Foundational AI Project 1

Multilayer Perceptrons (MLPs)

CSC 7700

Toheeb Balogun

1.0 Abstract

In this project, a custom Multilayer Perceptron (MLP) engine was developed and applied to two datasets: the MNIST dataset for classification and the Vehicle MPG dataset for regression. The MLP engine was built from scratch, implementing various activation and loss functions, as well as early stopping and L2 regularization techniques. The model achieved an accuracy of 94.79 % on the MNIST test set and a total testing loss of 0.109 on the Vehicle MPG dataset.

2.0 Methodology

The MLP engine was coded from the ground up using Python and key libraries such as NumPy and Pandas. The architecture included custom implementations of activation functions (e.g., ReLU, SoftMax, Linear), loss functions (CrossEntropy, SquaredError), and optimization techniques like RMSProp and early stopping.

For the MNIST classification, a network architecture with 784 input neurons, two hidden layers (128 and 64 neurons with ReLU activation), and 10 output neurons with Softmax activation was used. The Vehicle MPG regression model used a smaller network with ReLU activation and a Linear output layer.

Challenges included implementing the Softmax derivative correctly and avoiding data leakage during validation. These were addressed by refining the backpropagation logic and ensuring proper data splitting strategies.

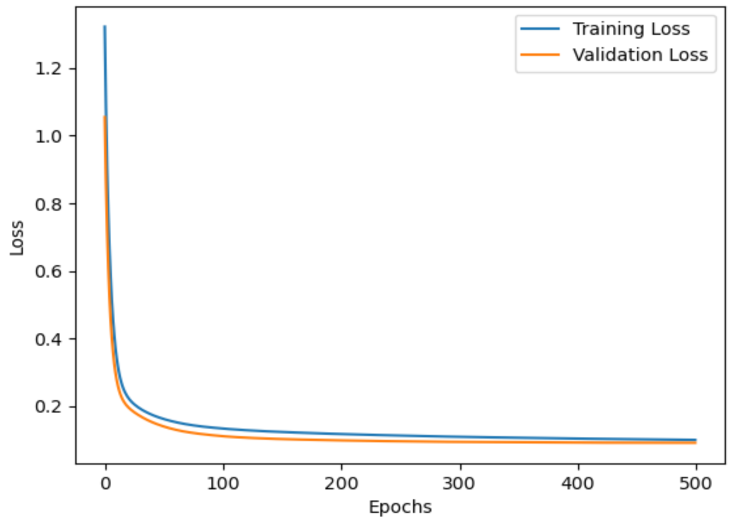
3.0 Results

3.1 Vehicle MPG Dataset

- Total Testing Loss: **0.109**
- **Predicted vs True MPG Table**:

No	Predicted MPG	True MPG
1	1.003171	0.294705
2	0.187896	0.419452
3	-0.860596	-0.952771
4	0.944210	0.294705
5	0.230395	0.182432
6	0.462751	0.294705
7	-0.635210	-0.890397
8	-1.168604	-1.327014
9	-0.450023	-0.466255
10	-0.668187	-0.678326

