Balaj Khalid

CSCI 561 – Homework 3 Report

For this homework assignment, I developed a multi-layer perceptron to analyze the New York housing market. I started by loading and cleaning the data, which included removing duplicates, updating data types, and checking for null values. During exploratory data analysis, I identified significant noise and outliers in the price and property square feet columns, which could potentially destabilize the model. To address this, I decided to bin these variables for better stability.

Additionally, I discovered that latitude and longitude alone were insufficient predictors of housing prices. However, properties closer to Central Park tended to be more expensive, prompting me to create a new feature based on the distance from Central Park using these coordinates. This feature, along with information on locality and sublocality, enriched the dataset sufficiently to proceed with training the model. I deemed other address-related columns too granular and therefore not particularly useful for our analysis.

With the preprocessing and feature engineering complete, I constructed a custom multi-layer perceptron model that utilized both RELU and Sigmoid activation functions. I then optimized the model by experimenting with various hyper-parameters. The selected hyper-parameters, as shown below, produced the best results:

```
hidden_layers = [50]
learning_rate = 0.03
n_epochs = 1000
batch_size = 500
activation_function = 'relu'
regularization_coefficient = 0.01
```

Results:

```
Data 1, accuracy of the model on Training: 0.5346, Testing: 0.4809, Percentage Baseline: 102.61%
Data 2, accuracy of the model on Training: 0.5271, Testing: 0.4564, Percentage Baseline: 96.82%
Data 3, accuracy of the model on Training: 0.53, Testing: 0.4666, Percentage Baseline: 93.31%
Data 4, accuracy of the model on Training: 0.527, Testing: 0.4599, Percentage Baseline: 92.04%
Data 5, accuracy of the model on Training: 0.5145, Testing: 0.4702, Percentage Baseline: 95.81%
```

In my efforts to fine-tune the model, I focused on optimizing the following hyperparameters:

Weight Initializations

I experimented with several methods of weight initialization, including Random, Xavier, He, and Uniform Initialization,

Random Initialization:

```
Data 1, accuracy of the model on Training: 0.5334, Testing: 0.4687, Percentage Baseline: 99.99%
Data 2, accuracy of the model on Training: 0.5331, Testing: 0.4693, Percentage Baseline: 99.55%
Data 3, accuracy of the model on Training: 0.5291, Testing: 0.4589, Percentage Baseline: 91.78%
Data 4, accuracy of the model on Training: 0.5348, Testing: 0.462, Percentage Baseline: 92.46%
Data 5, accuracy of the model on Training: 0.5326, Testing: 0.4864, Percentage Baseline: 99.1%
```

Xavier Initialization:

```
Data 1, accuracy of the model on Training: 0.5312, Testing: 0.4721, Percentage Baseline: 100.72%
Data 2, accuracy of the model on Training: 0.5396, Testing: 0.4643, Percentage Baseline: 98.49%
Data 3, accuracy of the model on Training: 0.5354, Testing: 0.4526, Percentage Baseline: 90.53%
Data 4, accuracy of the model on Training: 0.5309, Testing: 0.4606, Percentage Baseline: 92.18%
Data 5, accuracy of the model on Training: 0.5398, Testing: 0.4783, Percentage Baseline: 97.46%
```

He Initialization:

```
Data 1, accuracy of the model on Training: 0.5413, Testing: 0.4544, Percentage Baseline: 96.94%
Data 2, accuracy of the model on Training: 0.5351, Testing: 0.4614, Percentage Baseline: 97.88%
Data 3, accuracy of the model on Training: 0.543, Testing: 0.4666, Percentage Baseline: 93.31%
Data 4, accuracy of the model on Training: 0.5376, Testing: 0.4467, Percentage Baseline: 89.39%
Data 5, accuracy of the model on Training: 0.5288, Testing: 0.4724, Percentage Baseline: 96.26%
```

Uniform Initialization:

```
Data 1, accuracy of the model on Training: 0.5269, Testing: 0.468, Percentage Baseline: 99.85%
Data 2, accuracy of the model on Training: 0.5274, Testing: 0.4686, Percentage Baseline: 99.4%
Data 3, accuracy of the model on Training: 0.5257, Testing: 0.461, Percentage Baseline: 92.2%
Data 4, accuracy of the model on Training: 0.5288, Testing: 0.4557, Percentage Baseline: 91.2%
Data 5, accuracy of the model on Training: 0.5193, Testing: 0.4783, Percentage Baseline: 97.46%
```

Observation: Random and Uniform Initializations yield comparable test accuracies. However, I opted for Uniform Initialization due to smaller generalization gap.

