Intro Deep Learning Homework 1

Jaskin Kabir
Student Id: 801186717
GitHub:
`https://github.com/jaskinkabir/Intro_ML/tree/main/HM7`

January 2025

1 Problem 1: Multilayer Perceptrons For Image Classification

1a. *Three Hidden Layers*

Using three hidden layers, consisting of 64, 32, and 16 neurons respectively, the model was trained for 20 epochs. As seen in the curves graphed in Figure 1, the training loss curve has not yet begun to converge to the optimal solution after just 20 epochs. To fully train the model, more epochs would be required. There is also significant overfit, as indicated by the gap between the training and validation accuracy curves.

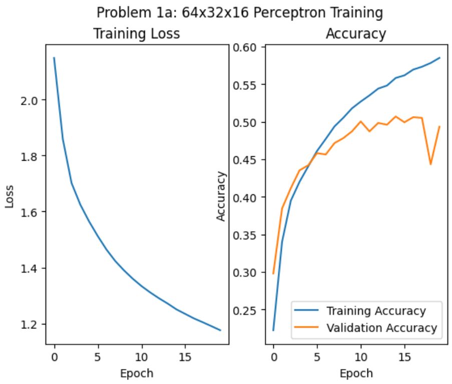

Figure 1: Loss and Accuracy Curves for Three Hidden Layers

The confusion matrix for this model can be seen in Figure 2.

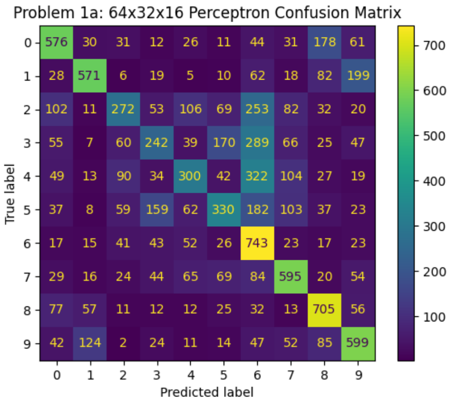

Figure 2: Confusion Matrix for Three Hidden Layers

1b. *Five Hidden Layers*

Using five hidden layers, consisting of 256, 128, 64, 32, and 16 neurons respectively, the model was trained for 20 epochs. As seen in the curves graphed in Figure 3, the model has not yet converged to the optimal solution after 20 epochs. There is also significant overfit, but not as pronounced as it is in the three hidden layer model. The confusion matrix for this model can be seen in Figure 4.

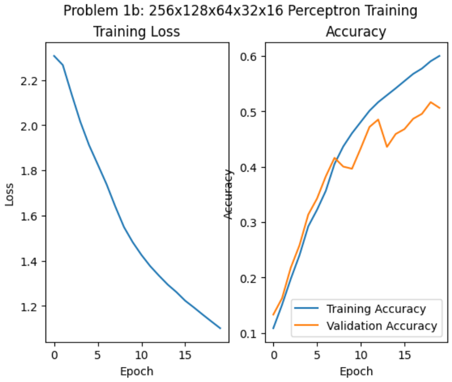

Figure 3: Loss and Accuracy Curves for Five Hidden Layers

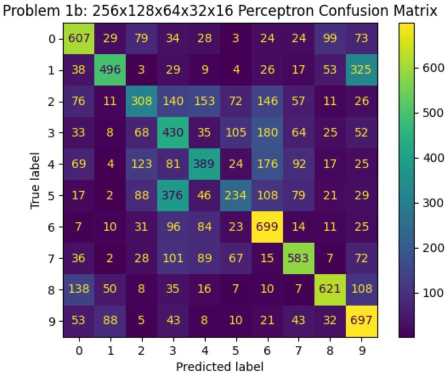

Figure 4: Confusion Matrix for Five Hidden Layers

Comparison

Table 1 shows that the more complex model performed better in every metric. However, even the 5-layer model was barely more accurate than a coin toss. The models are likely underfit, as they have not yet converged to the optimal solution. Additionally, the multilayer perceptron is not the ideal paradigm for image classification. A convolutional neural network would likely outperform both of these models.

Metric (%)	3-Layer Model	5-Layer Model	Δ
Accuracy	49.3	50.6	2.59
Precision	49.8	51.6	3.37
Recall	49.3	50.6	2.59
F1 Score	48.2	50.1	3.69

Table 1: Comparison of Evaluation Metrics

2 Problem 2: Housing Price Regression

1a. *Perceptron Regressor*

The housing dataset has several categorical features. In order to use this data to train the model, the categorical values like 'yes' or 'no' were converted to 1 and 0 repsectively. Using 2 layers, woth 64 and 32 neurons respectively, the model was trained for 500 epochs. As seen in the curves graphed in Figure 5, the model has

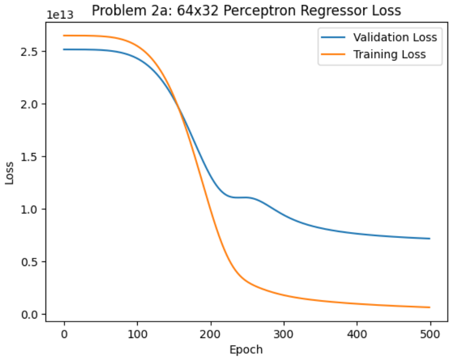

Figure 5: Loss and Accuracy Curves for Perceptron Regressor