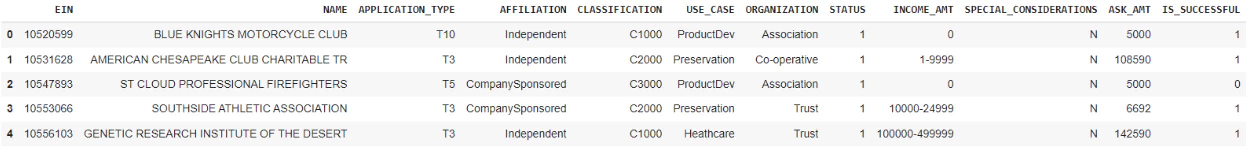
Ricardo G. Mora, Jr.
February 19, 2022

USING DEEP LEARNING TO PREDICT THE SUCCESS OR FAILURE OF ALPHABET SOUP FUNDING

OVERVIEW

The purpose of this assignment was to examine the accuracy of a TensorFlow Keras model in predicting whether funding will be used effectively by an applicant to the Alphabet Soup foundation. Historical data contained in a CSV file was used to train and evaluate the model. Standard preprocessing techniques were employed such as scaling, one-hot encoding, removal of irrelevant variables, and binning of categorical items that don't appear often. After the initial training and testing of the model, further preprocessing of the data was utilized along with several optimization techniques to try to increase the accuracy of the model.

Sample data from the CSV file loaded into a data frame:



	EIN	NAME	APPLICATION_TYPE	AFFILIATION	CLASSIFICATION	USE_CASE	ORGANIZATION	STATUS	INCOME_AMT	SPECIAL_CONSIDERATIONS	ASK_AMT	IS_SUCCESSFUL
0	10520599	BLUE KNIGHTS MOTORCYCLE CLUB	T10	Independent	C1000	ProductDev	Association	1	0	N	5000	1
1	10531628	AMERICAN CHESAPEAKE CLUB CHARITABLE TR	T3	Independent	C2000	Preservation	Co-operative	1	1-9999	N	108590	1
2	10547893	ST CLOUD PROFESSIONAL FIREFIGHTERS	T5	CompanySponsored	C3000	ProductDev	Association	1	0	N	5000	0
3	10553066	SOUTHSIDE ATHLETIC ASSOCIATION	T3	CompanySponsored	C2000	Preservation	Trust	1	10000-24999	N	6692	1
4	10556103	GENETIC RESEARCH INSTITUTE OF THE DESERT	T3	Independent	C1000	Heathcare	Trust	1	100000-499999	N	142590	1

MODEL SETUP AND RESULTS (INITIAL ATTEMPT)

The following list summarizes the preprocessing parameters, model setup, and model results. This model can be run in the Jupyter Notebook: "AlphabetSoupCharity.ipynb".

- Target Variable: "IS_SUCCESSFUL"
- Feature Variables: "APPLICATION_TYPE", "AFFILIATION", "CLASSIFICATION", "USE_CASE", "ORGANIZATION", "STATUS", "INCOME_AMT", "SPECIAL_CONSIDERATIONS", "ASK_AMT"
- Unimportant Variables: "EIN", "NAME"
- Items used less than 500 times in "APPLICATION_TYPE" binned together as "OTHER"
- Items used less than 1000 times in "CLASSIFICATION" binned together as "OTHER"
- Layer 1: 80 neurons, "relu" activation function
- Layer 2: 30 neurons, "relu" activation function
- Output Layer: 1 neuron, "sigmoid" activation function
- Number of Epochs: 100
- Model Data Loss: 56.2%
- Model Predictive Accuracy: 72.4% (Less than the desired 75%)

MODEL SETUP AND RESULTS (OPTIMIZATION ATTEMPT)

The following list summarizes the preprocessing parameters, model setup, and model results. This model can be run in the Jupyter Notebook: "AlphabetSoupCharity_Optimization.ipynb".

- Target Variable: "IS_SUCCESSFUL"
- Feature Variables: "NAME", "APPLICATION_TYPE", "AFFILIATION", "CLASSIFICATION", "USE_CASE", "ORGANIZATION", "INCOME_AMT", "ASK_AMT"
- Unimportant Variables: "EIN", "STATUS", "SPECIAL_CONSIDERATIONS"
- Items used less than 5 times in "NAME" binned together as "OTHER"
- Items used less than 500 times in "APPLICATION_TYPE" binned together as "OTHER"
- Items used less than 1000 times in "CLASSIFICATION" binned together as "OTHER"
- Layer 1: 100 neurons, "relu" activation function
- Layer 2: 30 neurons, "sigmoid" activation function
- Layer 3: 10 neurons, "sigmoid" activation function
- Output Layer: 1 neuron, "sigmoid" activation function
- Number of Epochs: 100
- Model Data Loss: 44.8%
- Model Predictive Accuracy: 79.4% (Greater than the desired 75%)

SUMMARY

The initial model failed to achieve the desired 75% predictive accuracy rate. However, it was possible to boost the accuracy up to close to 80% and simultaneously reduce the data loss to close to 45% by increasing the number of neurons in the first layer, adding an additional layer, and changing the activation function in a couple of the layers. Also, upon closer examination of the input data, the "NAME" variable was re-included since there were many applicants that had multiple applications which might indicate whether they have a history of successfully using their funds. Also, the "STATUS" and "SPECIAL_CONSIDERATIONS" variables were removed since each was dominated by a single value and did not contribute any useful meaning to the model. As an additional comparison, a Random Forest Classifier was used with the same preprocessed data as the optimized TensorFlow model, and it achieved a similar (but slightly less) accuracy rate of 77.9%. So in conclusion, the TensorFlow model performed reasonably well in predicting the funding effectiveness for Alphabet Soup's applicants, but only after careful preprocessing of the input data and careful optimization of the model. And its performance was only slightly better than other binary classifiers like Random Forest.