Assignment 7.1: Deep Learning for Regression and Classification

Patricia Enrique

Department of Engineering, University of San Diego

AAI-511: Neural Networks and Learning

Prof. Esnaeilli

August 14, 2023

Analysis Setup

The data used for this analysis is downloaded and separated into data and targets for both the regression and classification models. For the classification model, the distribution of the median value of owner-occupied homes (targets) is visualized and the median value is calculated to determine the price point that will separate the expensive from the inexpensive prices. From this, it is determined that targets with a value over 22 are expensive and in class 1, and prices equal to or below 22 are inexpensive and in class 0. The data for both the regression and classification models are then split into training and testing datasets before being scaled and normalized.

The baseline regression model consists of one hidden layer with 64 units. This model uses the Adam optimizer with a learning rate set to 0.01, a mean absolute error loss function, and a default epoch value of 10 (TensorFlow, 2023). The performance metric for the regression model is loss and the results from the baseline model are summarized in Table 1.

The baseline classification model consists of one hidden layer with 64 units. This model uses the Adam optimizer with a learning rate set to 0.01, a binary cross entropy loss function, and a default epoch value of 10. The performance metrics for the classification model are loss, accuracy, precision, and recall, and the results from the baseline model are summarized in Table 2.

Table 1

Baseline model performance metric for the regression model

Metric	Performance
Loss	2.34

Table 2

Baseline model performance metrics for the classification model

Metric	Performance

Loss	0.27
Accuracy	0.89
Precision	0.87
Recall	0.85

Hyperparameter Tuning and Results

Part 1: Regression Problem

The results from experimenting with different number of hidden layers, units, learning rates, and epochs in the deep neural network architecture for regression are summarized in the tables that follow. From Table 3 it can be seen that the performance of the model remains consistent with the addition of hidden layers. Each layer that is added contains the same number of units as the baseline model.

Table 3

Exploration of the number of hidden layers

Performance Metric	Baseline (1)	2	3	4
Loss	2.34	2.64	2.58	2.47

Table 4 summarizes the results from experimenting with units in the hidden layer. A reduction in units results in greater loss, and so does the addition of too many units. The baseline model has a similar result as increasing the units to 100, which is the optimal amount.

Table 4

Exploration of units

Performance Metric	10	Baseline(64)	100	250
Loss	3.73	2.34	2.26	2.71

Table 5 summarizes the results from experimenting with different learning rates. This parameter has the most influence on the model's performance with the baseline model performing the best and smaller learning rates obtaining drastically poorer performances.

Table 5

Exploration of learning rates

Performance Metric	0.1	Baseline(0.01)	0.001	0.0001
Loss	2.69	2.34	15.04	21.18

Table 6 summarizes the results from experimenting with different epochs. The baseline model has the worst performance and an epoch value of 50 obtains the best results.

Table 6

Exploration of epochs

Performance Metric	Baseline(10)	50	100	500
Loss	2.34	2.09	2.17	2.18

The final model uses the findings from the hidden layer, units, learning rate, and epochs exploration to optimize the performance. The model consists of 1 layer with 100 units, a learning rate of 0.01, and 50 epochs. The final model performance is compared with the baseline model performance in Table 7 and shows that they obtain similar results. Figure 1 and Figure 2 compare the training loss with the validation loss in the final model and the predictions of the median value of owner-occupied homes with the true values.

