**Charity Funding Neural Network Write-up**

**Overview**

The purpose of this analysis was to predict whether Charity applicants were successful or not. The company. The nonprofit foundation Alphabet Soup wants to use this information to select the best applicants with the best chance for success when funded.

**Results**

Data Processing

* When cleaning data we removed the “EIN” and “NAME” columns since they don’t relate to model.
* The variables remaining for the model are the following:

**APPLICATION\_TYPE**—Alphabet Soup application type

**AFFILIATION**—Affiliated sector of industry

**CLASSIFICATION**—Government organization classification

**USE\_CASE**—Use case for funding

**ORGANIZATION**—Organization type

**STATUS**—Active status

**INCOME\_AMT**—Income classification

**SPECIAL\_CONSIDERATIONS**—Special consideration for application

**ASK\_AMT**—Funding amount requested

**IS\_SUCCESSFUL**—Was the money used effectively

* The Dependent Variable that was chosen was “IS\_SUCCESSFUL” since this best presents the success rate of each applicant.
* The variables that were binned were the “APPLICATION TYPE” and “CLASSIFICATION” binning amount to reduce the population including the lower value amounts as OTHER.

Compiling, Training, and Evaluating the Model

* I used 2 hidden layers
* The neurons were 80 for Layer 1, and 30 for Layer 2. (based on suggested result)
* I used Relu and Sigmoid activation functions
* 100 epochs

268/268 - 0s - loss: 0.5621 - accuracy: 0.7301

Loss: 0.562099814414978, Accuracy: 0.7301457524299622

Attempts 1, 2, 3, 4 for 75% accuracy

Attempt 1

I reduced the data in columns “APPLICATION\_TYPE” to 6 items down from 10.

I increased the number of epochs to 200.

268/268 - 0s - loss: 0.5702 - accuracy: 0.7300

Loss: 0.5702178478240967, Accuracy: 0.7300291657447815

Attempt 2

I reduced the data in columns “APPLICATION\_TYPE” and “CLASSIFICATION” to 4 items down from 6.

I decreased the number of epochs to 50.

268/268 - 0s - loss: 0.5666 - accuracy: 0.7199

Loss: 0.5665838718414307, Accuracy: 0.719883382320404

Attempt 3

Used column processing as Attempt 2.

Added a 3rd hidden layer of 40 neurons.

I increased the number of epochs to 100.

268/268 - 0s - loss: 0.5705 - accuracy: 0.7198

Loss: 0.570472002029419, Accuracy: 0.7197667360305786

Attempt 4

I increased the data in columns “APPLICATION\_TYPE” and “CLASSIFICATION” to 9 and 7 items down respectively from 4.

Added a 3rd hidden layer of 15 neurons

Number of epochs was 100.

268/268 - 0s - loss: 0.5619 - accuracy: 0.7287

Loss: 0.5619065165519714, Accuracy: 0.7287463545799255

**Summary**

In summary, I tried to change the model for the better to get to 75%. I ended up making my model less accurate. I tried to reduce the data in the variable columns and it made the model less accurate. My model stayed around 72% to 73%. I think if there was more data columns added that showed more relation to the “IS\_SUCCESSFUL” variable we could this model to make better success predictions.