, model building, training, evaluation, and visualization. While the MLP provides a strong foundation and yields promising results, several avenues exist for future improvement.  
  
Future work may include:  
  
**Advanced Architectures:** Transitioning from an MLP to a CNN for better performance.  
**Hyperparameter Optimization:** Employing techniques such as grid search or Bayesian optimization to systematically explore the hyperparameter space.  
**Data Augmentation:** Integrating augmentation techniques to artificially expand the dataset and improve model robustness.  
**Regularization Techniques:** Incorporating L2 regularization to further reduce the risk of overfitting.  
  
Contributions

**Avneesh :** Finalizing the project and developing the readme and report.

**Srijan :**

**Rajarshi :**

**Om :**