Number of Layers

In exploring both deep and wide networks, I tested configurations with different numbers of hidden layers,

250, 500, 250 perceptron:

```
Data 1, accuracy of the model on Training: 0.5628, Testing: 0.4789, Percentage Baseline: 102.17%
Data 2, accuracy of the model on Training: 0.5612, Testing: 0.4807, Percentage Baseline: 101.98%
Data 3, accuracy of the model on Training: 0.5587, Testing: 0.468, Percentage Baseline: 93.59%
Data 4, accuracy of the model on Training: 0.5536, Testing: 0.4634, Percentage Baseline: 92.74%
Data 5, accuracy of the model on Training: 0.5593, Testing: 0.4769, Percentage Baseline: 97.16%
```

250 perceptron:

```
Data 1, accuracy of the model on Training: 0.5548, Testing: 0.468, Percentage Baseline: 99.85%
Data 2, accuracy of the model on Training: 0.5438, Testing: 0.4714, Percentage Baseline: 100.01%
Data 3, accuracy of the model on Training: 0.5548, Testing: 0.4603, Percentage Baseline: 92.06%
Data 4, accuracy of the model on Training: 0.5445, Testing: 0.4578, Percentage Baseline: 91.62%
Data 5, accuracy of the model on Training: 0.5445, Testing: 0.482, Percentage Baseline: 98.21%
```

100 perceptron:

```
Data 1, accuracy of the model on Training: 0.5331, Testing: 0.47, Percentage Baseline: 100.28%
Data 2, accuracy of the model on Training: 0.5325, Testing: 0.4636, Percentage Baseline: 98.34%
Data 3, accuracy of the model on Training: 0.5412, Testing: 0.4547, Percentage Baseline: 90.95%
Data 4, accuracy of the model on Training: 0.5373, Testing: 0.4613, Percentage Baseline: 92.32%
Data 5, accuracy of the model on Training: 0.5335, Testing: 0.4791, Percentage Baseline: 97.61%
```

50 perceptron:

```
Data 1, accuracy of the model on Training: 0.5346, Testing: 0.4809, Percentage Baseline: 102.61%
Data 2, accuracy of the model on Training: 0.5271, Testing: 0.4564, Percentage Baseline: 96.82%
Data 3, accuracy of the model on Training: 0.53, Testing: 0.4666, Percentage Baseline: 93.31%
Data 4, accuracy of the model on Training: 0.527, Testing: 0.4599, Percentage Baseline: 92.04%
Data 5, accuracy of the model on Training: 0.5145, Testing: 0.4702, Percentage Baseline: 95.81%
```

Observation: Deep and wide networks did not significantly impact performance. Consequently, I opted for a simpler model with one hidden layer of 50 perceptron, which I was able to train over more epochs for improved accuracy.

Activation Functions

I compared the performance of the RELU and Sigmoid activation functions:

RELU:

```
Data 1, accuracy of the model on Training: 0.5346, Testing: 0.4809, Percentage Baseline: 102.61%
Data 2, accuracy of the model on Training: 0.5271, Testing: 0.4564, Percentage Baseline: 96.82%
Data 3, accuracy of the model on Training: 0.53, Testing: 0.4666, Percentage Baseline: 93.31%
Data 4, accuracy of the model on Training: 0.527, Testing: 0.4599, Percentage Baseline: 92.04%
Data 5, accuracy of the model on Training: 0.5145, Testing: 0.4702, Percentage Baseline: 95.81%
```

Sigmoid:

```
Data 1, accuracy of the model on Training: 0.4743, Testing: 0.4285, Percentage Baseline: 91.42%
Data 2, accuracy of the model on Training: 0.4616, Testing: 0.45, Percentage Baseline: 95.46%
Data 3, accuracy of the model on Training: 0.4504, Testing: 0.4366, Percentage Baseline: 87.33%
Data 4, accuracy of the model on Training: 0.4667, Testing: 0.4202, Percentage Baseline: 84.09%
Data 5, accuracy of the model on Training: 0.4543, Testing: 0.4269, Percentage Baseline: 86.98%
```

Observation: RELU outperformed Sigmoid by 5% in test accuracy over datasets, making it the preferred choice for the final model settings.

