

Analysis

Overview

The nonprofit foundation Alphabet Soup wants a tool that can help it select the applicants for funding with the best chance of success in their ventures. By using machine learning and neural networks, an algorithm will be created to predict whether applicants will be successful.

A CSV file containing more than 34,000 organizations was read into Pandas.

Preprocess the Data

The data was processed by:

- Drop the `EIN` and `NAME` columns.
- Determine the number of unique values for each column.
- For columns that have more than 10 unique values, determine the number of data points for each unique value.
- Use the number of data points for each unique value to pick a cutoff point to bin "rare" categorical variables together in a new value, `Other`, and then check if the binning was successful.
- Use `pd.get_dummies()` to encode categorical variables.
- Split the preprocessed data into a features array, `X`, and a target array, `y`. Use these arrays and the `train_test_split` function to split the data into training and testing datasets.
- Scale the training and testing features datasets by creating a `StandardScaler` instance, fitting it to the training data, then using the `transform` function

Compile, Train, and Evaluate the Model

- Continue using the file in Google Colab in which you performed the preprocessing steps from the preprocessing step.
- Create a neural network model by assigning the number of input features and nodes for each layer using TensorFlow and Keras.

- Create the first hidden layer and choose an appropriate activation function.
- If necessary, add a second hidden layer with an appropriate activation function.
- Create an output layer with an appropriate activation function.
- Check the structure of the model.
- Compile and train the model.
- Create a callback that saves the model's weights every five epochs.
- Evaluate the model using the test data to determine the loss and accuracy.
- Save and export your results to an HDF5 file. Name the file `AlphabetSoupCharity.h5`.

Results

The attempt resulted in an accuracy score of 72.5%. This indicates that 72.5% of the model's predicted values align with the true values of the dataset. The hyperparameters used were: layers= 2 and epochs=100. Layer 1 had 9 neurons and the 'relu' activation function was used and layer 2 had 18 neurons and the 'relu' activation function was used.

Summary

The model was unable to achieve a target predictive accuracy of 75%. I would consider using another classification model to see if it is better at predicting whether applicants will be successful.