Neural Network Model Optimization

Alphabet Soup Charity

Purpose

Alphabet Soup Charity(ABC) hired me to create a tool to help it select applications for funding with the best chance of succeeding in their ventures (opening round funding).

Data

ABC provided a csv with more than 34,000 organizations that they have funded in the past. In the csv file, there is metadata for each organization.

Development

In order to use the information supplied by ABC, first we had to pre-process the data

- The target variable is the **binary outcome** (IS_SUCCESSFUL) of success (1 - yes, 2 - no)
- The feature variables are all indexes listed in the data sheet that are not IS_SUCCESSFUL
    - These include fields such as Name, Application Type, Affiliation, Classification, Use Case, Organization, Status, Income Amount, Special considerations and amount requested
- Prepare Data
    - Dropped the EIN and NAME columns as they are not relevant to creating our model
    - In an effort to get an idea of the volume of data, discern unique values and their counts, I used the “nunique” function.
        - Using this information a cutoff of 500 was established
        - All non-conforming values were binned to a group called “others”
    - Using pd.dummies, non-categorical data was encoded.
    - Split the data into a train and test group to prepare for neural network creation and training.
    - Scale the training and testing data by sing a StandardScaler instance
- Compile, Train and Evaluate the Model
    - Using TensorFlow and Keras, assign a number of input features and nodes
    - Create layers for the model
    - Create an output layer with an appropriate activation function
    - Check the structure of the model
    - Train and then Compile the model tracking the loss and accuracy.
        - adam was used as the optimizer.

    - Evaluate the model using the test data.
        - Model was found to be 78.46% accurate
- Check against alternate model for compared performan
    - Random Forest
        - Accuracy was 77.6%

Further Analysis

Data Preprocessing

- What variable(s) are the target(s) for your model?
    - The Target was the binary outcome of successful (1) or not successful (0)
- What variable(s) are the features for your model?
    - These include fields such as Name, Application Type, Affiliation, Classification, Use Case, Organization, Status, Income Amount, Special considerations and amount requested
- What variable(s) should be removed from the input data because they are neither targets nor features?
    - EIN and NAME were removed in the initial model creation.
    - In optimizing, EIN, STATUS, and SPECIAL_CONSIDERATIONS were found not to be relevant to predictions and removed

Compiling, Training, and Evaluating the Model

- How many neurons, layers, and activation functions did you select for your neural network model, and why?
    - Both models were run with 500 epochs
    - For the initial model, I selected an input layer, two hidden layers and an output layer. This yielded ~71% accuracy
        - Nodes were chosen to be 80 & 30 in the hidden layers
        - Input features were the length of the X_train data.
        - Activation ‘relu’ was used in the hidden layers
            - If the input x is positive, relu produces 1
            - If the input is zero or negative, relu produces 0

        - Sigmoid Activation was used in the output layer
            - Takes any real value as an input and outputs 1 or 0
    - For the optimization, I added an additional hidden layer
        - Nodes were changed to 100 in the first layer, 50 in the second and 25 in the third
        - First hidden layer used ‘relu’
        - Second hidden layer through the Output Layer
            - Sigmoid Activation used (Experimentally)
    - Model Evaluated had 78.46% accuracy and 47.7% loss
- Were you able to achieve the target model performance?
    - Yes, I was able to achieve and exceed the target of 75% accuracy
- What steps did you take in your attempts to increase model performance?
    - I added an additional layer, increased the number of nodes in each layer and used ‘sigmoid’ activations.
    - Also, a different and more impactful feature set was used.
        - Features removed were not relevant to the binary outcome

Conclusion

In closing, the results of the deep learning model were impactful for the task given. Achieving an accuracy of 78.5% is pretty good given the crude nature of the model. With more refinement, perhaps a correlation analysis and an ensemble model, accuracy could be improved. Additionally, adding hyperparameter tuning or PCA analysis prior to model training could be useful.

Other recommendations would be to try to utilize some sort of unsupervised learning module with this data to see if it would be more accurate. Modules like K-Means or some other clustering model may help improve accuracy. Given the binary outcome, clustering could provide a better picture of what set of features correlate to each outcome.