Module 19 -Deep Learning Challenge

Neural Network Report

Alphabet Soup Charity

Purpose

The nonprofit foundation Alphabet Soup has long been dedicated to empowering organizations that align with its mission by providing them with the necessary funding to achieve their goals. However, selecting the most promising applicants for funding is a complex task, given the vast number of organizations that seek support. To enhance the effectiveness of this process, Alphabet Soup seeks to implement a predictive tool that can identify the applicants with the highest likelihood of success in their ventures.

This analysis aims to leverage machine learning, specifically a neural network model, to develop a binary classifier that can predict the success of an applicant if funded by Alphabet Soup. Using a comprehensive dataset of over 34,000 previously funded organizations, which includes various features such as application type, affiliation, classification, use case, and funding amount requested, this model will enable Alphabet Soup to make data-driven decisions, ultimately improving the allocation of resources and maximizing the impact of its funding initiatives.

By accurately identifying successful applicants, Alphabet Soup can optimize its funding strategy, ensuring that the foundation's resources are utilized in the most effective and impactful manner.

Results

Data Preprocessing

- **What variable(s) are the target(s) for your model?**
    - **Target Variable:** The target variable identified is `IS_SUCCESSFUL`, which measures whether the funding provided was used effectively.
- **What variable(s) are the features for your model?**
    - Feature Variables: The features used to predict `IS_SUCCESSFUL` include:
        - 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
- **What variable(s) should be removed from the input data because they are neither targets nor features?**

- Removed Variables: EIN and NAME were excluded from the dataset as they are identifiers and do not contribute to the model's prediction capabilities.

Compiling, Training, and Evaluating the Model

How many neurons, layers, and activation functions did you select for your neural network model, and why?

Five different neural network models were tested, each with varying structures and optimizations:

1.	Description	Initial analysis after modifying Application and Classification features to collapse tail-end into “Other”
	Features	Columns= 'APPLICATION_TYPE', 'AFFILIATION', 'CLASSIFICATION', 'USE_CASE', 'ORGANIZATION', 'INCOME_AMT', 'SPECIAL_CONSIDERATIONS'
	Details	• # of Layers: Three • 1st layer: Activation = relu, 6 neurons • 2nd layer: Activation = sigmoid, 4 neurons • Output layer: Activation = sigmoid, 1 neuron • Optimizer: adam • Epoch: 100
	Parameters	Model: "sequential" Layer (type) | Output Shape | Param # dense (Dense) | (None, 6) | 264 dense_1 (Dense) | (None, 4) | 28 dense_2 (Dense) | (None, 1) | 5 Total params: 297 (1.16 KB) Trainable params: 297 (1.16 KB) Non-trainable params: 0 (0.00 B)
	Results:	Loss: 0.5520821213722229 Accuracy: 0.7332944869995117

2.	Description	**Optimization Attempt #1:** Same as above with

		hyperparameter tuning
	Features	Columns= 'APPLICATION_TYPE', 'AFFILIATION', 'CLASSIFICATION', 'ORGANIZATION', 'INCOME_AMT', 'SPECIAL_CONSIDERATIONS', 'USE_CASE'
	Details	• # of Layers: Six • 1st layer: Activation = relu, 100 neurons • 2nd layer: Activation = sigmoid, 75 neurons • 3rd layer: Activation = sigmoid, 50 neurons • 4th layer: Activation = sigmoid, 25 neurons • 5th layer: Activation = sigmoid, 10 neurons • Output layer: Activation = sigmoid, 1 neuron • Optimizer: adam • Epoch: 20
	Parameters	Model: "sequential" Layer (type) | Output Shape | Param # dense (Dense) | (None, 100) | 4,400 dense_1 (Dense) | (None, 75) | 7,575 dense_2 (Dense) | (None, 50) | 3,800 dense_3 (Dense) | (None, 25) | 1,275 dense_4 (Dense) | (None, 10) | 260 dense_5 (Dense) | (None, 1) | 11 Total params: 17,321 (67.66 KB) Trainable params: 17,321 (67.66 KB) Non-trainable params: 0 (0.00 B)
	Results:	Loss: 0.5507164597511292 Accuracy: 0.7330612540245056
3.	Description	**Optimization Attempt #2:** Searched for ASK_AMT outliers using a box plot, however no outlier was identified. Nevertheless, I identified a wide range of values. As a result, used logarithmic transformation to create bins for ASK_AMT

```
Log_ASK_AMT_Binned_Clean
$3K - $20K       26814
$20K - $90K       2711
$90K - $370K      2208
$370K - $1.4M     1310
$1.4M - $5.5M      737
$5.5M - $24M       280
$24M - $90M        143
$90M - $490M        56
$490M - $1.5B       33
$1.5B+               7
Name: count, dtype: int64
```

