**Neural Network Model Report**

**for Alphabet Soup**

**Shannon Hoffman**

**Overview**

The purpose of this analysis was to create and evaluate deep learning models aimed at predicting the success of applicants for funding, using data provided by Alphabet Soup. The goal was to develop a neural network model capable of accurately classifying applicants into successful and unsuccessful categories, ultimately assisting in the selection process for funding allocation.

**Results**

**Data Preprocessing**

* **Target Variable(s)**:
  + The target variable: ‘IS\_SUCCESSFUL’ indicating whether an applicant was successful in their venture.
* **Feature Variable(s)**:
  + The feature variables include all other columns in the dataset after dropping the non-beneficial ID columns (EIN and NAME). These features were one-hot encoded to convert categorical variables into a format suitable for the neural network.
* **Removed Variable(s)**:
  + The columns EIN and NAME were removed from the input data as they do not contribute to the model’s predictive power.

**Compiling, Training, and Evaluating the Models**

* **Model A**:
  + **Architecture**:
    - First Hidden Layer: 80 neurons, ReLU activation
    - Second Hidden Layer: 30 neurons, ReLU activation
    - Output Layer: 1 neuron, Sigmoid activation
    - Total Parameters: 11,821
  + **Performance**:
    - Accuracy: 73.13%
    - Loss: 0.5634
  + **Rationale**:
    - Model A was designed as a relatively simple neural network with two hidden layers, aiming to balance complexity with the ability to learn from the data.
* **Model B**:
  + **Architecture**:
    - First Hidden Layer: 100 neurons, ReLU activation
    - Second Hidden Layer: 50 neurons, ReLU activation
    - Third Hidden Layer: 25 neurons, ReLU activation
    - Output Layer: 1 neuron, Sigmoid activation
    - Total Parameters: Higher complexity due to additional neurons and layers
  + **Performance**:
    - Accuracy: 72.71%
    - Loss: 0.5863
  + **Rationale**:
    - Model B increased the complexity by adding more layers and neurons, aiming to capture more complex patterns in the data. However, this increase in complexity did not lead to better performance compared to Model A.
* **Model C**:
  + **Architecture**:
    - First Hidden Layer: 64 neurons, ReLU activation, L2 regularization
    - Dropout: 20%
    - Second Hidden Layer: 32 neurons, ReLU activation, L2 regularization
    - Dropout: 20%
    - Output Layer: 1 neuron, Sigmoid activation
    - Total Parameters: Moderate complexity with regularization techniques
  + **Performance**:
    - Accuracy: 72.54%
    - Loss: 0.5754
    - (Only ran for 24 epochs due to early stopping)
  + **Rationale**:
    - Model C introduced regularization techniques (L2 and dropout) to combat overfitting, which is common in deep learning models. However, the regularization slightly reduced accuracy, suggesting that further tuning is needed.
* **Target Model Performance**:
  + The target performance was not explicitly defined, but among the models tested, Model A provided the highest accuracy at 73.13%.
* **Steps to Increase Performance**:
  + **Model Complexity**: Increased the number of layers and neurons in Model B, but it led to a slight decrease in accuracy.
  + **Regularization**: Applied L2 regularization and dropout in Model C to prevent overfitting, which balanced the model's performance but did not achieve higher accuracy than Model A.
  + **Early Stopping**: Used early stopping in Model C to prevent overfitting and reduce training time.

**Summary**

The deep learning models developed in this analysis provided modest accuracy in predicting the success of applicants. Model A, with a simple two-layer architecture, achieved the highest accuracy of 73.13%. Increasing the model complexity in Model B did not yield better results, and applying regularization techniques in Model C provided a balance between model complexity and overfitting but did not surpass Model A in accuracy.

**Recommendation:**

* **Alternative Model**: A potential next step could be to explore ensemble methods, such as Random Forests, which are powerful in handling classification tasks with structured data. These models could be compared to the neural network approach to determine if they offer better performance for this specific classification problem.
* **Further Tuning**: If neural networks remain the preferred model, further hyperparameter tuning, more sophisticated regularization, and potentially more complex architectures could be explored to improve accuracy beyond what was achieved with Model A.