Tensorflow & Keras NN

Dexter Corley

Argyros School of Business & Economics

&

Fowler School of Engineering

CPSC 393 - 01

Dr. Nicholas LaHaye

March 7, 2023

Abstract

The goal of this assignment is to re-do the textbook examples and construct neural networks for datasets we are interested in doing. This report will outline the steps taken for each neural network and include relevant visual aids. The outside-the-book neural networks were centered around a binary classification model and regression model.

Introduction

A Neural Network is a complex series of algorithms that can be plainly explained by having an activation function acting on values with weights associated, and stacked layers that end with an output goal of minimizing the loss function. For the three introductory examples, the textbook outlined a binary classification model, multiclass classification model, and a regression model. In order as stated the topics are IMDB movie rating sentiment classification, newswire multiclass classification, and housing price regression. Additionally, the further experiments encouraged altering each model's parameters to see differences in the results. On top of the textbook models, additional practice was required, and regression and binary classification datasets were chosen. The regression dataset included columns for predicting a car's miles per gallon. The binary classification contained columns to predict whether someone would default on their card. Both datasets had neural network models built for them which will be described in greater depth next section.

Body

1.1 IMDB Binary Classification

The IMDB classification problem laid out in the textbook covers the step-by-step process for building a binary classification neural network using IMDB ratings data. The end goal of the model is to predict whether the review was positive or negative. The words making up each review were categorized into numbers corresponding to the specific word. One of the early parameters capped these words to the top 10000 most popular reducing the computational load and only 10000 rows passed through for the training and testing set. Additionally, rmsprop

was used as the optimizer and binary_crossentropy the loss function ending with a sigmoid activation function on the output node.

The results for this model were good as it rose to an 87% accuracy for the testing set. The graphs are linked in the performance experiments section.

1.2 Newswire Multiclass Classification

The newswire database was like the IMDB classification as they are both classification models. However, the Reuters dataset needs to be trained with a multiclass classification model that has an end goal of categorizing the records into different topics published. The last model contained only 16 units whereas this model started with 64 to ensure information bottlenecks wouldn't occur. The model used a relu activation function and ended with softmax. The optimizer used was rmsprop and the loss function categorical_crossentropy.

The first model become overfit and required a re-training. The new model came included more epochs and ended up reaching an 80% accuracy. This is a good result and one found even with a rougher first model.

1.3 Boston Housing Regression

The third textbook problem addressed was the Boston housing regression neural network. The dataset containing information aimed to predict the final price of the house. As the price of the house is a continuous variable, the model needs to be a regression function that can take in given data and weigh it in effort of predicting the price of the house. One of the first necessary tasks with this model is to normalize the data. Without this, the model would see data for each field with completely different ranges. Next, when making a model with little training data,

reducing the number of nodes decreases the likelihood of overfitting, in this case the model has two hidden layers with 64 units. The activation function for these two being relu and the loss being mean squared error. Additionally, the model was paired with 3-fold cross validation which splits the data into three cross partitions and takes the average 3 validation scores for the result. Eventually, before the model began to overfit reached a point where the mean absolute error was 2.55. In the context of this problem, the model being off in predictions by an average of $2,550.

2. Credit Card Default Binary Classification

Transitioning into new datasets for exploratory purposes, the first analyzed was a dataset about credit card defaults and information to predict what people are most likely to default. This dataset was found from the UCI Machine Learning library and contained un-normalized data which had to be scaled to ensure the model would accurately be able to predict the results. The activation function for the input and two hidden layers was relu and the output sigmoid. The loss function used was binary cross entropy and the model underwent 50 epochs with a validation loss early stop monitor to prevent overfitting.

The results for this model ended up being 80% accurate.

