

## Script Documentation: MNIST Digit Classifier using TensorFlow and Keras

### Overview

This Python script implements a simple neural network using TensorFlow and Keras to classify handwritten digits from the MNIST dataset. The script includes data preprocessing, model definition, training, evaluation, and visualization of performance metrics.

### Dependencies

Install required packages:

***pip install tensorflow matplotlib***

### Dataset

The script uses the MNIST dataset, which contains 60,000 training images and 10,000 test images of handwritten digits (0 through 9). Each image is 28x28 pixels.

### Script Workflow

1. Load the MNIST dataset using TensorFlow/Keras.
2. Normalize the image pixel values to the range 0–1.
3. Define a neural network model with the following layers:
  - Flatten: Converts 28x28 image into a 1D vector.
  - Dense (128 neurons): Hidden layer with ReLU activation.
  - Dense (10 neurons): Output layer with softmax activation.
4. Compile the model with Adam optimizer and sparse categorical crossentropy loss function.
5. Train the model for 5 epochs with 20% of training data used for validation.
6. Evaluate the model on test data and print the test accuracy.
7. Plot training/validation accuracy and loss graphs.

### Model Architecture

- Input Layer: Flatten layer with input\_shape=(28, 28)
- Hidden Layer: Dense layer with 128 neurons and 'relu' activation
- Output Layer: Dense layer with 10 neurons and 'softmax' activation

### Visualization

The script generates two plots after training:

- Accuracy over epochs (training vs validation)

- Loss over epochs (training vs validation)

## **Model Evaluation**

After training, the model is evaluated on the test set using `model.evaluate()`. The script prints the test accuracy.

## **How to Run**

Ensure TensorFlow and matplotlib are installed, then run the script using Python:

***python mnist\_classifier.py***

## **Notes**

- Ensure you have a stable internet connection when loading the MNIST dataset for the first time.
- The training might take longer depending on your hardware (CPU vs GPU).