Detailed Analysis of Multilayer Perceptron

Introduction

Implement the following assignment in Python.

MLP Assignment for MNIST Dataset

1. **Read the MNIST Dataset**
   Load the MNIST dataset, which consists of handwritten digit images, into your program.

2. **Identify the Total Number of Class Labels**
   Determine the total number of unique class labels (digits) in the MNIST dataset.

3. **Sample k Images**
   Select a sample of k images ($k > 10$) per class label from the MNIST dataset, ensuring representation of all classes.

4. **Augment the Dataset to 500 Images**
   Apply various data augmentation techniques (e.g., rotation, scaling, flipping) to increase the dataset size to exactly 500 images.

5. **Design the MLP Architecture**
   Define the architecture of the Multi-Layer Perceptron (MLP) by specifying:

   - Number of hidden layers
   - Activation functions for each layer
   - Number of nodes in each hidden layer
   - Number of nodes in the output layer

6. **Choose a Weight Initialization Technique**
   Research various weight initialization techniques (e.g., Xavier, He, random initialization) and select an appropriate method for initializing the weights and biases of the MLP.

7. **Determine the Training and Testing Split**

Decide the proportion of the dataset to be used for training and testing (e.g., 80% training, 20% testing).

8. **Select Optimizers for Training**
   Identify a list of optimizers (e.g., SGD, Adam, RMSprop) to train the MLP.

9. **Select a Loss Function**
   Identify a list of loss functions suitable for the MNIST classification task (e.g., cross-entropy loss, mean squared error) and select an appropriate loss function for training the MLP.

10. **Define Termination Conditions for Training**
    Specify termination conditions for the training phase, including regularization techniques (e.g., L1, L2, dropout) and other criteria (e.g., number of epochs, early stopping).

11. **List Performance Metrics**
    Identify a set of performance metrics (e.g., accuracy, precision, recall, F1-score) to evaluate the MLP's performance.

12. **Calculate Total Number of Parameters**
    Compute and display the total number of parameters in the chosen MLP architecture.

13. **Training and Testing Phase**
    Train the MLP using the training dataset and evaluate its performance on the testing dataset.

14. **Prepare a Performance Metrics Table**
    Create a table summarizing the performance metrics for at least 10 different combinations of:

    - Weight initialization techniques
    - Optimizers
    - Loss functions
    - Regularization methods
    - Termination conditions

15. **Store Trained Models**
    Save all trained MLP models to disk for future use.

16. **Reload a Random Model**
    Randomly select and reload one of the saved trained models.

17. **Test on the Entire MNIST Dataset**
    Evaluate the reloaded model on the complete MNIST dataset and report its performance using the identified metrics to realize the importance of

sample size and quality.