**Using Deep Learning to Select Best Applicants for Funding**Oct 2023 B. Kocurek

A blue robot with arrows and circles

Description automatically generated

**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. Using the provided dataset, machine learning and neural networks were used to create a binary classifier to predict whether applicants will be successful if funded by Alphabet Soup.

**Methods:**

1. The dataset was preprocessed using Pandas and scikit-learn’s StandardScaler.

* The charity\_data.csv file was read into a Pandas DataFrame.
* The column names EIN and NAME were dropped.
* The number of unique values for each column were determined.
* For columns that had more than 10 unique values, the number of data points for each unique value were determined.
* The number of data points for each unique value were used to pick a cutoff point to bin "rare" categorical variables together in a new value, Other, and then check if the binning was successful.
* Pd.get\_dummies was used to encode categorical variables.
* The preprocessed data was split into a features array, X, and a target array, y. These arrays were used and the train\_test\_split function used to split the data into training and testing datasets.
* The training and testing features datasets were scaled by creating a StandardScaler instance, fitting it to the training data, then using the transform function.

1. Compile, Train, and Evaluate the Model

* TensorFlow was used to design a neural network, or deep learning model, to create a binary classification model to predict if an Alphabet Soup-funded organization will be successful based on the features in the dataset. Consideration was given to how many inputs there are before determining the number of neurons and layers in the model.
* The binary classification model was compiled, trained, and evaluated to calculate the model’s loss and accuracy.

**Results:**

* **What variable(s) are the target(s) for your model?**
  + IS\_SUCCESSFUL
* **What variable(s) are the features for your model?**
  + APPLICATION\_TYPE
  + AFFILIATION
  + CLASSIFICATION
  + USE\_CASE
  + ORGANIZATION
  + ASK\_AMT
* **What variable(s) should be removed from the input data because they are neither targets nor features?**
  + STATUS
  + INCOME\_AMT
  + SPECIAL\_CONSIDERATIONS
* **How many neurons, layers, and activation functions did you select for your neural network model, and why?**
  + The initial model had two layers with 9 nodes in layer 1 and 18 nodes in layer 2.
    - The number of nodes used were based on the recommendation to use 2-3 times as many nodes as there are input features.
  + The initial activation function used for the hidden layers was reLU and sigmoid was used for the outer layer.
    - The reLU function was used for the hidden layers since it models positive non-linear data and is a good starting point.
    - The sigmoid function was used for the outer layer since the activation function for the outer layer should be less complex than those used for the hidden layers.
* **Were you able to achieve the target model performance?**
  + No, a target model performance of 75% or more was not achieved.
* **What steps did you take in your attempts to increase model performance?**
  + Each step was done one at a time to determine impact on model performance:
    - Dropped additional columns:
      * STATUS – 99.9% of data was a “1”.
      * INCOME\_AMT – mixed data types with a majority being “0”. Including this data could skew the model if associated with IS\_SUCCESSFUL.
      * SPECIAL\_CONSIDERATIONS – 99.9% of data was “N”
    - Added 3rd Hidden Layer to increase complexity
    - Increased number of nodes
    - Changed activation functions
    - Reduced number of epochs to determine if training was overfitting the data

|  |  |  |
| --- | --- | --- |
| **Model \*** | **Train Accuracy** | **Model Accuracy** |
| 1 |  |  |
| 2  Dropped additional columns |  |  |
| 3  Added 3rd hidden layer |  |  |
| 4  Added nodes |  |  |
| 8  Changed activations |  |  |

**\*See h5 files for additional details**

**Summary:**

Several attempts to optimize the model’s performance from 73% to 75% were unsuccessful.   
It is possible that different combination of dropped columns, hidden layers and number of notes, as well binning of some features would be useful. An alternative model to try would be the random forest classifer.