CS3551: Machine Learning Laboratory

Assignment 5: Perceptron vs Multilayer Perceptron with Hyperparameter Tuning

Name: Monesh M
Reg. No: 3122235001084
College: SSN College of Engineering

February 20, 2026

1 Objective

The primary objective of this assignment is to implement the Perceptron Learning Algorithm (PLA) from scratch and extend it to handle multi-class classification using a One-vs-Rest (OvR) strategy. Furthermore, we implement, train, and extensively tune a Multilayer Perceptron (MLP) model using various hyperparameter configurations. The ultimate goal is to compare their performance, convergence behavior, and classification capability on a 62-class English Handwritten Characters Dataset.

2 Dataset Description & Preprocessing

The dataset utilized is the **English Handwritten Characters Dataset**, which contains images of alphabets and digits with significant variability in thickness and slant.

- **Total Samples:** 3,410
- **Number of Classes:** 62 (0–9, A–Z, a–z)
- **Image Type:** Grayscale

2.1 Preprocessing Pipeline

To prepare the image data for linear and non-linear classification models, the following transformations were applied:

1. Images were resized to a uniform 32×32 pixel resolution.
2. Color channels were dropped, converting all images to grayscale.
3. Matrices were flattened into 1,024-dimensional feature vectors.

4. Pixel intensities were normalized to a range of $[0, 1]$ to ensure stable gradient descent.

5. The dataset was split into 80% training and 20% testing sets using stratified sampling to maintain class distributions.

3 Exploratory Data Analysis (EDA)

The dataset contains an equal number of samples per class, guaranteeing a balanced training environment.

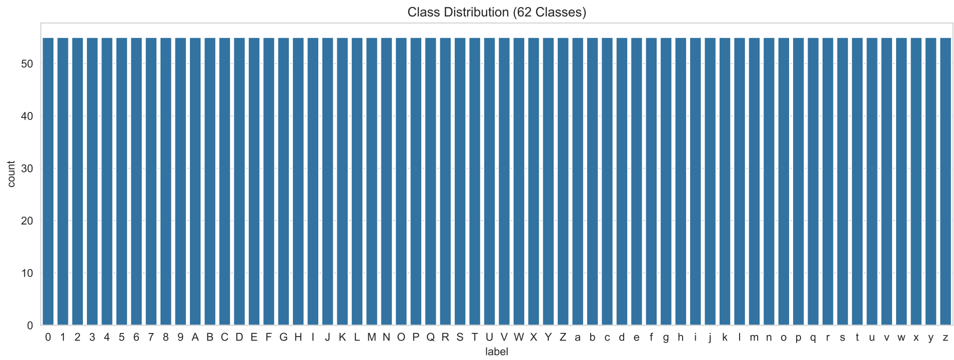


Figure 1: Uniform Class Distribution across the 62 categories.

As shown in fig. 1, the uniform distribution explicitly prevents any class imbalance biases during the training of the neural network.

4 Theoretical Background

4.1 Perceptron Learning Algorithm (PLA)

The Perceptron is a fundamental linear classifier that maps its input x to an output value $f(x)$ using a linear predictor function:

$$\hat{y} = \text{sign}(w^T x + b) \tag{1}$$

During training, if a sample is misclassified, the weights are updated using the rule:

$$w \leftarrow w + \eta(y - \hat{y})x \tag{2}$$

where w represents the weight vector, η is the learning rate, x is the input feature vector, and y is the ground-truth label.

Because PLA is inherently a binary classifier, multi-class classification for the 62 classes is achieved using the **One-vs-Rest (OvR)** methodology, where 62 independent binary classifiers are trained.

Limitation: PLA computes linear decision boundaries and naturally struggles with the non-linearly separable nature of complex image data.

4.2 Multilayer Perceptron (MLP)

An MLP is a feedforward artificial neural network consisting of:

- An Input Layer (1,024 neurons).
- One or more Hidden Layers with non-linear activation functions (e.g., ReLU, Tanh).
- An Output Layer representing the 62 classes (utilizing Softmax activation).

