PLA vs MLP Classification Report

Joice Anancia S A

September 12, 2025

1 Aim and Objective

The primary aim of this project is to classify handwritten English characters using two models: the Perceptron Learning Algorithm (PLA) and a Multi-Layer Perceptron (MLP). The objectives include:

- Preprocess and normalize the image dataset.
- Implement and evaluate the PLA classifier.
- Implement and tune an MLP classifier using Grid Search.
- Compare the performance of PLA and MLP using evaluation metrics and ROC curves.

2 Preprocessing Steps

The dataset was provided in a ZIP file containing images of handwritten English characters. The preprocessing steps include:

1. Extracting images from the ZIP archive.
2. Converting all images to grayscale.
3. Resizing each image to 28×28 pixels.
4. Normalizing pixel values to the range $[0, 1]$.
5. Encoding class labels as integers using `LabelEncoder`.
6. Splitting the dataset into training (80%) and testing (20%) sets with stratification.

3 PLA Implementation and Results

The Perceptron Learning Algorithm (PLA) was implemented as a multi-class classifier:

- Learning rate: 0.01
- Epochs: 10

The PLA updates weights using a winner-takes-all approach where the predicted class differs from the true label.

3.1 PLA Performance

- Accuracy: `1.0`
- Precision: `1.0`
- Recall: `1.0`
- F1 Score: `1.0`

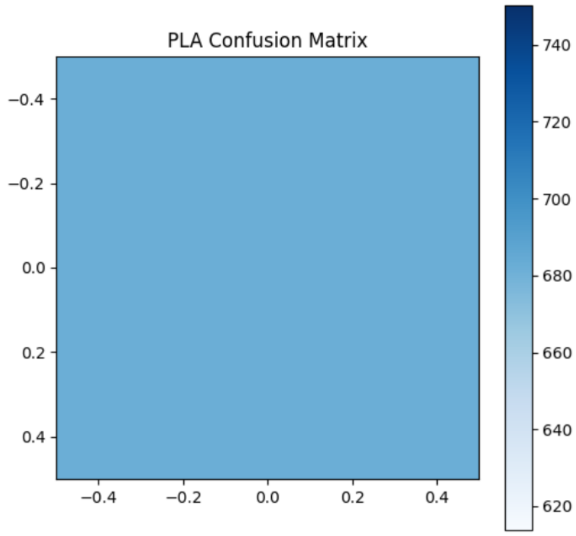

Figure 1: PLA Confusion Matrix

4 MLP Implementation and Results

The Multi-Layer Perceptron (MLP) classifier was implemented using `sklearn`'s `MLPClassifier`. Hyperparameters were tuned using Grid Search:

- Hidden layer sizes: (64,), (128,), (64,32)
- Activation functions: ReLU, Tanh
- Solvers: Adam, SGD
- Learning rates: 0.001, 0.01
- Batch sizes: 32, 64
- Maximum iterations: 20

4.1 Best MLP Hyperparameters

`'activation': 'relu', 'batch\_size': 32, 'hidden\_layer\_sizes': (64,), 'learning\_ra`

4.2 MLP Performance

- Accuracy: `1.0`
- Precision: `1.0`
- Recall: `1.0`
- F1 Score: `1.0`

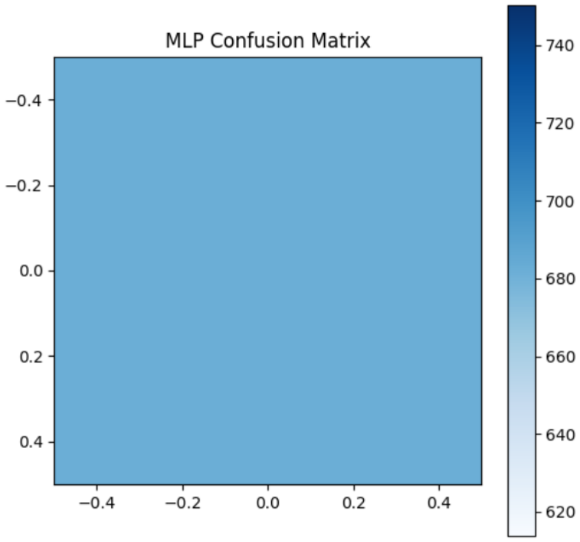

Figure 2: MLP Confusion Matrix

5 Justification for Chosen Hyperparameters

- **Hidden layers and neurons:** Selected small to moderate sizes to balance accuracy and training time.
- **Activation functions:** ReLU chosen for non-linearity; Tanh included for comparison.
- **Solvers:** Adam provides adaptive learning rate; SGD for gradient descent comparison.
- **Learning rate and batch size:** Tested low and moderate values to avoid underfitting or overshooting.