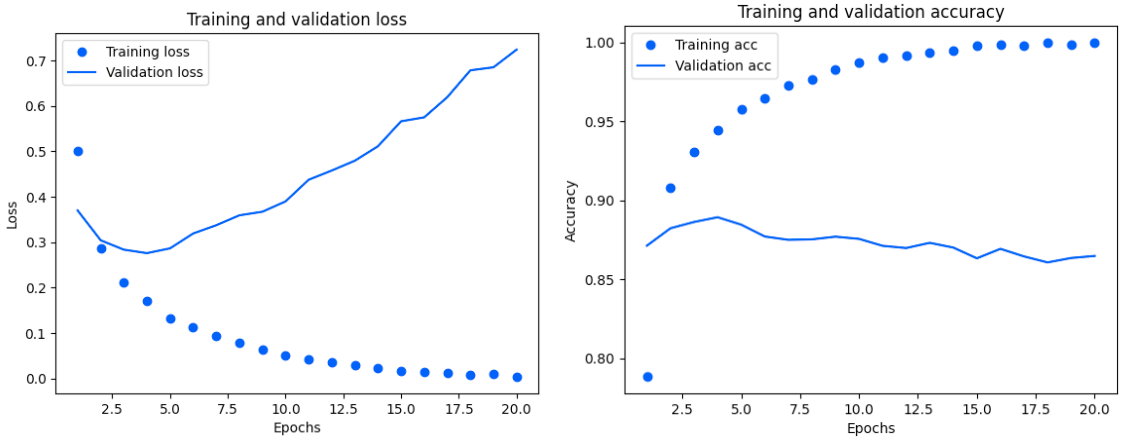
2.1 Miles per Gallon Regression

The miles per gallon dataset contained all numeric fields that provide useful information in predicting mpg such as horsepower, origin, cylinders, weight, and model year. As each column didn't have the same scales, each were scaled using standardscaler. Another thing taken care of in this dataset was parsing out non-numeric values from the horsepower column. Once the data preparation was complete, the model was cross validated using k-fold crossvalidation and the

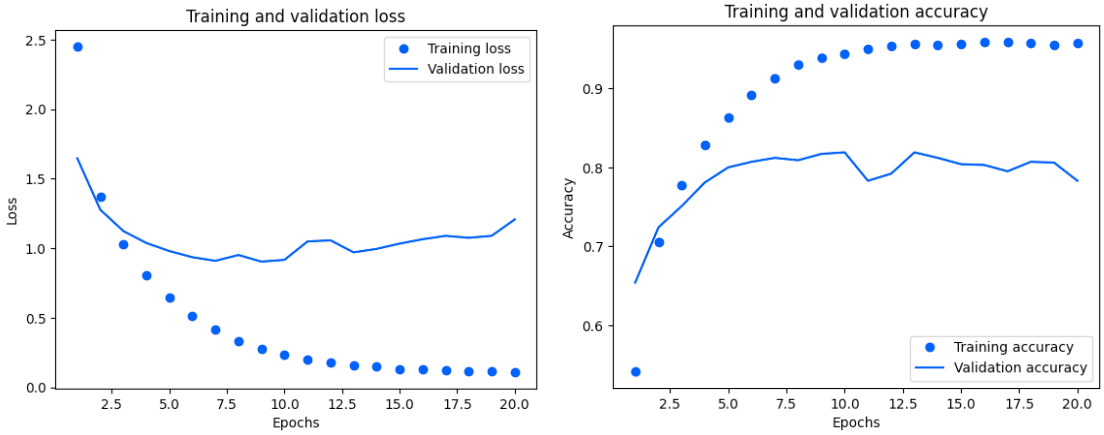
resulting mean absolute error score was 6.26. In the context of this problem saying that on average the prediction was off by an average of 6.26 gallons.

Performance Experiments

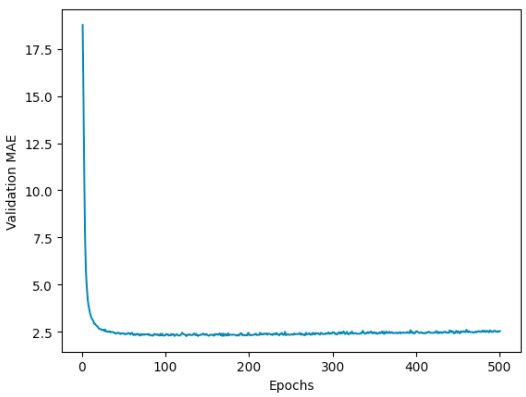
1.1 Binary Classification IMDB training and validation loss and accuracy

