# 21-deep-learning-challenge

Alphabet Soup Foundation funding

**ABSTRACT:**

• Accuracy for the initial model was 72%

• Accuracy did not improve above 73% with the 3 trials of optimization

• Optimization included increasing the number of nodes, adding a third hidden layer, and using less bins with the application\_type

**OVERVIEW**

The nonprofit foundation Alphabet Soup wants a tool that can help it select the applicants for funding with the best chance of success in their ventures. With your knowledge of machine learning and neural networks, you’ll use the features in the provided dataset to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup.

From Alphabet Soup’s business team, you have received a CSV containing more than 34,000 organizations that have received funding from Alphabet Soup over the years. Within this dataset are a number of columns that capture metadata about each organization, such as:

•EIN and NAME—Identification columns

•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

•IS\_SUCCESSFUL—Was the money used effectively

**RESULTS**

Data Preprocessing:

•The target variables is the column 'IS\_SUCCESSFUL'. Was the applicant successful?

•The feature variables are APPLICATION\_TYPE, AFFILITATION, CLASSIFICATION, USE\_CASE, ORGANIZATION, STATUS, INCOME\_AMT, SPECIAL\_CONSIDERATIONS, ASK\_AMT

•The variables removed are EIN, and NAME

Compiling, Training, and Evaluating the Model:

• The initial neural network is comprised of:

1) input layer of 43 since there are 43 columns once the get.dummies to encode the categorical variables

2) first hidden layer with 6 nodes - based on information learned in class

3) second hidden layer with 3 nodes - based on information learned in class

4) activation function 'relu' in the two hidden layers

5) activation function 'sigmoid' in the output layer

• The first attempt at optimization has the following adjustment:

1) first hidden layer increased to 86 nodes - decided to double the nodes based on the input value of 43

2) second hidden layer increased to 43 - decided to halve the number nodes in first hidden layer

In our class activities, the input layer usually used a value of 2 features. In comparison, this homework challenge is using an input value of 43 features once the get.dummies is used. I essentially decided to use significantly more nodes as my first optimization attempt.

• The second attempt at optimization has the following adjustment:

1) third hidden layer added with 21 nodes - the rationale was to halve the number of nodes from the second hidden layer

All other variables remained the same and the only adjustment in the second attempt was the addition of the third hidden layer.

• The third attempt at optimization has the following adjustment:

1) decreased the number of bins for the APPLICATION\_TYPE and set it to <500. This

adjustment provided 8 bins instead of the 9 bins in the original network.

All other prior adjustments remained the same when I ran the code using the third optimization attempt.

**RESULTS**

The initial neural network demonstrated an accuracy of 72% which is a good starting point for Alphabet Soup Charity to identify which applicants will be successful in funding. The graph demonstrates the accuracy rate of the initial network. However, the model should be further evaluated to determine if a higher accuracy rate can be achieved.



Three attempts were made to optimize the model and improve its accuracy. None of the attempts achieved a higher accuracy rate than 73%. This graph demonstrates the accuracy rate of the first optimization attempt that increased the number of nodes in the two hidden layers.



Only 1% accuracy was achieved with the three optimization attempts. The question arises as to next steps Alphabet Soup Charity may want to consider:

1) Continue with additional optimization attempts which may include other hyperparameters, different algorithms, increasing data size if possible

2) Is overfitting an issue?

3) Resources - Alphabet Soup Charity may need to consider it resources and whether continuing or stopping is the best business tactic to take. Does the company want or have the time, computational power or data availability to make further optimization attempts business effective?

4) Business objectives - reconsider the business objectives. Perhaps 73% accuracy meets the objectives. Perhaps the business objective(s) need to be re-evaluated.