Learning Rate

I tested various learning rates to find the optimal speed for convergence:

0.0001:

```
Data 1, accuracy of the model on Training: 0.2909, Testing: 0.2875, Percentage Baseline: 61.33%
Data 2, accuracy of the model on Training: 0.3162, Testing: 0.3157, Percentage Baseline: 66.97%
Data 3, accuracy of the model on Training: 0.3117, Testing: 0.3224, Percentage Baseline: 64.48%
Data 4, accuracy of the model on Training: 0.2882, Testing: 0.269, Percentage Baseline: 53.83%
Data 5, accuracy of the model on Training: 0.2982, Testing: 0.3123, Percentage Baseline: 63.62%
```

0.005:

```
Data 1, accuracy of the model on Training: 0.4783, Testing: 0.4401, Percentage Baseline: 93.89%
Data 2, accuracy of the model on Training: 0.4855, Testing: 0.4521, Percentage Baseline: 95.91%
Data 3, accuracy of the model on Training: 0.4749, Testing: 0.4401, Percentage Baseline: 88.02%
Data 4, accuracy of the model on Training: 0.4824, Testing: 0.4258, Percentage Baseline: 85.21%
Data 5, accuracy of the model on Training: 0.4757, Testing: 0.4467, Percentage Baseline: 91.02%
```

0.01:

```
Data 1, accuracy of the model on Training: 0.5009, Testing: 0.4489, Percentage Baseline: 95.78%
Data 2, accuracy of the model on Training: 0.5052, Testing: 0.4586, Percentage Baseline: 97.28%
Data 3, accuracy of the model on Training: 0.4964, Testing: 0.4526, Percentage Baseline: 90.53%
Data 4, accuracy of the model on Training: 0.5018, Testing: 0.4418, Percentage Baseline: 88.42%
Data 5, accuracy of the model on Training: 0.4902, Testing: 0.4592, Percentage Baseline: 93.57%
```

0.03:

```
Data 1, accuracy of the model on Training: 0.5346, Testing: 0.4809, Percentage Baseline: 102.61%
Data 2, accuracy of the model on Training: 0.5271, Testing: 0.4564, Percentage Baseline: 96.82%
Data 3, accuracy of the model on Training: 0.53, Testing: 0.4666, Percentage Baseline: 93.31%
Data 4, accuracy of the model on Training: 0.527, Testing: 0.4599, Percentage Baseline: 92.04%
Data 5, accuracy of the model on Training: 0.5145, Testing: 0.4702, Percentage Baseline: 95.81%
```

Observation: Except for 0.03, all other rates resulted in slower convergence than desired across the set number of epochs.

Regularization Value

To combat overfitting, I implemented L2 regularization, experimenting with different coefficients:

0.0001:

```
Data 1, accuracy of the model on Training: 0.5799, Testing: 0.4809, Percentage Baseline: 102.61%
Data 2, accuracy of the model on Training: 0.5845, Testing: 0.4707, Percentage Baseline: 99.85%
Data 3, accuracy of the model on Training: 0.5811, Testing: 0.4589, Percentage Baseline: 91.78%
Data 4, accuracy of the model on Training: 0.5894, Testing: 0.4683, Percentage Baseline: 93.71%
Data 5, accuracy of the model on Training: 0.5861, Testing: 0.4761, Percentage Baseline: 97.01%
```

0.01:

```
Data 1, accuracy of the model on Training: 0.5346, Testing: 0.4809, Percentage Baseline: 102.61%
Data 2, accuracy of the model on Training: 0.5271, Testing: 0.4564, Percentage Baseline: 96.82%
Data 3, accuracy of the model on Training: 0.53, Testing: 0.4666, Percentage Baseline: 93.31%
Data 4, accuracy of the model on Training: 0.527, Testing: 0.4599, Percentage Baseline: 92.04%
Data 5, accuracy of the model on Training: 0.5145, Testing: 0.4702, Percentage Baseline: 95.81%
```

0.05:

```
Data 1, accuracy of the model on Training: 0.4697, Testing: 0.4312, Percentage Baseline: 92.0%
Data 2, accuracy of the model on Training: 0.466, Testing: 0.4464, Percentage Baseline: 94.7%
Data 3, accuracy of the model on Training: 0.4528, Testing: 0.4422, Percentage Baseline: 88.44%
Data 4, accuracy of the model on Training: 0.4697, Testing: 0.4307, Percentage Baseline: 86.18%
Data 5, accuracy of the model on Training: 0.4614, Testing: 0.4379, Percentage Baseline: 89.22%
```

Observation: A regularization coefficient of 0.01 produced the most favorable outcomes. While a coefficient of 0.0001 shows higher test accuracy, it also exhibits a larger generalization gap, suggesting potential overfitting of the training data by the model.