Table 7

Final model performance metrics on test dataset

Metric	Baseline	Final Model
Loss	2.34	2.17

Figure 1

Training and validation loss comparison

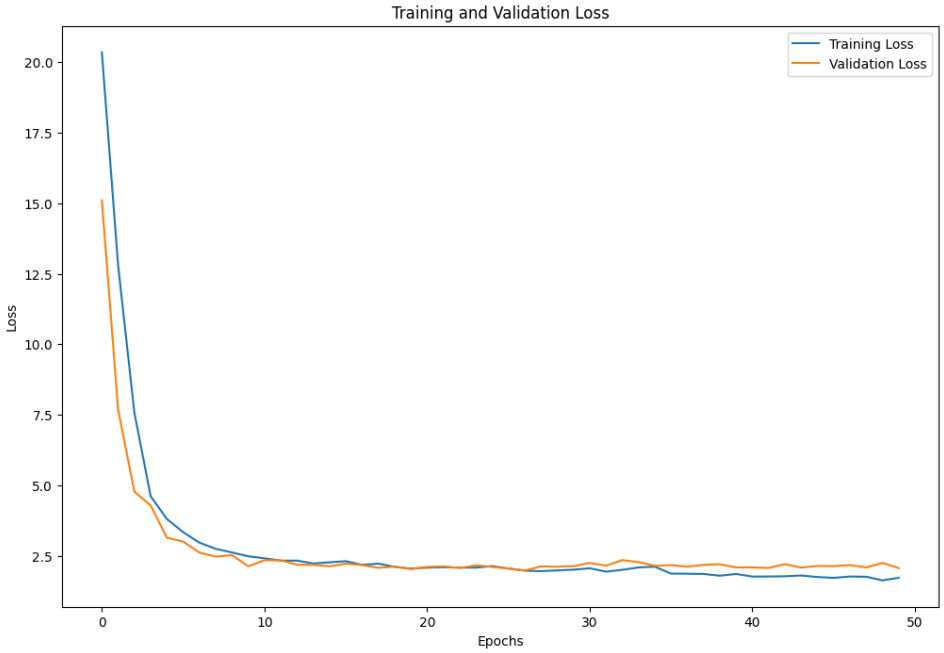
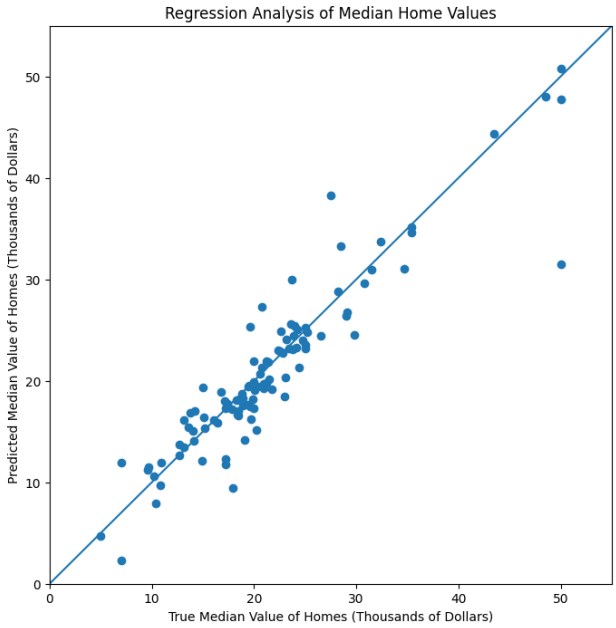


Figure 2

Predictions and true median values of owner-occupied homes



Part 2: Classification Problem

The results from experimenting with different number of hidden layers, units, learning rates, and epochs in the deep neural network architecture for classification are summarized in the tables that follow. From Table 8 it can be seen that the performance of the model remains consistent with the addition of hidden layers. Each layer that is added contains the same number of units as the baseline model.

Table 8

Exploration of the number of hidden layers

Performance Metric	Baseline (1)	2	3	4
Loss	0.27	0.28	0.25	0.29
Accuracy	0.89	0.88	0.90	0.89
Precision	0.87	0.91	0.87	0.83
Recall	0.85	0.77	0.87	0.90

Table 9 summarizes the results from experimenting with units in the hidden layer. A reduction in units results in greater loss, and so does the addition of too many units. The baseline model has the optimal performance as the recall and accuracy worsens with any other unit value.

Table 9

Exploration of units

Performance Metric	10	Baseline(64)	100	250
Loss	0.30	0.27	0.27	0.32
Accuracy	0.92	0.89	0.87	0.88
Precision	0.94	0.87	0.84	1.00
Recall	0.84	0.85	0.82	0.69

Table 10 summarizes the results from experimenting with different learning rates. This parameter has the most influence on the model's performance with the baseline model performing the best and smaller learning rates obtaining poorer performances.