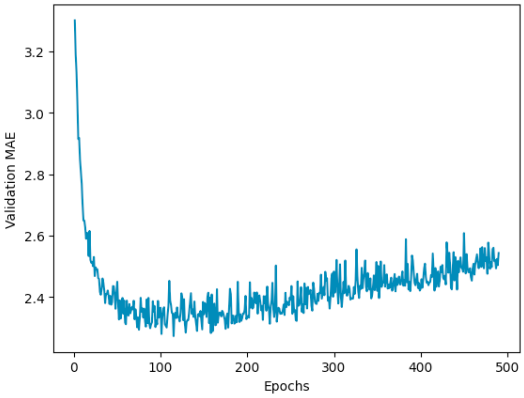


1.2 Multiclass Classification Reuters newswire training and validation loss & accuracy



1.3 Regression Boston Housing data validation mean average error per epoch

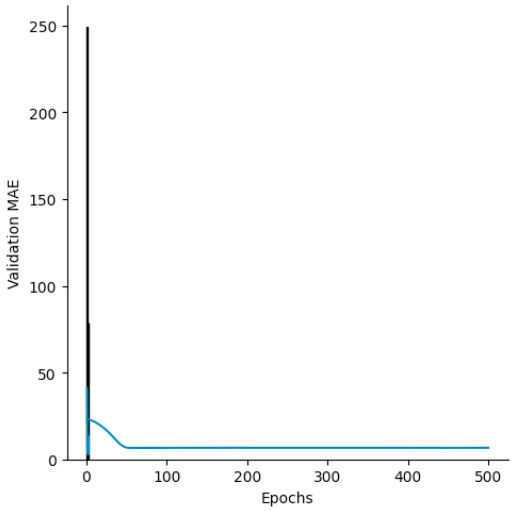


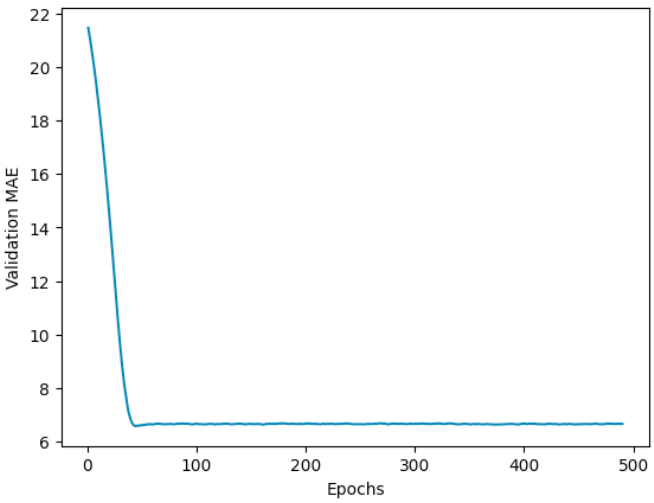


2.0 Credit Card Default Binary Classification

```
[[6691  128]
 [1614  366]]

              precision    recall  f1-score   support

           0       0.81      0.98      0.88      6819
           1       0.74      0.18      0.30      1980

    accuracy                           0.80      8799
   macro avg       0.77      0.58      0.59      8799
weighted avg       0.79      0.80      0.75      8799
```

2.1 Miles Per Gallon Regression





Conclusions

In conclusion, the neural networks that were created during this project all were a little different. The binary and multiclass classification resulted well; the parameters were dialed in for them. As for the regression neural networks, they turned out well but didn't exceed expectations and had mediocre results. In the future larger datasets would benefit a model to give it the opportunity for more training time.

Citations

Kaggle. (n.d.). Default of credit card clients dataset. Retrieved from https://www.kaggle.com/datasets/mariosfish/default-of-credit-card-clients.

Kaggle. (n.d.). Auto MPG Dataset [Mobile application software]. Retrieved from https://www.kaggle.com/datasets/uciml/autompg-dataset