Deep Learning: Charity Fund Predictor

Deep learning and neural networks were used in this project to predict whether funding candidates would be successful if funded by Alphabet Soup using information from over 34,000 organizations.

Data Preprocessing

First the dataset was cleansed of any unnecessary information by removing the EIN and NAME columns from the model. Then the remaining columns were analyzed to determine an acceptable number of categories within those columns. Due to the large number of unique values in the CLASSIFICATION and APPLICATION\_TYPE columns, those categories containing less than 100 data points were combined into one “Other” column. The data was then split into the features and target data. The target is “IS\_SUCCESSFUL” where 1 is successful after funding and 0 is unsuccessful after funding. The remaining columns are the features for this dataset with the categorical variables being encoded with pd.get\_dummies(). The data was then split into training and testing sets to be used by our neural network.

Compiling, Training, and Evaluating the Model

In the first neural network, two hidden layers were used with the first layer containing 2 nodes and the second containing 8 nodes. This generated 133 parameters and after 100 epochs received 71.5% accuracy which was below the goal of 75%.

Text

Description automatically generated

Optimization of the Model

In the next neural network, the number of nodes in each layer was flipped so the first layer contained 8 nodes and the second contained 2. This generated 421 parameters and after 100 epochs marginally improved to 72.9% accuracy.

Text

Description automatically generated

Next, another hidden layer was added to the model with the number of nodes being 8, 6, 4. This generated 487 parameters. The number of epochs was scaled down to 75 since the accuracy didn’t appear to be improving with more epochs in the previous models. This model received 72.7% accuracy.

Text

Description automatically generated

Finally, a model was created with 2 hidden layers, since 3 hidden layers did not improve accuracy. The number of nodes in each layer was increased to 14 and 8. The second hidden layer’s activation was also changed to tanh to determine if this activation function would be better equipped to classify the data. This created 851 parameters and epochs reduced to 50, this model received 72.9% accuracy.

Text

Description automatically generated

The models were unable to achieve the target model performance. Going forward, I would increase the number of epochs back to 100 in model 3 since the accuracy seemed to be steadily increasing. Further examination of the data may help choose an activation function that best matches the structure of the data. I would consider adding a third hidden layer with that activation function to further predict and classify incoming information.

Graphical user interface

Description automatically generated with medium confidence