Deep Learning: Charity Funding Predictor Report

Overview

For this project machine learning and neural networks were used to create a binary classifier that could predict whether an applicant would be successful if it was funded by Alphabet Soup. The project used a CSV that held more than 34,000 organizations that received funding from Alphabet Soup over the years. The dataset held columns that captured the metadata about each organization.

Processing the Data

The first step for processing the data was to remove the irrelevant columns in the dataset including ‘EIN’ and ‘NAME’. As the two columns were not numeric.

Graphical user interface, application

Description automatically generated

Image above shows the data set that includes the features and targets

The target for the dataset was the ‘IS\_SUCCESSFUL’ column, the rest of the columns were the features. If the applicant was successful a ‘1’ was shown, and unsuccessful a ‘0’.

Text

Description automatically generated

*Image above shows the unique values in the dataset*

‘APPLICATION\_TYPE’ and ‘CLASSIFICATION’ had more than 10 unique values within their respective column. When a dataset has a large number of unique values, it could make noise within the dataset. To reduce the possibility of noise within this dataset binning was used. Binning the groups replaces values that were contained into small intervals. This is one way that can help improve the accuracy of the models.

Before Binning Text, table

Description automatically generated with medium confidenceTable

Description automatically generated After Binning

Compile, Train, and Evaluate the Model

After preprocessing the data, the following step was to compile, train, and evaluate the model.

Text

Description automatically generated with medium confidence

*Image above shows the model summary*

The first model had three layers, two that were hidden, and one that was the output layer. For the first hidden layer there were seven units with a relu activation, for the hidden layer there were sixteen units with a relu activation. For the output layer there was one unit and a sigmoid activation. Units stand for the dimensionality of the output, and the activation stands for the function used.



*Image above shows the model performance*

The accuracy of the model was seventy-two percent, and does not meet the target of seventy-five percent accuracy, also the model had a high loss at fifty-five. The next step was to optimize the model by adjusting the input data, adding more neurons to hidden layers, adding more hidden layers, and adding or reducing the number of epochs in the training regimen.

Summary

Overall, after adjusting the input data, adding neurons to hidden layers, and adding layers; the models did change as follows:



*Image above shows the model performance- optimization one*



*Image above shows the model performance- optimization two*



*Image above shows the model performance- optimization two*

One suggestion is to complete more research on how to increase accuracy and decrease loss in models. As the changes made within the optimizations dropped the accuracies and increased the losses.