Alphabet Soup Model Report

Overview:

The main purpose of this project was to create a model that would help select the best candidates for funding that will also have the greatest chance of success. This is imoprtant as funding can often be limited and choosing the organizations with the best chance of success will help make the best use of that funding.

Results:

- Preprocessing
    - The key target of the model was whether or not each entry was successful or not, as see in the "IS_SUCCESSFUL" column in the csv. This was chosen as the target because whether or not a funding venture would be successful or not would be the main deciding factor for choosing which applicant to fund.

    - The features for this model would be most of the other columns of information in the csv like application type, funding amount, and type of organization. All these factors are considered as they could all have some effect on whether a funding venture will be successful.

    - Originally both the "EIN" and "NAME" factors were removed as for the some cases they are unique. However "NAME" was found to often be repeated as the same organizations would apply multiple times for different amounts. In the optimized model "EIN" was removed as it is used for unique ID numbers.

- Compiling, Training, and Evaluating the Model

    - The initial model was built using 2 hidden layers and 1 output layer. The activation functions used were the ReLU with 6 neurons for the hidden layers and a sigmoid function with 1 neuron for the output layer.

    - While the initial model work well enough to gain a 73% accuracy The optimized model worked much better and was able to achieve an 80% accuracy.

- To increase the accuracy and performance of the model, I had to change several things. First I included the “NAMES” column in the model as some organizations would apply multiple times. I also included more unique values for the “CLASSIFICATION” column. Finally, I edited the model itself using the results I got from KerasTuner. I lowered the amount of epochs from 100 to 20, changed the function of the hidden layers from ReLU to tahn and then added another hidden layer. I also changed the number of neurons in the layers from 6 to 6, 21, and 26.

Summary:

Overall I think the model in its current state after optimizing works fairly well. It was able to take several factors from the dataframe and produce a fairly accurate model. However the loss was still at about 50%. I would also recommend trying a linear model as the target can only be one of two options and can be linearly separated. I would also consider removing columns like “STATUS” and “SPECIAL_CASE” as very few applications tend to deviate from the majority so it may not be as needed and can help lighten on resources for running the model.