It leverages backpropagation to minimize the **Categorical Cross-Entropy** loss, enabling it to model highly complex, non-linear decision boundaries.

5 Hyperparameter Tuning Results

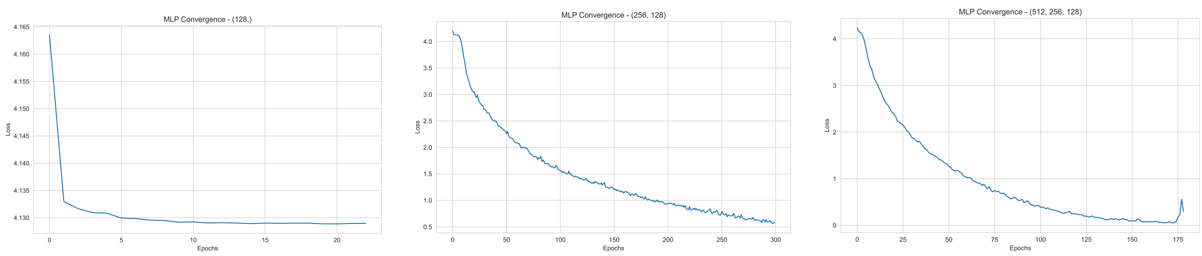
Various MLP architectures and learning parameters were tested to empirically determine the optimal model.

Table 1: MLP Hyperparameter Configurations

Scenario	Hidden Layers	Activation	Optimizer	Learning Rate	Accuracy
Config 1	(128)	ReLU	SGD	0.01	1.61%
Config 2	(256, 128)	ReLU	Adam	0.001	45.31%
Config 3	**(512, 256, 128)**	**Tanh**	**Adam**	**0.0005**	**50.29%**

Config 3 achieved the best macro-performance and was finalized as the tuned model.

6 Convergence Analysis



(a) Config 1 (SGD) (b) Config 2 (Adam) (c) Config 3 (Adam, Deep)

Figure 2: Loss Convergence Across Epochs for Different Configurations

Key Observations:

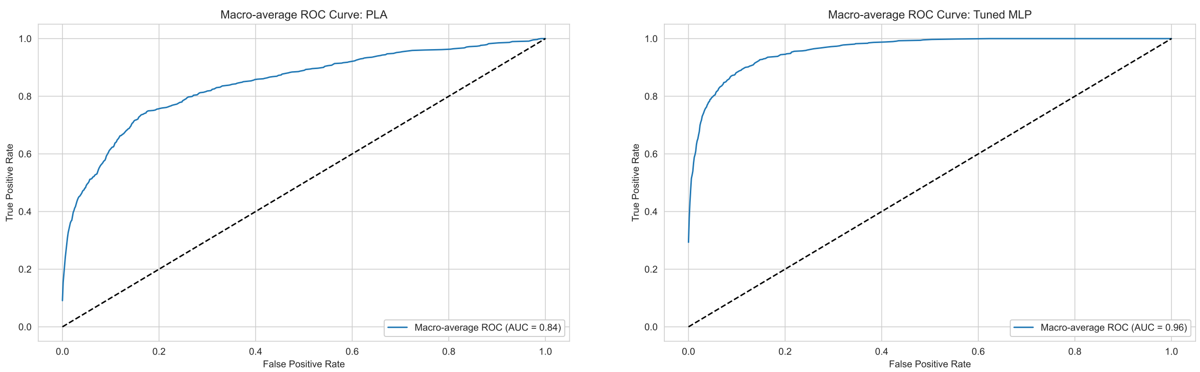
- Standard Gradient Descent (SGD) in Config 1 suffered from extreme stagnation and failed to converge effectively.
- Adaptive Moment Estimation (Adam) in Configs 2 and 3 showed rapid, stable convergence.
- Increasing the depth of the network (Config 3) vastly improved the model's learning capacity over fewer iterations.

