

Overview:

The objective of the analysis is to develop a binary classification model using machine learning and neural networks that can predict the success of applicants for funding by Alphabet Soup. This tool aims to help Alphabet Soup select the applicants with the highest potential for successful ventures based on historical data of over 34,000 organizations that have previously received funding.

Results: Using bulleted lists and images to support your answers, address the following questions:

- Data Preprocessing
 - What variable(s) are the target(s) for your model?
 - The target IS_SUCCESSFUL is the variable indicating whether the funding received by an organization was used effectively. It is the target variable for the model, as the objective is to predict the success of applicants based on their characteristics.
 - What variable(s) are the features for your model?
 - The remaining columns in the dataset serve as features that the model will use to make predictions. These are:
 - APPLICATION_TYPE: Type of application submitted to Alphabet Soup
 - AFFILIATION: Sector of industry the organization is affiliated with
 - CLASSIFICATION: Government classification of the organization
 - USE_CASE: Purpose for which the funding is requested
 - ORGANIZATION: Type of organization
 - STATUS: Whether the organization is active or not
 - INCOME_AMT: Classification of the organization's income

- SPECIAL_CONSIDERATIONS: Special considerations associated with the application
 - ASK_AMT: Amount of funding requested
- What variable(s) should be removed from the input data because they are neither targets nor features?
 - The following were removed from the input data:
 - EIN: Identification column (generally not used as a feature for prediction but for reference)
 - NAME: Identification column (generally not used as a feature for prediction but for reference)
- Compiling, Training, and Evaluating the Model
 - How many neurons, layers, and activation functions did you select for your neural network model, and why?
 - First Hidden Layer: 80/100 neurons
 - Second Hidden Layer: 30/100 neurons
 - Output Layer: 1 neuron
 - Hidden Layers: ReLU (Rectified Linear Unit)
 - Output Layer: ReLU (Rectified Linear Unit) / sigmoid
 - Were you able to achieve the target model performance?
 - No. Even after optimizations I was unable to reach 75% accuracy.
 - What steps did you take in your attempts to increase model performance?
 - Dropped 2 more columns: "Status" and "Special Considerations"
 - Changed the neurons from 80/30 to 100
 - Changed layer from sigmoid to tanh in one iteration
 - Changed all layers to relu in another

Summary: Summarize the overall results of the deep learning model. Include a recommendation for how a different model could solve this classification problem, and then explain your recommendation.

Results of optimization: 268/268 - 1s - loss: 0.6696 - accuracy: 0.7272 - 953ms/epoch - 4ms/step; Loss: 0.6695956587791443, Accuracy: 0.7272303104400635

Perhaps further hyperparameter tuning such as increasing Epochs and Batch Sizes, changing activation functions, and/or increasing the number of layers/ neurons could improve accuracy.