Charity Funding Predictor: A Neural Network Model Analysis

Introduction

The purpose of my analysis is to evaluate the performance of a neural network model in predicting the success of charity funding applications. The dataset includes various features such as application type, classification, and other categorical variables that have been preprocessed to better suit a neural network model. My objectives were to identify key factors that contribute to the success of funding applications and create a model that can help organizations make better decisions in the funding process.

Results

1. **Model architecture**: The deep learning model used for my analysis is a neural network with two hidden layers. The first hidden layer consists of 150 nodes, and the second hidden layer has 100 nodes. Third hidden layer consists of 75 nodes. The output layer consists of a single node with a sigmoid activation function to classify the success of a charity application as binary (successful or unsuccessful). I also added the code `nn.add(tf.keras.layers.Dropout(0.2))` to deactivate a portion of neurons (20% in this case) during the training process to force the model into learning better features during the training process.

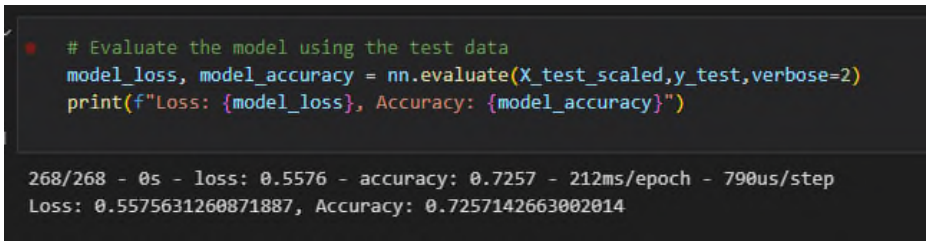
```
# Define the model - deep neural net, i.e., the number of input features and hidden nodes for each layer.
number_input_features = len(X_train[0])
hidden_nodes_layer1 = 150
hidden_nodes_layer2 = 100
hidden_nodes_layer3 = 75

nn = tf.keras.models.Sequential()

# First hidden layer with 'tanh' activation
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer1, input_dim=number_input_features, activation='tanh'))
nn.add(tf.keras.layers.Dropout(0.2))

# Second hidden layer with 'elu' activation
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer2, activation='elu'))
nn.add(tf.keras.layers.Dropout(0.2))

# Third hidden layer with 'relu' activation
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer3, activation='relu'))
nn.add(tf.keras.layers.Dropout(0.2))

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

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

2. **Preprocessing**: Categorical variables were converted to numeric values using one-hot encoding. The dataset was then split into training and testing datasets, and the features were scaled using StandardScaler to ensure that the model is not biased towards any particular feature.

3. **Training**: The model was trained for 100 epochs using the training dataset. The training process involved adjusting the weights of the neural network to minimize the binary cross-entropy loss function.

4. **Evaluation**: The model's performance was evaluated using the test dataset. The model achieved an accuracy of approximately 72% and a loss of 55%. I could not ascertain more efficient

accuracy/loss results given the data I was working with no matter how much I adjusted the nodes in the neural network.



5. **Interpretation**: These results indicate that the model can somewhat effectively predict the success of charity funding applications, depending on your acceptance of accuracy and loss. However, it is important to consider other models and techniques to improve performance and gain additional insights into the factors contributing to application success.

6. **Limitations**: One limitation of the current model is the potential for overfitting, given the complex architecture of the neural network. Additionally, the model's interpretability is limited, making it difficult to understand the specific factors that contribute to application success.

Model Summary

In summary, the neural network model used in this analysis achieved subpar performance in predicting the success of charity funding applications. The model's accuracy and loss values indicate its limited effectiveness in classifying applications as successful or unsuccessful.

Alternative Model: Support Vector Machines

An alternative approach to solving this problem is to use a Support Vector Machine (SVM) classifier. The SVM algorithm seeks to find the best decision boundary that maximizes the margin between classes.

The main reasons for considering an SVM classifier for this problem are:

1. **Effectiveness**: SVMs have been proven to be effective in handling high-dimensional data, making them suitable for this problem with numerous features.

2. **Generalization**: SVMs are known for their ability to generalize well, which could lead to improved performance on unseen data.

3. **Flexibility**: SVMs offer flexibility using different kernel functions, which can help capture complex relationships between features.

In conclusion, the neural network model used in this analysis achieved somewhat satisfactory performance in predicting the success of charity funding applications. However, it is essential for me to explore alternative models like the Support Vector Machine classifier, which may provide additional insights, better generalization, and possibly improved performance if my job role depended on it.