Batch Size

Different batch sizes were tested to optimize learning:

200:

```
Data 1, accuracy of the model on Training: 0.5361, Testing: 0.4673, Percentage Baseline: 99.7%
Data 2, accuracy of the model on Training: 0.5456, Testing: 0.4571, Percentage Baseline: 96.98%
Data 3, accuracy of the model on Training: 0.5481, Testing: 0.4659, Percentage Baseline: 93.18%
Data 4, accuracy of the model on Training: 0.5424, Testing: 0.4585, Percentage Baseline: 91.76%
Data 5, accuracy of the model on Training: 0.5323, Testing: 0.4842, Percentage Baseline: 98.66%
```

500:

```
Data 1, accuracy of the model on Training: 0.5346, Testing: 0.4809, Percentage Baseline: 102.61%
Data 2, accuracy of the model on Training: 0.5271, Testing: 0.4564, Percentage Baseline: 96.82%
Data 3, accuracy of the model on Training: 0.53, Testing: 0.4666, Percentage Baseline: 93.31%
Data 4, accuracy of the model on Training: 0.527, Testing: 0.4599, Percentage Baseline: 92.04%
Data 5, accuracy of the model on Training: 0.5145, Testing: 0.4702, Percentage Baseline: 95.81%
```

800:

```
Data 1, accuracy of the model on Training: 0.5104, Testing: 0.453, Percentage Baseline: 96.65%
Data 2, accuracy of the model on Training: 0.516, Testing: 0.4571, Percentage Baseline: 96.98%
Data 3, accuracy of the model on Training: 0.5121, Testing: 0.4429, Percentage Baseline: 88.58%
Data 4, accuracy of the model on Training: 0.5224, Testing: 0.4564, Percentage Baseline: 91.34%
Data 5, accuracy of the model on Training: 0.5095, Testing: 0.4578, Percentage Baseline: 93.27%
```

Observation: A batch size of 500 proved to be the most effective for this problem. Although batch sizes of 200 and 800 yielded comparable test accuracies, the batch size of 500 offered the best balance between test accuracy and runtime, particularly important given our 5-minute time constraint.

Epochs

I varied the number of training epochs to determine an adequate training duration:

200:

```
Data 1, accuracy of the model on Training: 0.4923, Testing: 0.4428, Percentage Baseline: 94.47%
Data 2, accuracy of the model on Training: 0.4954, Testing: 0.4486, Percentage Baseline: 95.16%
Data 3, accuracy of the model on Training: 0.4855, Testing: 0.4415, Percentage Baseline: 88.3%
Data 4, accuracy of the model on Training: 0.483, Testing: 0.4341, Percentage Baseline: 86.88%
Data 5, accuracy of the model on Training: 0.4736, Testing: 0.4555, Percentage Baseline: 92.82%
```

500:

```
Data 1, accuracy of the model on Training: 0.5135, Testing: 0.4605, Percentage Baseline: 98.25%
Data 2, accuracy of the model on Training: 0.5106, Testing: 0.4636, Percentage Baseline: 98.34%
Data 3, accuracy of the model on Training: 0.5058, Testing: 0.4575, Percentage Baseline: 91.5%
Data 4, accuracy of the model on Training: 0.5097, Testing: 0.4557, Percentage Baseline: 91.2%
Data 5, accuracy of the model on Training: 0.4997, Testing: 0.4739, Percentage Baseline: 96.56%
```

1000:

```
Data 1, accuracy of the model on Training: 0.5346, Testing: 0.4809, Percentage Baseline: 102.61%
Data 2, accuracy of the model on Training: 0.5271, Testing: 0.4564, Percentage Baseline: 96.82%
Data 3, accuracy of the model on Training: 0.53, Testing: 0.4666, Percentage Baseline: 93.31%
Data 4, accuracy of the model on Training: 0.527, Testing: 0.4599, Percentage Baseline: 92.04%
Data 5, accuracy of the model on Training: 0.5145, Testing: 0.4702, Percentage Baseline: 95.81%
```

2000:

```
Data 1, accuracy of the model on Training: 0.5337, Testing: 0.4673, Percentage Baseline: 99.7%
Data 2, accuracy of the model on Training: 0.5393, Testing: 0.4686, Percentage Baseline: 99.4%
Data 3, accuracy of the model on Training: 0.5418, Testing: 0.4673, Percentage Baseline: 93.45%
Data 4, accuracy of the model on Training: 0.5409, Testing: 0.4606, Percentage Baseline: 92.18%
Data 5, accuracy of the model on Training: 0.5374, Testing: 0.4754, Percentage Baseline: 96.86%
```

Observation: Training beyond 1000 epochs led to overfitting, resulting in minimal or even negative impacts on test performance. Therefore, 1000 epochs were chosen as optimal, as they not only provided a test accuracy improvement of over 2% compared to fewer epochs but also fit within our 5-minute runtime constraint.