Overview

This is a deep learning model using tensorflow to predict the success of charities using multiple categorical data such as affiliation, use case, and organization.

Results

Data Preprocessing

(What variables are the targets for your model?)

* This section defines the IS\_SUCCESSFUL column to be the target variable.

(What variables are the features for your model?)

* Since EIN and NAME were dropped, and IS\_SECCESSFUL is the target, the rest of the columns are features.

(What variables should be removed from the input data because they are neither targets nor features?)

* Name and EIN was removed because each value is unique. This will skew the model accuracy. Name was dropped because the goal was not to find which charity was successful, it was to find what makes that charity successful.

Compiling, training, and evaluating the model

(How many neurons, layers, and activation functions did you select for y9our neural network model, and why?)

* I chose the number of layers and neurons to be the same as what the starter code output indicated. It seemed to perform in a similar way, so I was satisfied. I can add that I think these values to be a decent number for a simple neural net such as this one, that wont underfit, or overfit, or take too long to compile. Relu was chosen for the first two since most of the data is categorical. I figured a nonlinear activation would be best. Sigmoid was the last layer because the output classification is binary. Though my specific knowledge of how these perform is limited.

(Were you able to achieve the target model performance?)

* No. In the end I could not get the model to perform any better than 72.5-73%

(What steps did you take in your attempts to increase model performance?)

* I made a few changes but nothing seemed to improve the performance. I tried increasing the layers and epochs, removing classification, application type, status, and special considerations. Since the accuracy never improved, I put application type and classification back in. I removed all but the first hidden layer, and lowered the epochs to 50. This resulted in a model with the same accuracy and loss with less than half the process time.