**Neural Network Model Report for Alphabet Soup Funding Success Prediction**

**Overview of the Analysis**

The purpose of this analysis was to create a binary classification model for Alphabet Soup, a nonprofit foundation that provides funding to organizations. The foundation wanted a tool to help predict which funding applicants would be successful if funded, allowing them to make more effective decisions about where to allocate their resources. Using a dataset containing information on over 34,000 previously funded organizations, I developed and optimized a deep learning neural network model to identify patterns that indicate successful funding outcomes.

The primary goal was to build a model that could achieve at least 75% prediction accuracy, ensuring that Alphabet Soup can reliably identify promising funding candidates and maximize the impact of their financial support. This required careful preprocessing of the data, thoughtful design of the neural network architecture, and iterative optimization to improve performance.

**Results**

**Data Preprocessing**

* **Target Variable(s)**

The target for the model is the IS\_SUCCESSFUL column, which indicates whether the money was used effectively by the funded organization (1 = successful, 0 = not successful). This binary outcome is what we aim to predict with our neural network.

* **Feature Variables**

The features used for the model include:

* + NAME - Organization name (after binning into categories)
  + APPLICATION\_TYPE - Alphabet Soup application type
  + AFFILIATION - Affiliated sector of industry
  + CLASSIFICATION - Government organization classification
  + USE\_CASE - Use case for funding
  + ORGANIZATION - Organization type
  + STATUS - Active status
  + INCOME\_AMT - Income classification
  + SPECIAL\_CONSIDERATIONS - Special considerations for application
  + ASK\_AMT - Funding amount requested

These features provide the model with information about the organization's identity, type, purpose, and financial aspects that might influence funding success.

* **Variables Removed from Input Data**
  + EIN (Employer Identification Number) was removed because it's simply an identification number with no predictive value. Including it could confuse the model into finding patterns where none exist.

In the original model, NAME was also removed, but the optimized model retained this column after an appropriate binning, which significantly improved model performance.

**Compiling, Training, and Evaluating the Model**

* **Neural Network Architecture**

The optimized neural network model featured:

* + Three hidden layers with the following structure:
    - First hidden layer: 100 neurons with ReLU activation
    - Second hidden layer: 30 neurons with sigmoid activation
    - Third hidden layer: 10 neurons with sigmoid activation
  + Output layer: 1 neuron with sigmoid activation (appropriate for binary classification)
  + Total of 43,251 trainable parameters

This architecture was chosen because:

* + The increased depth (three layers instead of two) allows the model to learn more complex patterns in the data
  + The funnel structure (100→30→10→1) compresses information gradually, forcing the model to extract the most important features
  + Using ReLU activation in the first layer helps with efficient learning and mitigates vanishing gradients
  + Using sigmoid activations in later layers enhances binary classification performance
* **Target Model Performance**

Yes, the optimized model achieved an accuracy of 78.9% on the test data, exceeding the target performance of 75%. This represents a significant improvement over the original model's accuracy of 74.9%.

* **Steps to Increase Model Performance**

Several key strategies were implemented to enhance model performance:

* + **Feature Engineering with the NAME Column**
    - Instead of dropping the NAME column, it was retained and transformed
    - Organizations appearing 5 or fewer times were grouped into an "Other" category
    - This allowed the model to learn patterns based on organizations with established track records
  + **Enhanced Model Architecture**
    - Added a third hidden layer to increase model capacity
    - Increased the number of neurons in the first layer from 80 to 100
    - Used a combination of activation functions (ReLU for the first layer, sigmoid for subsequent layers)
  + **Refined Categorical Variable Handling**
    - For APPLICATION\_TYPE, values with fewer than 500 occurrences were grouped as "Other"
    - For CLASSIFICATION, values with fewer than 1000 occurrences were grouped as "Other"
    - This prevented overfitting to rare categories and reduced the dimensionality of the feature space

**Summary**

The deep learning neural network model successfully achieves the target performance, with an accuracy of 78.9% on the test data. This means that Alphabet Soup can use this model to identify applicants with a high probability of funding success nearly 80% of the time. The most significant finding was that the organization's name (identity) contained valuable predictive information, highlighting that past performance is often indicative of future success.

**Recommendation for an Alternative Model**

I recommend that Alphabet Soup also consider implementing a Random Forest classifier for this problem. During my optimization process, I tested a Random Forest model which achieved 77.6% accuracy, comparable to the neural network's performance.

Random Forests would be particularly well-suited for this application for several reasons:

1. **Interpretability**: Unlike neural networks, which function as "black boxes," Random Forests provide feature importance metrics that can help Alphabet Soup understand exactly which factors most strongly influence funding success.
2. **Robustness**: Random Forests handle non-linear relationships and outliers well without requiring extensive data preprocessing, making them more resilient to variations in the input data.
3. **Less Risk of Overfitting**: The ensemble nature of Random Forests (combining multiple decision trees) makes them less prone to overfitting than single deep learning models, particularly with limited data.
4. **Efficiency**: Random Forests typically require less computational resources and training time than deep neural networks while maintaining comparable performance.
5. **No Hyperparameter Sensitivity**: Random Forests are less sensitive to hyperparameter choices than neural networks, making them easier to implement and maintain over time.

Given that both models achieved similar performance, Alphabet Soup could use the neural network model as the primary prediction tool while employing the Random Forest model to gain insights into which features are most predictive of funding success. This combined approach would provide both high accuracy predictions and actionable insights to further refine their funding decision process.