Module 21 Challenge
Neural Network Model Analysis

Overview

The purpose of this challenge is to utilize the knowledge we have gained on machine learning and neural networks to create a tool for the nonprofit organization Alphabet Soup that would help them select applicants for funding with the best chance of success in their ventures. We received a CSV containing 34,000 organizations that have received funding from Alphabet Soup over the years. We will need to use the features in the provided dataset to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup.

Results

1. Data Preprocessing

   The first necessary step to create the tool for Alphabet Soup is to preprocess the data. The CSV had two columns that needed to be removed, since the values that they contained were neither targets nor features, these were the "EIN" and "NAME" columns. For this CSV, the target variable was the "IS_SUCCESFUL" column and the features were all of the other columns in the dataset.

2. Compiling, Training, and Evaluating the Model

   For the first neural network model I decided to use three layers. One input layer, one hidden layer, and one output layer. The input layer had 10 units or neurons and used a 'relu' activation function. The hidden layer had 8 units and used a 'relu' activation function as well. The output layer had 1 unit and used a 'sigmoid' activation function. With these parameters the model achieved an accuracy of 72.5%, which fails to meet the organization's goal of at least 75%. I made three attempts at optimizing the model but was not able to achieve a better accuracy than 72%. In these attempts, I added more units per layer, and tried using more hidden layers, but manually, nothing seemed to be working at achieving a better accuracy.

Module 21 Challenge
Neural Network Model Analysis

Summary

The neural network model was only able to achieve 72.5% accuracy, with a loss of 55%. I would recommend trying a different model or using the Keras library and it's tuner to find the best hyperparameters for the model and attempt at achieving a higher accuracy. Manually optimizing a neural network is troublesome and time consuming, our best bet would be to make use of the technology and let the algorithm of the Keras tuner try different parameters until the best ones are found. After that, we can use those hyperparameters and test them out with our neural network model.