Aaron Shabanian

Assignment 2

Machine Learning

Abstract

In this project, I worked on three different neural network examples, a binary classification model, a multicategorical classification model, and a regression model. I made the binary classification model with a credit default dataset and the multicategorical classification and regression model with an albacore dataset.

Background

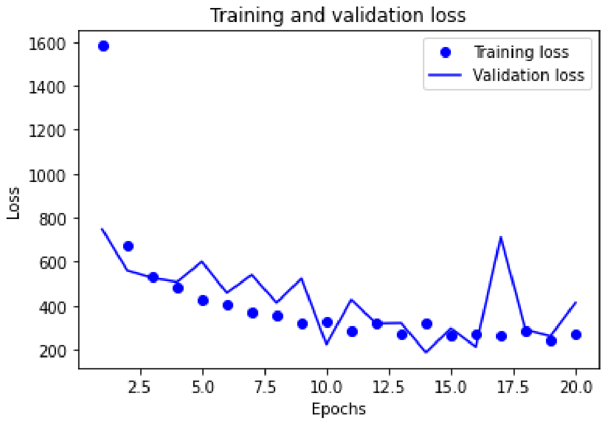
Artificial neural networks are one of the biggest methods used in machine learning. They are mimics of actual neural networks which are the way biological neurons send signals to each other. Artificial neural networks behave similarly with layers of nodes that take in input and then emit an output signal to the next layer depending on what the input was. The final layer will then output the final result from the neural networks. Keras and Tensorflow are currently one of the most accessible ways for people to create their own neural networks with python.

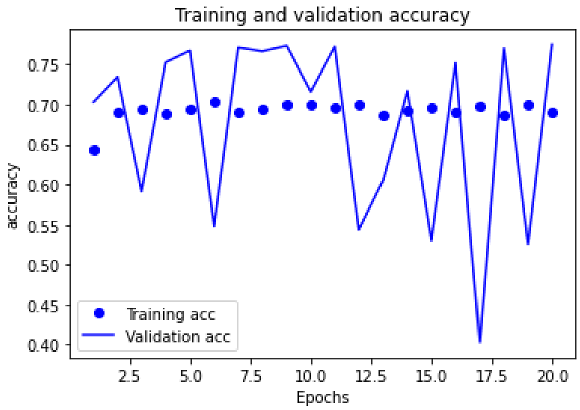
Findings

One of the biggest challenges when working on all of these models was preparing the data for the neural network. Before feeding the data into the neural network, I had to split the data into a train and test set and then set those into numpy arrays that are friendly for neural networks

Binary Classification model

In this model, I used credit data to determine if someone would default on their payment next month. I used 2 hidden layers in this model and then trained it with 1-20 epochs.





In this, I was able to see that the loss and accuracy after 4 epochs on this model has diminishing returns. I ran the same thing again with only 4 epochs and was able to get an accuracy of 76%. This showed me that more epochs are not always better and can even result in overfitting sometimes.

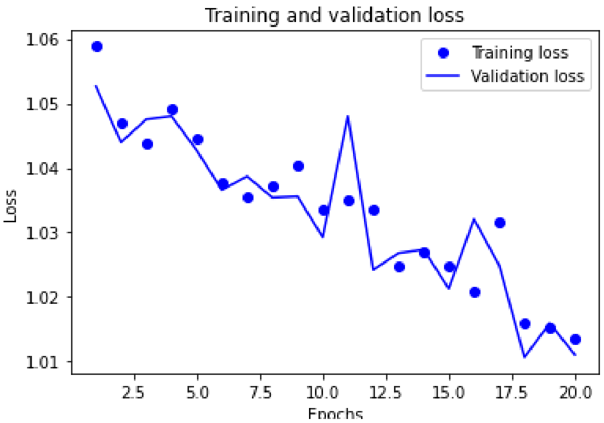
In my further experiments I was able to experiment with using 1 or 3 layers and examine the difference. Using one hidden layer was pretty similar to using 2 if not a little better. Adding a third hidden layer did not help the model at all. This shows that more layers are not always better.

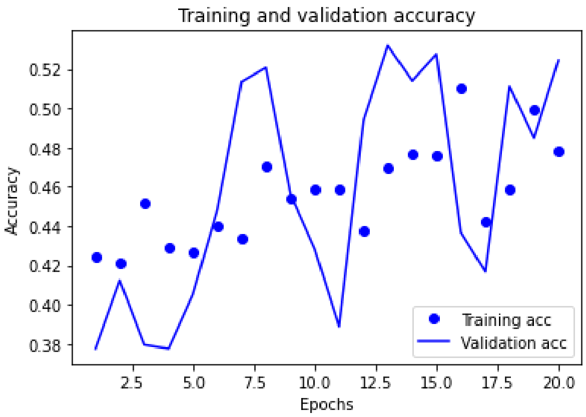
I also used layers with 64 units to examine the difference from the layers with 32 units. I didn’t see much of any difference when using the 64 unit vs the 32 units layers. I guess that could be because this dataset did not need too much dimensionality to begin with compared to the dataset used in the textbook example.

I also used tanh instead of relu function and it didn't change much in terms of accuracy or loss function.

Multiclass classification model

For this model I used a dataset of albacore information. I used a neural network classification method to determine the sex of an albacore as F, M or I based on the albacores characteristics. I had to learn how to categorize them as integers as neural networks need that to be fed in rather than text.

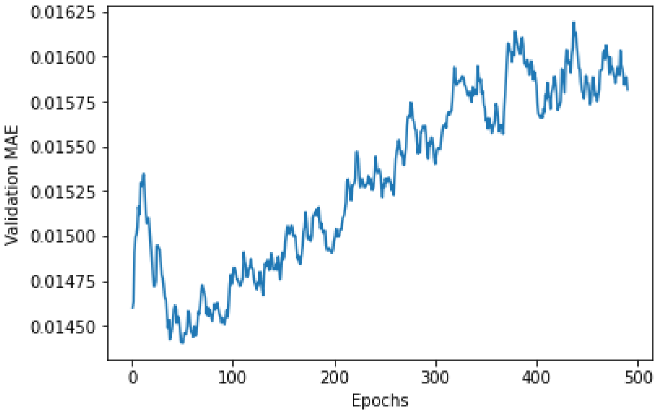




I can immediately see that the accuracy is much better than a random guess which would have been at .33. I was also able to see that the accuracy was bouncing up and down but was still on a general uptrend. The loss was also continuously going down with every new epoch. This model could have done better with more epochs.

Regression Model

I continued using the same dataset as the previous example but this time predicted the length of the albacore. I experimented with the mae over 500 epochs. I also used k-fold validation so I can get a better train test split and more opportunities to test.



I was able to see that the MAE is best at around 50 or 60 epochs and gets worse after that. It continues to show that more Epochs are not always better.

Conclusion

Overall I was able to learn a lot about how to use Keras and better understand feed forward neural networks. I was able to see that more epochs and more layers are not always better. I was also able to see that dimensionality of each neural network can also cause a difference in accuracy in the model.