Neural Network Model Report:

By Rosanna Reza-Giles, Module 21 Data Analysis & Visualization

Overview:

The nonprofit foundation Alphabet Soup wants a tool that can help it select the applicants for funding with the best chance of success in their ventures. Using machine learning and neural networks, I will use the features in the provided dataset to create a binary classifier that can predict whether applications will be successful if funded by Alphabet Soup. The accuracy goal is 75%.

Results:

I was finally able to achieve a 72.58% accuracy after my 3rd attempt which was a marked improvement from my first attempt. I started each model with the same features and targets. I varied the models as follows in the attempt to achieve the 75% accuracy goal. I believe that if I had been successful in my attempt to optimize my model, that I would have probably gotten much closer to the goal.

Deep Learning Challenge Code v1 (first attempt)

Application Type binning less than 500

Classification greater than 1500

1st Hidden Layer 80

2nd Hidden Layer 30

Yielded 5,981 trainable parameters and no non-trainable parameters.

100 Epochs

Resulted in model loss of .74 and accuracy of .63.

A screen shot of a computer code

Description automatically generated with low confidence

Deep Learning Challenge v1-1 (second attempt)

Application Type binning less than 500

Classification greater than 1500

1st Hidden Layer 90

2nd Hidden Layer 40

Yielded 7,641 trainable parameters and no non-trainable parameters.

100 Epochs

Resulted in model loss of .66 and accuracy of .553.

A screen shot of a computer code

Description automatically generated with low confidence

Deep Learning Challenge v2 (third attempt)

Application Type binning less than 500

Classification greater than 100

1st Hidden Layer 80

2nd Hidden Layer 30

Yielded 6,461 trainable parameters and no non-trainable parameters.

100 Epochs

Resulted in model loss of .55 and accuracy of .73.

A screen shot of a computer code

Description automatically generated with low confidence

Data Processing:

**What variables are targets in your model?**

IS\_SUCCESSFUL with be “1” if the non-profit is predicted to be successful and “0 “ zero if un successful.

**What variables are features in your model?**

APPLICATION\_TYPE 17

AFFILIATION 6

CLASSIFICATION 71

USE\_CASE 5

ORGANIZATION 4

STATUS 2

INCOME\_AMT 9

SPECIAL\_CONSIDERATIONS 2

ASK\_AMT 8747

“Name” and “EIN” columns were dropped from the model.

**What variables should be removed from the input data because they are neither targets nor features?**

“EIN” and “Name” columns were removed from the input data because they are believed to have no quantifiable bearing on the data set.

**Compiling training and evaluating the model:**

The model was compiled with a “binar\_crossentropy” for loss, using an “adam” optimizer and metrics that enforced “accuracy”, this was the same across all of the models. Using Tensorflow library to build the learning model.

**How many neurons, layers, and activation functions did you select for your neural network model and why?**

The Model had 3 layers including the output layer and 2 hidden layers, 100 epochs. My first and last model had 2 hidden layers 80 & 30 while the second model had 2 layers 90 & 40. Interestingly my second model showed a small improvement in loss from .74 to .66 while there was a clear loss of accuracy.

**Were you able to achieve the target model’s performance?**

The target model’s accuracy performance had a goal of 75%, the highest I was able to achieve was 73%. I believe that with a few more adjustments and possibly a few mor hidden layers I would be able to achieve that goal.

**What steps did you take in your attempts to increase model performance?**

I realize that attempting to be methodical, in my attempt so improve performance, defined as negligible loss as close to zero as possible, while endeavoring to increase accuracy as close to 100% as possible. I was not sure which would be the key components, so I incrementally adjusted the binning for the “Application\_Type” and for the “Classification”. I also adjusted the number of layers to see which combinations would give me the best results. It was the 3rd attempt where the adjustments were as follows:

Application Type binning less than 500

Classification greater than 100

1st Hidden Layer 80

2nd Hidden Layer 30

Summary:

I think that I had the right approach to adjusting incrementally, if I had more time, I would have readjusted each the number of hidden layers to 3 and then changed the binning quantities into smaller batches. I did attempt to create an optimization model, but I had trouble getting it to work. I also wonder if reclassifying the “Name” column instead of completely eliminating it would have helped me achieve the accuracy highest accuracy score if those other adjustments had not worked as intended.