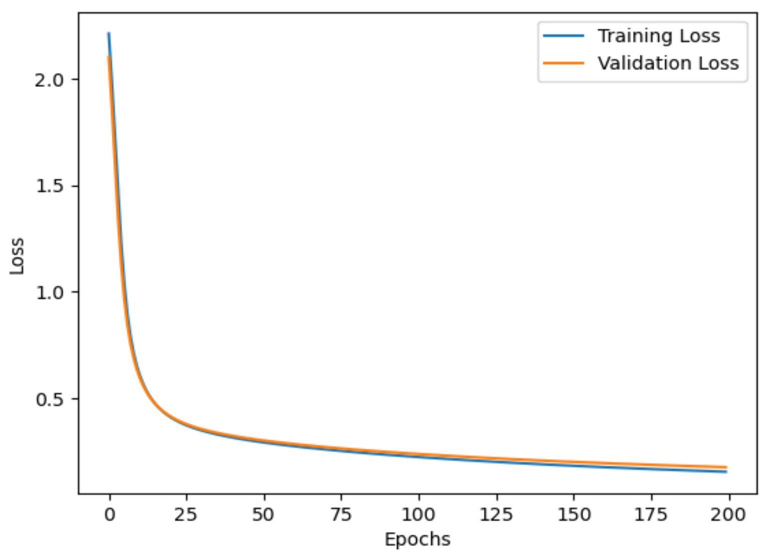
- **Loss Curves**



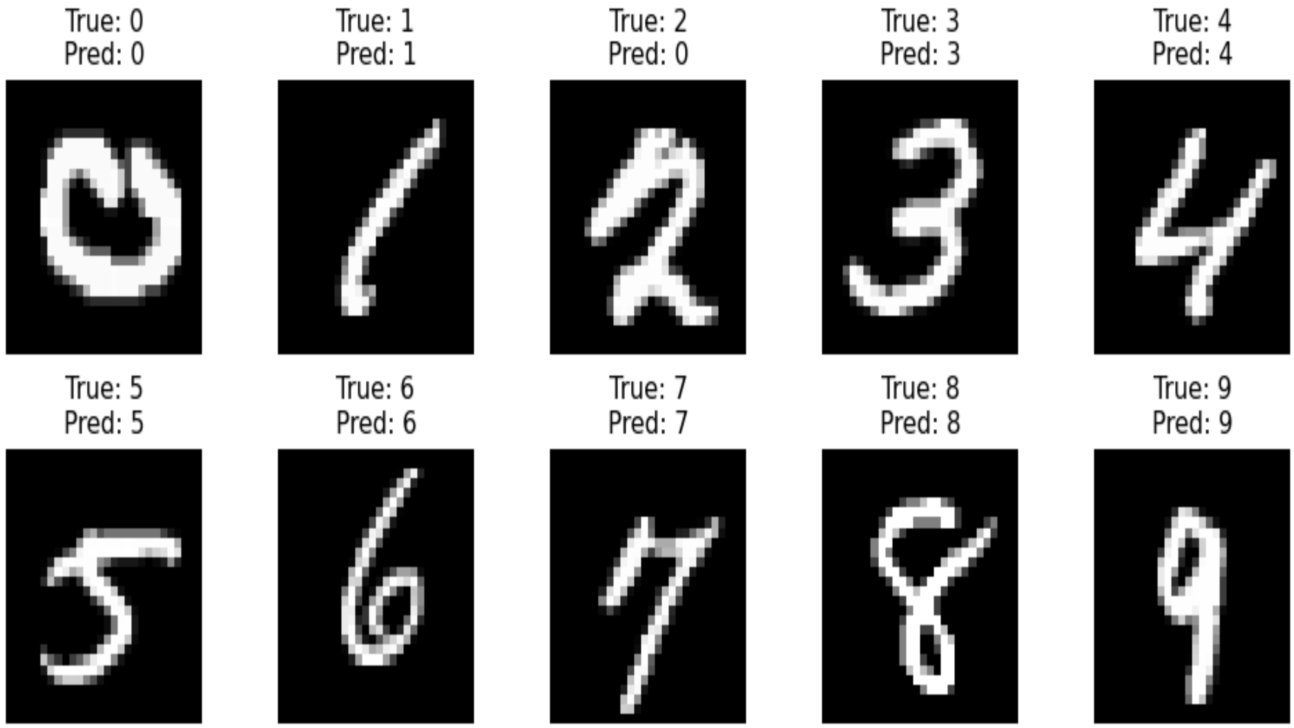
3.2 MNIST Dataset

- **Test Accuracy**: 94.79 %

- **Loss Curves**



- **Sample Image Predictions**



Code Repo Link: https://github.com/btoheeb1/CSC-7700.git

Discussion & Conclusion

This project provided valuable insights into building neural networks from scratch. The custom MLP engine performed well on both classification and regression tasks, demonstrating the flexibility of neural networks. Key learnings included the importance of data preprocessing, the impact of activation function choices, and the benefits of early stopping in preventing overfitting.

Future work could explore more advanced architectures, such as Convolutional Neural Networks (CNNs) for image data, and integrating more sophisticated optimization techniques to further enhance performance.