Detailed Report on MLP-Based Network for 10-Class Classification

Part 1

Introduction
This report details the design, experimentation, and analysis of a Multi-Layer Perceptron (MLP) network for performing 10-class classification on the MNIST dataset. The goal was to achieve the highest classification performance by experimenting with different network architectures, training parameters, and interpretability techniques. The report also includes an analysis of misclassified examples and suggestions for improving the model.

Problem Statement
The task was to design a simple neural network for 10-class classification using the MNIST dataset. The key objectives were:
1. **Design and Experimentation**: Experiment with different numbers of layers, neurons, and training parameters to achieve the highest classification performance.
2. **Interpretability**: Analyze the features learned by the model, visualize intermediate representations, and interpret these representations.
3. **Misclassification Analysis**: Analyze misclassified examples to understand the model's shortcomings and suggest improvements.

Methodology
1. **Data Loading and Preprocessing**: The MNIST dataset was loaded and normalized. The dataset was split into training, validation, and test sets.
2. **Model Architecture**: An MLP model with configurable layers and neurons was implemented. The model included ReLU activation and dropout for regularization.
3. **Training**: The model was trained using the Adam optimizer and CrossEntropyLoss. Training parameters such as learning rate, number of epochs, and batch size were experimented with.
4. **Evaluation**: The model's performance was evaluated using accuracy, precision, recall, F1 score, and confusion matrix.
5. **Interpretability**: Techniques such as visualizing layer activations, feature importance, and weight distributions were used to interpret the model's learning.

Experiments and Results
Several experiments were conducted with different architectures:
- **Small MLP**: 2 hidden layers (128, 64 neurons)
- **Medium MLP**: 3 hidden layers (256, 128, 64 neurons)
- **Large MLP**: 4 hidden layers (512, 256, 128, 64 neurons)
- **Wide MLP**: 2 hidden layers (1024, 512 neurons)
- **Deep MLP**: 5 hidden layers (256, 256, 128, 128, 64 neurons)

The results of these experiments are summarized below:

Experiment	Test Accuracy	Test Precision	Test Recall	Test F1 Score
Small MLP	0.9780	0.9781	0.9780	0.9780
Medium MLP	0.9776	0.9777	0.9776	0.9776
Large MLP	0.9780	0.9781	0.9780	0.9780
Wide MLP	0.9790	0.9791	0.9790	0.9790
Deep MLP	0.9745	0.9747	0.9745	0.9745

The **Wide MLP** achieved the highest test accuracy of **0.9790**.

Interpretability and Analysis
1. **Feature Importance**: The feature importance map highlighted the regions of the input images that were most influential in the model's predictions. This helped in understanding which parts of the images the model focused on.
2. **Layer Activations**: Visualizing the activations of different layers provided insights into how the model transformed the input data through its layers. The activations for correctly classified and misclassified examples were compared to understand the differences in feature extraction.
3. **Weight Distributions**: The distribution of weights in each layer was plotted to analyze the learning patterns and ensure that the model was not overfitting.

Misclassification Analysis
The misclassified examples were analyzed to understand the model's shortcomings:
- **Common Patterns**: Misclassified examples often involved digits with similar shapes (e.g., 4 and 9, 2 and 7). This suggests that the model struggled with distinguishing between digits that have overlapping features.
- **Feature Extraction**: The model might not be capturing subtle differences in the curvature or orientation of certain digits, leading to misclassifications.

Conclusion and Recommendations
1. **Model Shortcomings:** The model's primary shortcoming is its difficulty in distinguishing between digits with similar shapes. This could be due to insufficient feature extraction or the need for more sophisticated architectures.
2. **Improvement Suggestions:**

- **Data Augmentation**: Introduce data augmentation techniques to increase the diversity of the training data, helping the model learn more robust features.
- **Advanced Architectures**: Experiment with more advanced architectures like Convolutional Neural Networks (CNNs) that are better suited for image data.
- **Class Weighting**: Adjust the class weights to give more importance to harder-to-classify examples, potentially improving the model's performance on misclassified digits.
- **Ensemble Methods**: Use ensemble methods to combine the predictions of multiple models, which can help in reducing misclassifications.

Final Thoughts

The Wide MLP model achieved a high accuracy of 97.9%, demonstrating the effectiveness of MLPs for the MNIST dataset. However, the analysis of misclassified examples revealed areas for improvement. By incorporating more sophisticated techniques and architectures, the model's performance can be further enhanced.