**Neural Network Model: Alphabet Soup Charity**

**Overview**

The purpose of this report is provided analysis on the results of the first model and subsequent optimization method outcomes. The analysis will aid the Alphabet Soup business team in predicting which applicants are most likely to be successful in their ventures. This will help the charity make best use of funding awards by increasing the probability of funding successful applicants.

**Results**

Outcomes of each model are follows:

* The first model resulted in a loss of .553 and an accuracy of .727 with 40 nodes in the first layer and 30 nodes in the second layer.

A screenshot of a computer code

Description automatically generated with low confidence

**Figure 1.** First model using two hidden node layers with 70 total nodes.

* The second model resulted in a loss of .553 and an accuracy of .723 with optimization method 1. Optimization method 1 used the first model with an additional hidden layer of 14 nodes.

A screenshot of a computer code

Description automatically generated with low confidence

**Figure 2.** Method 1: Add hidden layer optimization model.

* The third model resulted in a loss of .607 and an accuracy of .661 with optimization method 2. Optimization method 2 used the first model with two additional columns removed such as STATUS and AFFILIATION.

A screenshot of a computer code

Description automatically generated with low confidence

**Figure 3.** Method 2: Remove STATUS and AFFILIATION ID columns optimization model.

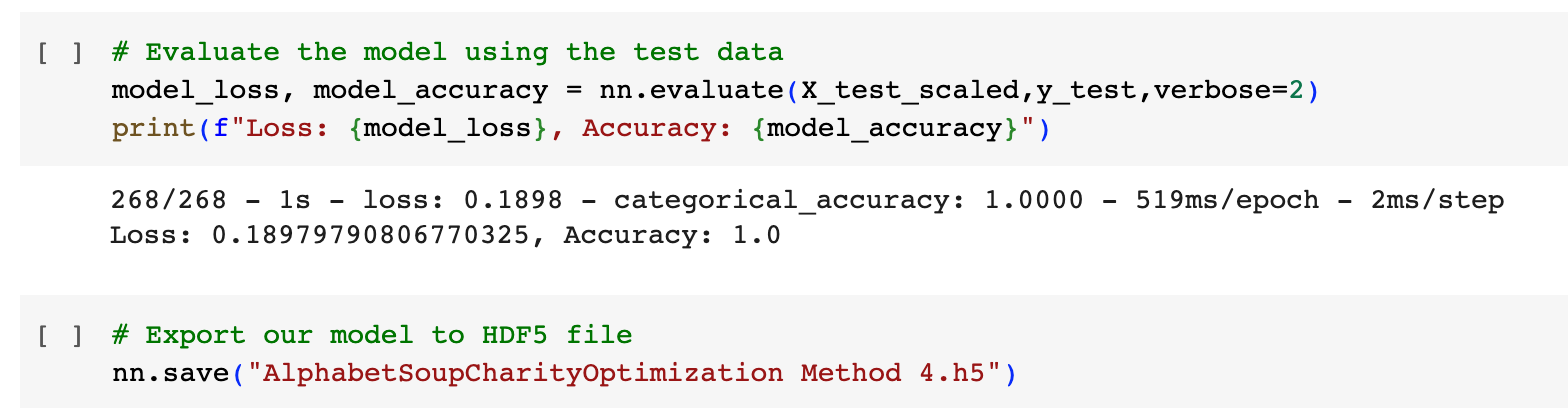
* The fourth model resulted in a loss of .559 and an accuracy of .725 with optimization method 3. Optimization method 3 used the first model with a change in activation layer from sigmoid to tanh.

A screenshot of a computer code

Description automatically generated with low confidence

**Figure 4.** Method 3: Change Activation layer from sigmoid to tanh.

* The fifth model resulted in a loss of .190 and an accuracy of 1.0 with optimization method 4. Optimization method 4 used the first model with a change in the compile method from the ‘adam’ optimizer to the ‘sgd’ optimizer.



**Figure 5.** Method 4: Change compile from ‘Adam’ optimizer to ‘SGD.’

**Results Questions**

* Data Preprocessing
  + What variable(s) are the target(s) for your model?

Variable that is the target for my model is the ‘IS\_SUCCESSFUL’ column.

* + What variable(s) are the features for your model?

Features for my model are all columns except for the ‘IS\_SUCCESSFUL’ column.

* + What variable(s) should be removed from the input data because they are neither targets nor features?

I removed the EIN and NAME because they had no effect on the outcome.

* Compiling, Training, and Evaluating the Model
  + How many neurons, layers, and activation functions did you select for your neural network model, and why?

I chose a total of 70 neurons spread out among two hidden layers with relu activations for my sigmoid output layer model because instructional material implied that most models only require one hidden layer, 11 epochs, and 42 neurons. I had 43 inputs and wanted to keep the neuron count under 86 neurons from information provided here:

<https://medium.com/geekculture/introduction-to-neural-network-2f8b8221fbd3>

* + Were you able to achieve the target model performance?

I was not able to achieve 75% target and maintained approximately 72% accuracy for all models except for optimization methods 2 and 4.

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

I removed feature columns, added a hidden layer, changed the output layer activation, and changed the compile method.

**Summary**

Overall results demonstrated that removing certain feature columns and invoking the sgd optimizer decreased and increased model performance respectively. Removing the STATUS and AFFILIATION columns decreased performance with an increase in loss and a decrease in accuracy because these features affected the outcome of the model. STATUS and AFFILIATION are features that matter to the outcome of the model and the model did not perform as well without them. Conversely, the sgd optimizer improved the performance of the model by decreasing losses to .19 and increasing accuracy to 1.0 Sgd is argued to do a better jog than adam as described here: <https://medium.com/geekculture/a-2021-guide-to-improving-cnns-optimizers-adam-vs-sgd-495848ac6008>.