	Features	Columns= 'APPLICATION_TYPE', 'AFFILIATION', 'CLASSIFICATION', 'USE_CASE', 'ORGANIZATION', 'STATUS', 'INCOME_AMT', 'SPECIAL_CONSIDERATIONS', 'IS_SUCCESSFUL', 'ASK_AMT_BIN'
	Details	• # of Layers: Six • 1st layer: Activation = relu, 100 neurons • 2nd layer: Activation = sigmoid, 75 neurons • 3rd layer: Activation = sigmoid, 50 neurons • 4th layer: Activation = sigmoid, 25 neurons • 5th layer: Activation = sigmoid, 10 neurons • Output layer: Activation = sigmoid, 1 neuron • Optimizer: adam • Epoch: 20
	Parameters	Model: "sequential_1" Layer (type) | Output Shape | Param # dense_6 (Dense) | (None, 100) | 5,400 dense_7 (Dense) | (None, 75) | 7,575 dense_8 (Dense) | (None, 50) | 3,800 dense_9 (Dense) | (None, 25) | 1,275 dense_10 (Dense) | (None, 10) | 260 dense_11 (Dense) | (None, 1) | 11 Total params: 18,321 (71.57 KB) Trainable params: 18,321 (71.57 KB) Non-trainable params: 0 (0.00 B)
	Results:	Loss: 0.5574935674667358 Accuracy: 0.7223323583602905

4.	Description	**Optimization Attempt #3:** Ran a KerasClassifier and identified the weakest and strongest features, and removed the weakest

		features (Use_Case, ASK_AMT_BIN) Weakest feature: INCOME_AMT_0 with importance -0.0006647230320700026 Weakest feature: INCOME_AMT_25000-99999 with importance -0.0006064139941691193 Weakest feature: ASK_AMT_BIN_$3K - $20K with importance -0.00045481049562685617 Weakest feature: ORGANIZATION_Co-operative with importance -0.00025655976676385305 Weakest feature: INCOME_AMT_5M-10M with importance -0.0001516034985422965 Weakest feature: ASK_AMT_BIN_$1.5B+ with importance -0.0001282798833819432 Weakest feature: USE_CASE_Heathcare with importance -3.3306690738754695e-17 Weakest feature: USE_CASE_Other with importance 0.0
	Features	Columns= 'APPLICATION_TYPE', 'AFFILIATION', 'CLASSIFICATION', 'ORGANIZATION', 'STATUS','SPECIAL_CONSIDERATIONS'
	Details	• # of Layers: Six • 1st layer: Activation = relu, 100 neurons • 2nd layer: Activation = sigmoid, 75 neurons • 3rd layer: Activation = sigmoid, 50 neurons • 4th layer: Activation = sigmoid, 25 neurons • 5th layer: Activation = sigmoid, 10 neurons • Output layer: Activation = sigmoid, 1 neuron • Optimizer: adam • Epoch: 20
	Parameters	Model: "sequential_2" Layer (type) | Output Shape | Param # dense_12 (Dense) | (None, 100) | 3,000 dense_13 (Dense) | (None, 75) | 7,575 dense_14 (Dense) | (None, 50) | 3,800 dense_15 (Dense) | (None, 25) | 1,275 dense_16 (Dense) | (None, 10) | 260 dense_17 (Dense) | (None, 1) | 11 Total params: 15,921 (62.19 KB) Trainable params: 15,921 (62.19 KB) Non-trainable params: 0 (0.00 B)
	Results:	Loss: 0.5609097480773926 Accuracy: 0.7262973785400391
5.	Description	**Optimization Attempt #4:** Ran neural network auto-optimizer which tries different combinations of Activation ('relu', 'tanh', 'sigmoid'), Layers, Neurons, and Optimizers ('adam', 'rmsprop', 'sgd'). Moved forward with the best of 60 tests.