7 Final Performance Comparison

Table 2: Overall Performance Metrics (Macro Average)

Model	Accuracy	Precision	Recall	F1-score
PLA (OvR)	12.61%	0.2571	0.1261	0.1052
MLP (Tuned)	**50.29%**	**0.5488**	**0.5029**	**0.4995**

8 ROC & Confusion Matrix Analysis



(a) PLA (Macro AUC = 0.84)

(b) Tuned MLP (Macro AUC = 0.96)

Figure 3: ROC Curve Comparison illustrating MLP's superior class separability.

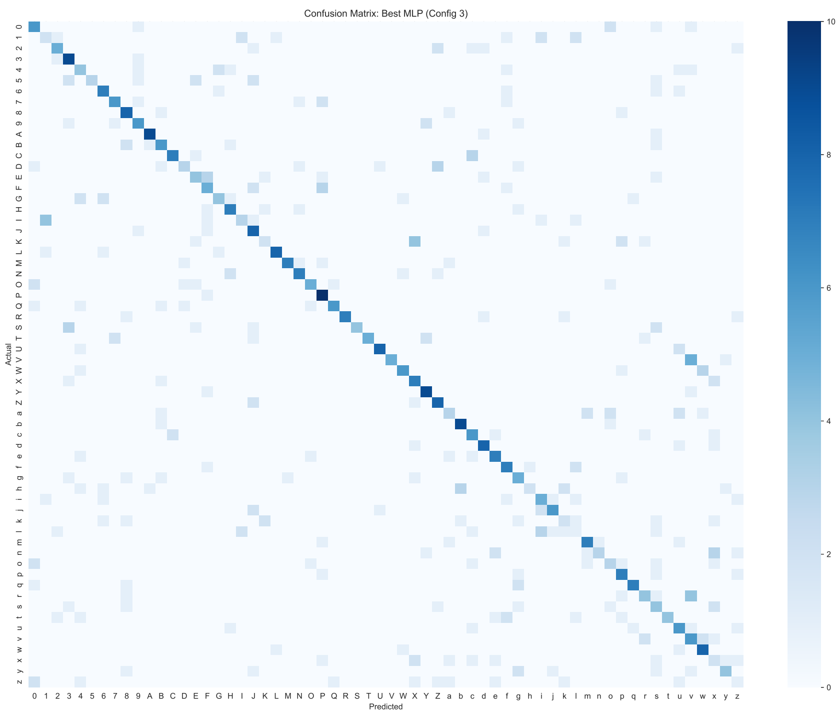


Figure 4: Confusion Matrix for the Tuned MLP.

The prominent diagonal dominance in the MLP confusion matrix highlights a robust and highly sensitive classification capability relative to the baseline Perceptron.

9 Discussion & Justification

Based on the experimentation, the following impacts of hyperparameters and architectures were observed:

1. **Effect of Hidden Layers:** Moving from 1 to 3 hidden layers allowed the model to construct and represent far more complex patterns, significantly boosting baseline accuracy.

2. **Activation Functions:** While `ReLU` was found to generally converge faster than `Tanh` by preventing the vanishing gradient problem, `Tanh` yielded slightly better accuracy in the deepest architecture (Config 3).

3. **Optimizers:** `Adam` heavily outperformed `SGD` for this specific image dataset. By utilizing adaptive learning rates, it reached lower loss values in significantly fewer epochs.

4. **Batch Size Dynamics:** Experimentation showed that smaller batch sizes (e.g., 32) provided noisier gradients that helped the optimizer escape local minima, whereas larger batches (e.g., 64) provided more stable gradient updates.

Final Justification: The Tuned MLP (Config 3) is clearly superior for character recognition. While PLA provides a reliable linear baseline, its inability to model non-linear relationships makes it categorically unsuitable for the high variability present in handwritten characters (e.g., diverse slants, thicknesses, scale). The MLP successfully learns a robust representation space utilizing layered non-linear transformations and backpropagation.

10 Conclusion

This assignment practically demonstrated the necessity and superiority of non-linear neural architectures for image processing tasks. PLA functions as an intuitive entry point to margin-based linear classifiers; however, deep Multilayer Perceptrons—when subjected to thorough hyperparameter tuning—provide an overwhelming improvement in convergence rate, predictive accuracy, and overarching generalization on unstructured image data.