Task-1: Neural Network with MNIST Dataset -

Comprehensive Documentation

This documentation provides a detailed overview of the Python script "Creating a neural network with MNIST.ipynb," which focuses on building and training a Convolutional Neural Network (CNN) using TensorFlow for image classification on the MNIST dataset. The documentation is structured to cover various aspects, including installation, usage, and a breakdown of each task within the code.

1. **Introduction:**
   - The script is designed to showcase the creation and training of a CNN for image classification using the popular MNIST dataset.
   - Tasks include data import, exploration, preprocessing, model building, training, evaluation, and performance analysis.

2. **Installation:**
   - The script is written in Python and requires certain libraries to be installed.
   - Ensure you have the required dependencies by installing TensorFlow, NumPy, Matplotlib, Pandas, and Seaborn.

3. **Usage:**
   - Download the script file ("Creating a neural network with MNIST.ipynb") from the provided link or clone the entire Jupyter Notebook from the Colab link.
   - Open the script using a Jupyter Notebook environment or a compatible Python IDE.
   - Run the script cell by cell to observe the step-by-step execution.
   - Note: Ensure that the required datasets and dependencies are available.

4. **Tasks Breakdown:**

Data Import and Exploration:
   - Loads the MNIST dataset and provides information about its structure.
   - Displays an example image from the dataset.

Data Preprocessing:
   - Reshapes and normalizes the input data.

Building the CNN Model:
   - Defines a CNN architecture using TensorFlow's Functional API.
   - Compiles the model with the specified optimizer, loss function, and metrics.

Model Training:

- Trains the CNN on the training data with specified epochs and batch size.
- Saves the best model weights using ModelCheckpoint.

Model Evaluation:

- Loads the trained model and evaluates its performance on the test set.

Model Performance Metrics:

- Calculates and visualizes various metrics, including accuracy, loss, precision, recall, and a confusion matrix.
- Plots training and validation loss and accuracy over epochs.

Visualizing Misclassified Images:

- Identifies and displays misclassified images from the test set.

Heatmap of Confusion Matrix:

- Generates a heatmap to visualize the confusion matrix.

5. Conclusion:

- The script provides a comprehensive tutorial on building and evaluating a CNN for image classification using TensorFlow on the MNIST dataset.
- Users can gain insights into model performance and visually assess misclassified images.

By following this documentation, users can effectively utilize the provided script for learning and experimenting with CNNs for image classification tasks on the MNIST dataset.