Table 10

Exploration of learning rates

Performance Metric	0.1	Baseline(0.01)	0.001	0.0001
Loss	0.29	0.27	0.45	0.60
Accuracy	0.91	0.89	0.83	0.74
Precision	0.97	0.87	0.77	0.88
Recall	0.79	0.85	0.79	0.36

Table 11 summarizes the results from experimenting with different epochs. The models have similar performances for accuracy, precision, recall regardless of the epochs used; however, the loss grew substantially once 500 epochs are used.

Table 11

Exploration of epochs

Performance Metric	Baseline(10)	50	100	500
Loss	0.27	0.21	0.21	0.66
Accuracy	0.89	0.91	0.91	0.91
Precision	0.87	0.92	0.88	0.86
Recall	0.85	0.85	0.90	0.92

The final model uses the findings from the hidden layer, units, learning rate, and epochs exploration to optimize the performance. The model consists of 2 layers with 100 and 50 units respectively, a learning rate of 0.01, and 50 epochs. The final model performance is compared with the baseline model performance in Table 12 and shows that they obtain similar results. Figure 3 compares the training loss with the validation loss from the final model of owner-occupied homes.

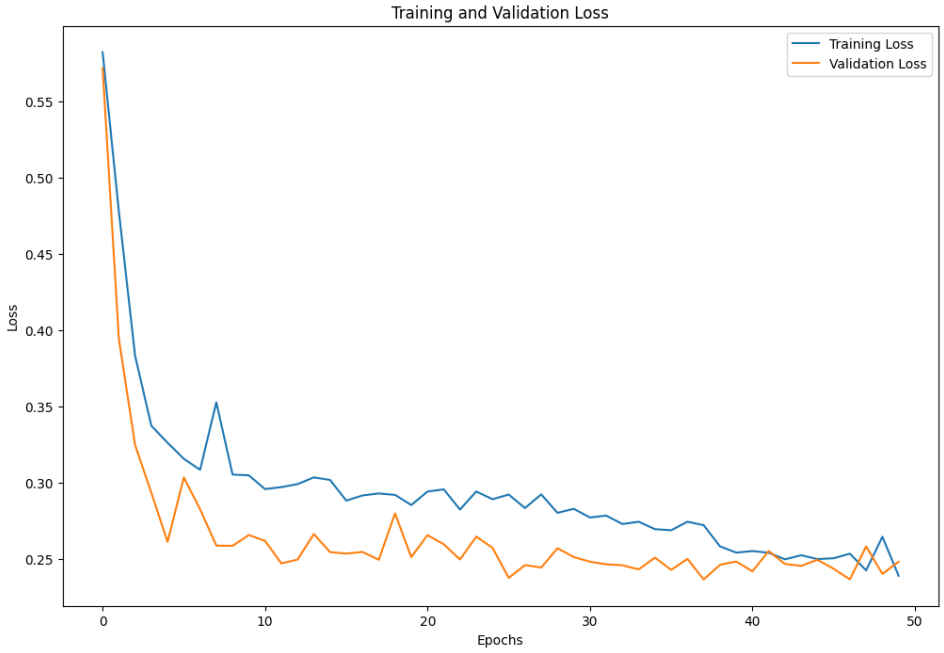
Table 12

Final model performance metrics on test dataset

Metric	Baseline	Final Model
Loss	0.27	0.23
Accuracy	0.89	0.91
Precision	0.87	0.94
Recall	0.85	0.82

Figure 1

Training and validation loss comparison



Conclusion and Future Work

Both the regression and classification models maintain similar results throughout the hyperparameter tuning, indicating that the techniques require little to no domain knowledge and parameter tuning to obtain a good performance. The parameters explored in this analysis are a few of the ones available. Other areas that can be further explored are the optimizer, and batch sizes. The Adam optimizer is used for all the model exploration; however, other functions such as AdamW, Adafactor, or Lion have the possibility of achieving improved results. Similarly, the default batch size of 32 is maintained throughout the analysis; however, modifying this value can have an influence on the model performance.

References

TensorFlow. (2023, August 9). Basic regression: Predict fuel efficiency. https://www.tensorflow.org/tutorials/keras/regression#regression_using_a_dnn_and_multiple_inputs