	Features	Columns= 'APPLICATION_TYPE', 'AFFILIATION', 'CLASSIFICATION', 'ORGANIZATION', 'STATUS','SPECIAL_CONSIDERATIONS'
	Details	• # of Layers: Six • 1st layer: Activation = sigmoid, 7 neurons • 2nd layer: Activation = sigmoid, 3 neurons • 3rd layer: Activation = sigmoid, 9 neurons • 4th layer: Activation = sigmoid, 9 neurons • 5th layer: Activation = sigmoid, 3 neurons • 6th layer: Activation = sigmoid, 7 neurons • 7th layer: Activation = sigmoid, 7 neurons • Output layer: Activation = sigmoid, 1 neuron • Optimizer: adam • Epoch: 20
	Parameters	Model: "sequential_2" Layer (type) | Output Shape | Param # dense_12 (Dense) | (None, 100) | 3,000 dense_13 (Dense) | (None, 75) | 7,575 dense_14 (Dense) | (None, 50) | 3,800 dense_15 (Dense) | (None, 25) | 1,275 dense_16 (Dense) | (None, 10) | 260 dense_17 (Dense) | (None, 1) | 11 Total params: 15,921 (62.19 KB) Trainable params: 15,921 (62.19 KB) Non-trainable params: 0 (0.00 B)
	Results:	Loss: 0.5586107969284058 Accuracy: 0.7220991253852844

Were you able to achieve the target model performance?

The objective was to achieve a predictive accuracy of 75% or higher. While none of the models reached this threshold, the initial baseline model (Model 1) provided the best performance. This model's simpler structure and granular inputs may have contributed to its slightly superior results.

Model	Accuracy	Loss
#1 - Initial Baseline	0.73329	0.55208

#2 - Hyperparameter Tuning	0.73306	0.55071
#3 - Logarithmic Transformation Binning (ASK_AMT)	0.72233	0.55749
#4 - KerasClassifier (Weakness & Strength)	0.72629	0.56090
#5 - Auto Optimizer	0.72209	0.55861

What steps did you take in your attempts to increase model performance?

Several strategies were employed to enhance model performance:

- **Hyperparameter Tuning:** Adjusting the number of layers and neurons (Exhibit 1).
- **Logarithmic Transformation:** Binning `ASK_AMT` to reduce the complexity of unique values (Exhibit 2).
- **Feature Reduction:** Removing weak features identified through a KerasClassifier analysis (Exhibit 3).
- **Auto-Optimization:** Implementing an automated search for the best combination of activation functions, layers, neurons, and optimizers (Exhibit 4).

Despite these efforts, the initial baseline model remained the most effective, suggesting that further refinement is needed. One potential avenue for improvement could be the strategic removal of low-performing features after the one-hot encoding (dummies) stage.

Exhibit 1

Model 1

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 6)	264
dense_1 (Dense)	(None, 4)	28
dense_2 (Dense)	(None, 1)	5

Total params: 297 (1.16 KB)
Trainable params: 297 (1.16 KB)
Non-trainable params: 0 (0.00 B)

Model 2

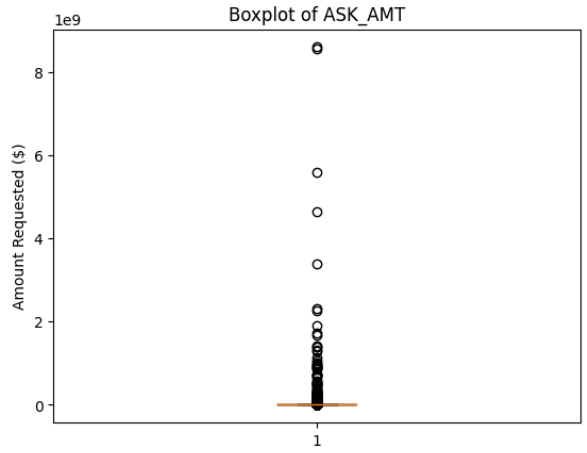
Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 100)	4,400
dense_1 (Dense)	(None, 75)	7,575
dense_2 (Dense)	(None, 50)	3,800
dense_3 (Dense)	(None, 25)	1,275
dense_4 (Dense)	(None, 10)	260
dense_5 (Dense)	(None, 1)	11

Total params: 17,321 (67.66 KB)
Trainable params: 17,321 (67.66 KB)
Non-trainable params: 0 (0.00 B)

Exhibit 2

ASK_AMT Box Plot



Logarithmic Transformation of ASK_AMT

```
Log_ASK_AMT_Binned_Clean
$3K - $20K        26814
$20K - $90K        2711
$90K - $370K       2208
$370K - $1.4M      1310
$1.4M - $5.5M       737
$5.5M - $24M        280
$24M - $90M         143
$90M - $490M         56
$490M - $1.5B        33
$1.5B+                7
Name: count, dtype: int64
```

