Using Deep Learning Model to Predict Charity Success

Overview of the Analysis:

The purpose of this analysis is to develop a deep learning model to predict the likelihood that charity organizations will be successful based on various features. This will help assist Alphabet Soup Charity in assessing how effective funding and resource allocation will be.

Results:

Data Preprocessing:

- **Target Variable(s)**:
    - IS_SUCCESSFUL is the target variable that indicates whether a charity organization was successful in achieving its goals.
- **Features for the Model**:
    - Classification and Application Type are considered features for the model.
- **Variables Removed**:
    - EIN and NAME columns were removed from the dataset because they are neither targets nor features.

Compiling, Training, and Evaluating the Model:

- **Neural Network, Layers, and Activation Functions**:
    - Input Layer: The input layer has the same number of neurons as the number of features in the dataset.
    - Hidden Layers: The model includes three hidden layers with varying numbers of neurons and ReLU activation functions. The numbers of neurons for each hidden layer are:
        - First hidden layer: 100 neurons
        - Second hidden layer: 80 neurons
        - Third hidden layer: 70 neurons

- Output Layer: The output layer has one neuron with a sigmoid activation function to predict the binary outcome.

```
# Define the model - deep neural net, i.e., the number of input features and hidden nodes for each layer.
nn = tf.keras.models.Sequential()
nn.add(tf.keras.layers.Dense(units=100, activation='relu', input_dim=X_train_scaled.shape[1]))

hidden_nodes_l1 =  80
hidden_nodes_l2 =  70
hidden_nodes_l3 =  10


# First hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_l1, activation="relu"))
nn.add(tf.keras.layers.Dropout(0.3))

# Second hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_l2, activation="relu"))

# Second hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_l3, activation="relu"))


# Output layer
nn.add(tf.keras.layers.Dense(units=1, activation="sigmoid"))

# Check the structure of the model
nn.summary()
```

- **Achievement of Target Model Performance**:
    - After training the model for 100 epochs with a batch size of 10,000, the achieved accuracy on the test data is approximately 73.35%. This was the best result out of three trained models.

```
# Evaluate the model using the test data
model_loss, model_accuracy = nn.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
268/268 - 1s - loss: 0.5521 - accuracy: 0.7313 - 614ms/epoch - 2ms/step
Loss: 0.5520956516265869, Accuracy: 0.7313119769096375
```

- **Steps to Increase Model Performance**:
    - Adjusted the number of neurons and layers in the model architecture.
    - Utilized dropout with a rate of 0.3 to mitigate overfitting.
    - Experimented with different learning rates to optimize model convergence.

Summary:

The deep learning model achieved an accuracy of approximately 73.35% on the test data, indicating its potential for predicting the success of charity organizations.

For improvement, using an alternative deep learning architecture. One that can capture complex patterns and relationships within dataset. Can also do more tuning to the parameter, and data augmentation techniques to enhance model performance.