Alphabet Soup Foundation Success

Overview

Dataset containing more than 34k organizations that have received funding from Alphabet Soup. Here I am using the provided dataset to create a binary classifier that helps to predict whether applicants will be successful if funded by Alphabet Soup.

Results

Data Preprocessing

- The target variables for my model are the columns showing if the money was used successfully and the classification column that shows the government classification.
- The remaining columns (other than the removed columns) are features for the model.
- The columns removed from the model were the EIN and NAME columns – these columns are strictly used for identification of the applicants.

Compiling, Training, and Evaluating the Model

For my initial evaluation with the model, I used 8 nodes in the first layer and 5 nodes in the second layer. I used 'relu' for both hidden layers and 'sigmoid' for the output layer in my first model. I chose the number of layers and nodes randomly to give myself a starting point. I stuck with 'relu' and 'sigmoid' because the data that was being used would not give negative values – so using 'relu' first to train the model and 'sigmoid' to finish up ('sigmoid' values are 0 -1) seemed to be a good choice.

Unfortunately, I was not able to achieve a target performance of 75%. Some different steps I took to increase model performance included changing cutoff values for the first model, adding more nodes (12, and 7) in the second model, increasing Epochs to 250 for the third model, and the fourth time I initiated the model I added another hidden layer and increased the nodes to 22, 12, and 7 for hidden layers 1, 2, and 3, respectively.

Summary

Overall, I was able to achieve a maximum of 73% performance with my model and my attempts to increase its performance. Some different approaches I would make if I used this dataset again might be to remove more data from the initial dataset, increase the number of Epochs to a larger number (500 or 100), and potentially use another layer with more nodes. I believe that with the dataset 'relu' and 'sigmoid' were the best choices of activation functions.