Exhibit 3

Weakest Features (20)

```
Weakest feature: INCOME_AMT_0 with importance -0.0006647230320700026
Weakest feature: INCOME_AMT_25000-99999 with importance -0.0006064139941691193
Weakest feature: ASK_AMT_BIN_$3K - $20K with importance -0.0004548104956268561 7
Weakest feature: ORGANIZATION_Co-operative with importance -0.0002565597667638530
Weakest feature: INCOME_AMT_5M-10M with importance -0.0001516034985422965
Weakest feature: ASK_AMT_BIN_$1.5B+ with importance -0.0001282798833819432
Weakest feature: USE_CASE_Heathcare with importance -3.3306690738754695e-17
Weakest feature: USE_CASE_Other with importance 0.0
Weakest feature: STATUS_0 with importance 1.1661807580176654e-05
Weakest feature: STATUS_1 with importance 1.1661807580176654e-05
Weakest feature: AFFILIATION_Other with importance 2.3323615160353307e-05
Weakest feature: INCOME_AMT_100000-499999 with importance 5.830903790084996e-05
Weakest feature: SPECIAL_CONSIDERATIONS_Y with importance 9.329446064139102e-05
Weakest feature: APPLICATION_TYPE_T7 with importance 0.0001282798833818768 8
Weakest feature: SPECIAL_CONSIDERATIONS_N with importance 0.0001282798833819209 7
Weakest feature: AFFILIATION_Regional with importance 0.0001632653061224287 3
Weakest feature: AFFILIATION_National with importance 0.000268221574343985 3
Weakest feature: ASK_AMT_BIN_$24M - $90M with importance 0.0002682215743439853
Weakest feature: ORGANIZATION_Corporation with importance 0.0003032069970845152 5
Weakest feature: ASK_AMT_BIN_$1.4M - $5.5M with importance 0.0003848396501457518 4
```

Strongest Features (20)

```
Strongest feature: AFFILIATION_CompanySponsored with importance 0.05385422740524776
Strongest feature: AFFILIATION_Independent with importance 0.051720116618075754
Strongest feature: APPLICATION_TYPE_T4 with importance 0.021807580174927076
Strongest feature: APPLICATION_TYPE_T10 with importance 0.010775510204081563
Strongest feature: CLASSIFICATION_Other with importance 0.008979591836734625
Strongest feature: APPLICATION_TYPE_T5 with importance 0.008804664723032008
Strongest feature: APPLICATION_TYPE_T6 with importance 0.007591836734693824
Strongest feature: ORGANIZATION_Association with importance 0.004664723032069928
Strongest feature: ORGANIZATION_Trust with importance 0.004641399416909564
Strongest feature: CLASSIFICATION_C3000 with importance 0.0039883381924197825
Strongest feature: CLASSIFICATION_C2100 with importance 0.0036034985422739862
Strongest feature: CLASSIFICATION_C2000 with importance 0.003486880466472231
Strongest feature: INCOME_AMT_1-9999 with importance 0.003008746355685088
Strongest feature: APPLICATION_TYPE_T19 with importance 0.002985422740524723 7
Strongest feature: INCOME_AMT_10M-50M with importance 0.0029154518950436636
Strongest feature: CLASSIFICATION_C1200 with importance 0.0020058309037900403
Strongest feature: ASK_AMT_BIN_$20K - $90K with importance 0.0018075801749270592
Strongest feature: APPLICATION_TYPE_T7 with importance 0.001236151603498481
Strongest feature: USE_CASE_CommunityServ with importance 0.0011661807580174322
Strongest feature: ASK_AMT_BIN_$5.5M - $24M with importance 0.0010379008746355112
```

Exhibit 4

Best hyperparameters based on Auto-Optimizer

```
{'activation': 'sigmoid',
 'first_units': 7,
 'num_layers': 4,
 'units_0': 3,
 'units_1': 9,
 'units_2': 3,
 'units_3': 7,
 'units_4': 5,
 'units_5': 1,
 'add_dropout_0': 0,
 'dropout_0': 0.30000000000000004,
 'optimizer': 'adam',
 'add_dropout_1': 1,
 'dropout_1': 0.1,
 'add_dropout_2': 0,
 'dropout_2': 0.0,
 'add_dropout_3': 1,
 'dropout_3': 0.0,
 'add_dropout_4': 0,
 'dropout_4': 0.30000000000000004,
 'add_dropout_5': 0,
 'dropout_5': 0.0,
 'tuner/epochs': 20,
 'tuner/initial_epoch': 7,
 'tuner/bracket': 1,
 'tuner/round': 1,
 'tuner/trial_id': '0015'}
```

Summary

In this project, five neural network models were constructed and evaluated. The initial baseline model emerged as the most effective predictor of success among the tested models. The results indicate that the model's simplicity and granular inputs played a key role in its performance. Future work could explore additional feature engineering techniques or alternative machine learning models to achieve the desired accuracy. Additionally, careful consideration should be given to the removal of low-impact features to further enhance model performance.