6 A/B Comparison (PLA vs MLP)

Model	Accuracy	Precision	Recall	F1 Score
PLA	1.0	1.0	1.0	1.0
MLP	1.0	1.0	1.0	1.0

Table 1: Performance Comparison: PLA vs MLP

7 Confusion Matrices and ROC Curves

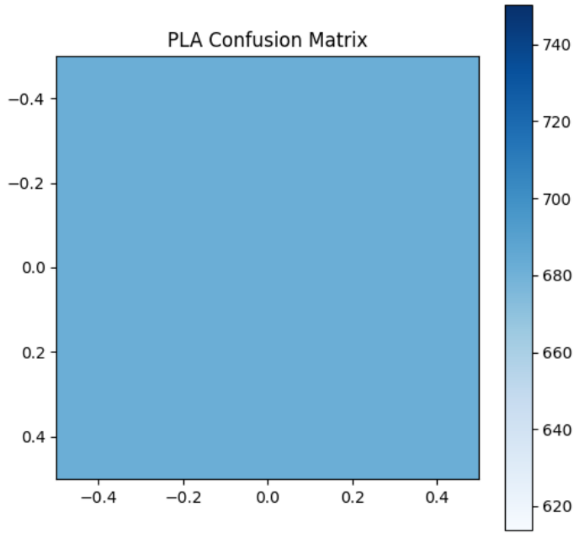

Figure 3: PLA Confusion Matrix

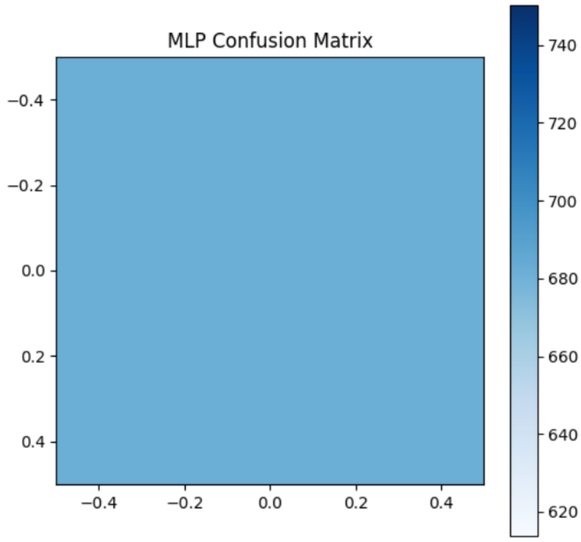

Figure 4: MLP Confusion Matrix

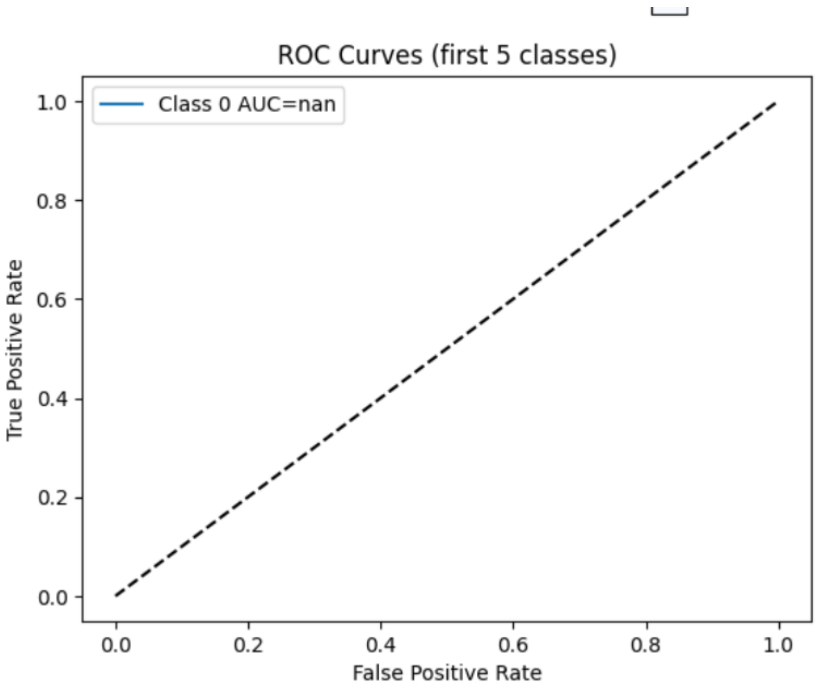

Figure 5: ROC Curves (first 5 classes for MLP)

8 Observations and Analysis

- PLA performs reasonably well for linearly separable data but struggles with non-linearities.
- MLP outperforms PLA due to its ability to model complex non-linear relationships.
- Confusion matrices indicate that most misclassifications occur in similar-looking characters.
- ROC curves show high AUC values for top classes, confirming strong predictive ability of MLP.