

# Alphabet Soup - Neural Network Model Analysis

## Overview

The purpose of this analysis is to evaluate the performance of the deep learning model that I created for Alphabet Soup. My first attempt at the model did not achieve target performance levels, so I had to try to optimize my model a couple of times. I will discuss which methods I used to optimize my model.

## Results

### Data Preprocessing

The variable(s) that are the target(s) for my model are the "IS\_SUCCESSFUL" column. The variable(s) that are features for my model are all of the other columns. I did not remove any variables for my model.

### Compiling, Training, and Evaluating the Model

For my neural network model, I chose 80 neurons, three layers (two hidden and one output layer), and two activation functions ("relu" and "sigmoid"). On my first attempt with my model, I was unable to achieve the target model performance. This is evident because my accuracy was only 73.09%, which is slightly below the target of 75%.

In my attempts to increase model performance, I took a few different methods. In my first optimization attempt, I added a third hidden layer and increased the number of epochs to 150. This did not increase performance and actually slightly decreased performance from 73.09% to 73.01%.

In my second optimization attempt, I decided to return the number of hidden layers and epochs to their original values, which was two hidden layers and 100 epochs. After I did that, I decided to change the activation function for the second hidden layer to "relu" instead of "sigmoid" and I reduced the number of neurons used in each layer. This ultimately reduced my accuracy even further to 72.84%.

For my third optimization attempt, I returned to my original number of neurons in each layer and changed my activation functions to "tanh." This third attempt at optimization resulted in an accuracy of 73.18%, which is slightly higher than my original attempt at my model which resulted in a 73.09% accuracy level.

## Summary

Overall, my deep learning model performed fairly. I did not achieve the 75% performance level, but was very close to it at 73%. A different model might solve this classification problem by removing some of the columns from the original provided data. This recommendation might work better than my model because some of the columns in the original dataframe may have been causing confusion for the model.