Module 21 – Deep Learning

# Neural Network Model Report:

Overview

The goal of this challenge was to create a neural network model that could accurately predict whether or not an applicant would be successful if funded by the fictitious nonprofit entity, Alphabet Soup.

Process

I had a CSV file containing information on applicants that had been funded by Alphabet Soup, including information about the applicant organizations and whether or not they were successful.

The first step in this challenge was to preprocess the data, before moving on to building and training a neural network model. To preprocess the data I:

* read the CSV file into pandas,
* dropped columns that were not needed (e.g. EIN),
* found unique values and value counts in order to properly bin the data,
* used get dummies to convert categorical data to numerical data,
* split the data into training and testing data using train\_test\_split,
* scaled the data using StandardScaler.

The target of the model was whether or not an applicant was successful (Is Successful column). The features were based on the following columns:

* Application Type
* Affiliation
* Classification
* Use Case
* Status
* Income Amount
* Ask Amount

After preprocessing my data there were 44 features (this is before I tried any optimization) – because of the categorical data that was converted to numeric.

The goal of this challenge was to create a model that would predict with 75% accuracy. I created three separate optimization attempts and still only had about 73% accuracy. The models I created are described below:

Initial Attempt (AlphabetSoupCharity file):

Hyperparameters:

* Layers = 2
  + Layer 1 – 11 nodes and relu activation function
  + Layer 2 – 7 nodes and tanh activation function
* Epochs = 100
* Accuracy & model loss = 72% and .56 loss

Optimization Attempt #1:

Hyperparameters:

* Layers = 2
  + Layer 1 – 70 nodes and tanh activation function
  + Layer 2 – 30 nodes and tanh activation function
* Epochs = 300
* Accuracy & model loss = 73% and .58 loss

Optimization Attempt #2:

Hyperparameters:

* Layers = 4
  + Layer 1 – 15 nodes and relu activation function
  + Layer 2 – 11 nodes and relu activation function
  + Layer 1 – 7 nodes and sigmoid activation function
  + Layer 2 – 5 nodes and tanh activation function
* Epochs = 75
* Accuracy & model loss = 73% and .57 loss

Optimization Attempt #3:

Hyperparameters:

* Layers = 4
  + Layer 1 – 35 nodes and tanh activation function
  + Layer 2 – 21 nodes and tanh activation function
  + Layer 1 – 17 nodes and tanh activation function
  + Layer 2 – 11 nodes and tanh activation function
* Epochs = 75
* Accuracy & model loss = 73% and .57 loss

None of my attempts at optimization yielded higher accuracy than my first attempt. I tried different things – more nodes and more layers, more or fewer epochs, different activation functions – but these did not seem to affect the accuracy of the model.