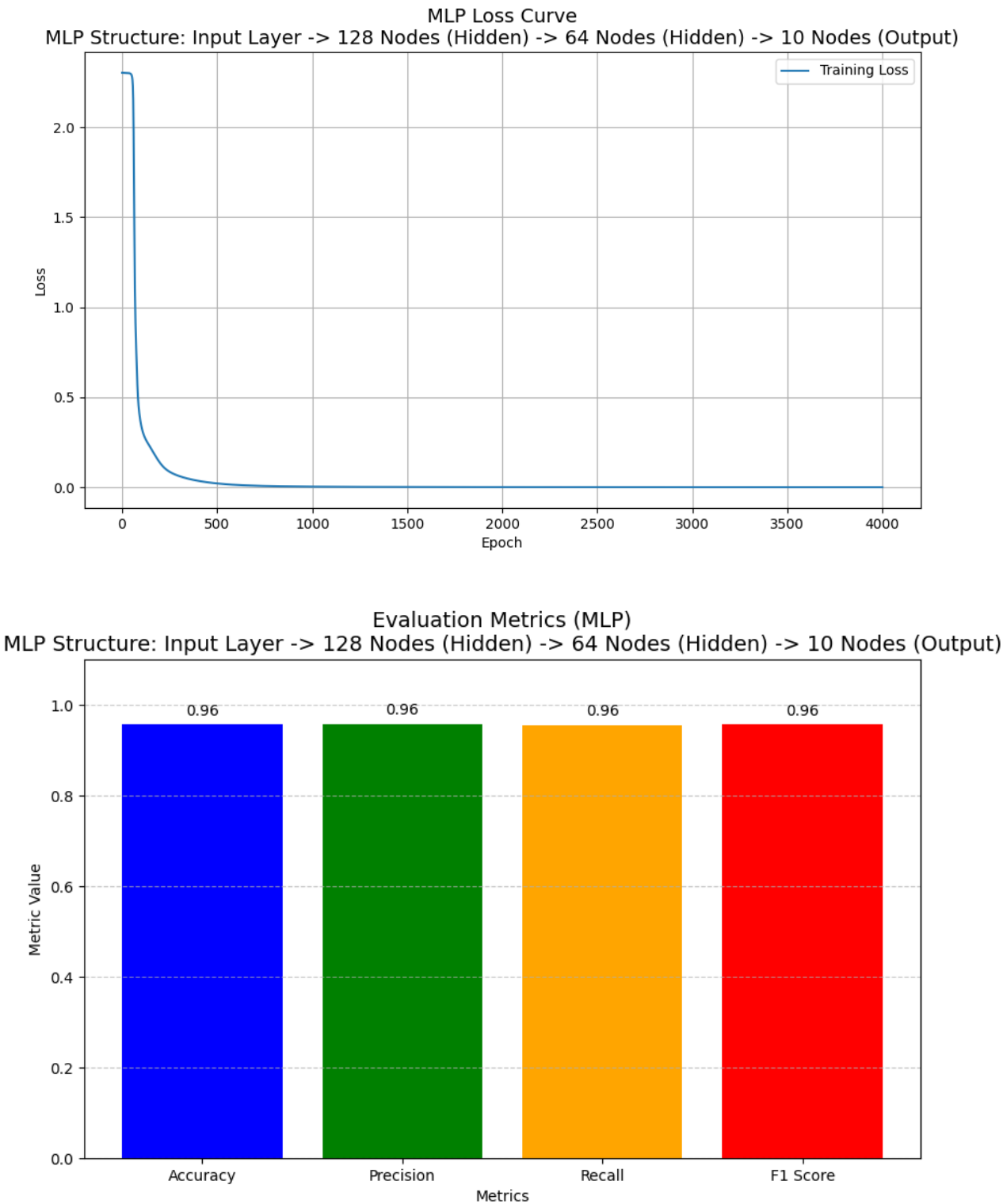The implementation begins with data preprocessing, where the 8x8 pixel image features are normalized to the range [0, 1], and numerical labels are converted to One-Hot encoding to suit a multi-class classification task. The MLP model is designed with an input layer, two hidden layers, and an output layer, using ReLU and Softmax as activation functions. Weights and biases are initialized with small random values. During forward propagation, activation values are computed for each layer. The cross-entropy loss function is used to measure the difference between predictions and true labels. Backpropagation calculates gradients and updates parameters to minimize the loss. The training process optimizes the model through multiple iterations and mini-batch gradient descent. Finally, the model's performance is evaluated using test data, calculating accuracy, precision, recall, and F1 score. Loss curves and evaluation metric bar charts are plotted to visually present the results.

MLP Loss Curve
MLP Structure: Input Layer -> 128 Nodes (Hidden) -> 64 Nodes (Hidden) -> 10 Nodes (Output)

Evaluation Metrics (MLP)
MLP Structure: Input Layer -> 128 Nodes (Hidden) -> 64 Nodes (Hidden) -> 10 Nodes (Output)

The MLP model achieved high performance with 96% accuracy, precision, recall, and F1 score. The loss curve shows rapid convergence and stabilization, indicating efficient learning and a well-fitted model.

View the code on GitHub