**Neural Network Model**

Overview:

This analysis aims to preprocess a dataset of charity applications and develop a deep neural network model to predict their success. By refining the preprocessing steps and optimizing the neural network architecture, we strive to achieve a model accuracy of 75% or higher in forecasting the success of charity applications.

Results:

Model Performance:

* Accuracy Achieved: 72.84%
* Target Accuracy: 75% or above
* Optimization Techniques Used: Batch Normalization, Dropout, and Hyperparameter Tuning.

A screenshot of a computer code

Description automatically generated

Model Architecture and Performance:

* Input Layer: 512 neurons, ReLU activation
* Hidden Layers: 256, 128, 64, and 32 neurons with ReLU activation
* Output Layer: 1 neuron, Sigmoid activation

Data Processing:

The column ‘IS\_Successful’ is the target variable for the model. This indicates whether a charity application was successful or not. The features for the model all other variables in the dataset that excludes target and any non-beneficial identifiers such as ‘Application\_Type’, ‘Affiliation’, ‘Classification’, ‘Use\_Case’, ‘Organization’, ‘Income\_Amt’, ‘Special\_Considerations’, and ‘Ask\_Amt’. The variables removed from the model are’EIN’ and “NAME’ because they do not contribute to predicting the target variable. I removed ‘Status’ and ‘Ask\_AMT’ as well in the preprocessing steps to improve model performance. After preprocessing, the features are transformed into numeric form using one-hot encoding, and rare occurrences in certain categorical variables are binned into an "Other" category to reduce the number of unique values. This helps in simplifying the model and potentially improving its performance.

Compiling, Training, and Evaluating the Model:

There is a total of six layers I’ve used to define the model. The input layer has a total of 512 neurons that helps capture a wider range of features from the input data. First through fourth hidden layer neurons respectively is 256, 128, 64, and 32 with activation function: ReLU. These hidden layer neurons reduce the dimensionality, maintaining the ability to learn complex features. The output layer has 1 neuron with activation function sigmoid. The sigmoid function outputs a probability between 0 and 1.

A screenshot of a computer

Description automatically generated

Steps I took to increase the model performance were data processing, model architecture adjustments, and hyperparameter tuning. For data processing, I removed non-beneficial columns: (‘EIN’, “Name’, ‘Status’, and ‘Ask\_AMT’). I binned rare occurrences in categorical variables into an “Other” category to reduce the number of unique values. Likewise, scaled numerical features using ‘StandardScaler’ to normalize the data. For model architecture adjustments, I increased the number of neurons in the initial layers to capture more complex patterns. I used Batch Normalization after each layer to stabilize and speed up the training process. Also, I applied Dropout to prevent overfitting. For hyperparameter tuning, I adjusted the learning rate between 0.001 and 0.005 to find the optimal rate for convergence. Moreover, increased the number of epochs to allow the model more time to learn from the data.

Despite my efforts, the target accuracy of 75% may still require further optimization. The process of model improvement involves continuous experimentation and validation to achieve the desired performance.

A screenshot of a computer code

Description automatically generatedA close-up of a computer code

Description automatically generated



Summary:

The deep learning model to predict the success of charity applications reached an accuracy of about 72-73%, which is just below the 75% goal. Even after trying batch normalization, dropout, and fine-tuning, the performance wasn't quite there. To get better results, using methods like Random Forest or Gradient Boosting could help. These techniques can manage complex data, reduce overfitting, and show which features are most important. With proper data preparation and tuning, these models might hit the accuracy target. Switching to these ensemble methods could make the predictions more accurate and reliable.