Alphabet Soup Neuro Network Assignment Report

# Overview

The purpose of this analysis was to aid Alphabet Soup in their selection of applicants for funding in order to increase their chance of successful outcomes. In order to achieve this we analysed data from 34,000 organisations who have previously been given funding by Alphabet Soup.

# Results

Bulletpoints and images:

## Data preprocessing:

* The variable from the dataset labelled “IS\_SUCCESSFUL” was identified as being the target variable
* The following variables were identified as being the features for the dataset:
* APPLICATION\_TYPE
* AFFILIATION
* CLASSIFICATION
* USE\_CASE
* ORGANIZATION
* STATUS
* SPECIAL\_CONSIDERATIONS
* ASK\_AMT
* INCOME\_ AMT
* The variables NAME and EIN were dropped from the dataset as they were not identified as either a feature or a target variable.

## Compiling, Training and Evaluation of Model

* The original model had 3 Neuron: the Input Neurons; one Hidden Layer; the Output Neuron. This is because it would not take too much computing power to run the model with 3 neurons.
* The result of this model was unsuccessful to achieve the 75% accuracy requested.

## Optimisation

Steps taken to optimize the results:

1. Remove the SPECIAL\_CONSIDERATIONS feature as this was heavily weighted to N. I believed having so few Y values it did not give the model much to learn from.
2. Change the bucketing to reduce the number of values in the APPLICATION\_TYPE feature. I believed this would help reduce the noise in the data.
3. Classify the ASK\_AMT into bins to reduce the number of values in the feature.
4. Categorise the ‘rare’ values in the AFFILIATION feature to Other to reduce the noise.
5. Categorise the ‘rare’ values in the ORGANIZATION feature to Other to reduce the noise.

At this point I ran the revised model to see the results.

### Model Revision 1

Text

Description automatically generated

A picture containing shape

Description automatically generated

The Accuracy had decreased with the changes made so A second revision was needed:

### Model Revision 2

The changes I made for model revision 2 were as follows:

1. Add another layer
2. Increase Epochs to 200

Text

Description automatically generated

A picture containing graphical user interface

Description automatically generated

The Accuracy in this new model had improved but not enough to reach the required 75%.

### Model Revision 3

I then made the following changes to the compile method:

1. Change Optimiser to Nadam
2. Added parameter steps\_per\_execution at 20

Text

Description automatically generated

A picture containing text

Description automatically generated

These changes reduced the Accuracy of the model to a similar accuracy as the original model.

### Model Revision 4

For this revision I reverted to the optimiser adam and added the following changes:

1. Increase the number of nodes in each layer to 8 (double)

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated with medium confidence

The accuracy on this model still did not meet the required 75% and was again weaker than the previous model.

### Model Revision 5

For this model I made the following changes:

1. Switch the activation parameters so that the output was Relu and the others were Sigmoid.

Text

Description automatically generated

Shape

Description automatically generated with low confidence

The accuracy on this model still did not meet the required 75% and was not the strongest, but did outperform model 4.

# Summary

In summary, I was unable to get to the required 75% with this model. The closest to the target was model revision 2 which scored an accuracy of 73.1%.