TECNOLÓGICO DE MONTERREY

DATA ANALYTICS BOOTCAMP

MODULE 19 – NEURAL NETWORKS AND DEEP LEARNING MODELS

---

1. OVERVIEW OF THE ANALYSIS

Becks works with the non-profit Alphabet Soup as a data scientist and coder. They're a charitable organization devoted to assisting groups that work to conserve the environment, improve people's lives, and bring the globe together. In the last 20 years, Alphabet Soup has raised and contributed nearly $10 billion. This money has been utilized to fund life-saving technology and forestry projects all across the world.

Beck is in charge of the whole organization's data collecting and analysis. Her role entails evaluating the impact of each contribution as well as screening potential receivers. This helps to guarantee that the foundation's funds are spent wisely.

Regrettably, not every gift made by the firm has an impact. In certain circumstances, a company will accept the money and then vanish. As a result, Alphabet Soup's president, Andy Glad, has asked Beck to forecast which groups are worthy of donations and which are too risky. He wants her to come up with a mathematical data-driven method that can accurately achieve this.

Beck has determined that the statistical and machine learning approaches she has deployed are insufficient to solve this challenge. She will instead create and train a deep learning neural network. This model will assess all forms of incoming data and generate a clear decision-making outcome.

I'll help Beck learn about neural networks and how to create and train them with the Python TensorFlow framework in this module. To test and enhance Ir models, I'll depend on various previous skills with statistics and machine learning. I will be able to build a strong deep learning neural network capable of comprehending huge complicated datasets by the conclusion of this module.

PURPOSE: Help Becks and Alphabet Soup determine which organizations should receive donations.

---

2. RESULTS OF THE ANALYSIS

2.1 Data Preprocessing

What variable(s) are considered the target(s) for your model?: IS_SUCCESSFUL column (whether or not the money was used effectively), is the target variable.

Image 1. Target variable

IS_SUCCESSFUL
1
1
0
1
1

What variable(s) are considered to be the features for your model?: 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 consideration for application, ASK_AMT—Funding amount requested columns were used as feature variables.

Image 2. Feature variables

APPLICATION_TYPE	AFFILIATION	CLASSIFICATION	USE_CASE	ORGANIZATION	STATUS	INCOME_AMT	SPECIAL_CONSIDERATIONS	ASK_AMT
T10	Independent	C1000	ProductDev	Association	1	0	N	5000
T3	Independent	C2000	Preservation	Co-operative	1	1-9999	N	108590
T5	CompanySponsored	C3000	ProductDev	Association	1	0	N	5000
T3	CompanySponsored	C2000	Preservation	Trust	1	10000-24999	N	6692
T3	Independent	C1000	Heathcare	Trust	1	100000-499999	N	142590

What variable(s) are neither targets nor features, and should be removed from the input data?: EIN and NAME—Identification columns, and APPLICATION_TYPE—Alphabet Soup application type were dropped as they didn't provide any value to the model.

Image 3. Removed variables from original dataset

	EIN	NAME
0	10520599	BLUE KNIGHTS MOTORCYCLE CLUB
1	10531628	AMERICAN CHESAPEAKE CLUB CHARITABLE TR
2	10547893	ST CLOUD PROFESSIONAL FIREFIGHTERS
3	10553066	SOUTHSIDE ATHLETIC ASSOCIATION
4	10556103	GENETIC RESEARCH INSTITUTE OF THE DESERT

2.2 Compiling, Training, and Evaluating the Model

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

Image 4. Shape of the original model

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 80)                3520

 dense_1 (Dense)             (None, 30)                2430

 dense_2 (Dense)             (None, 1)                 31

=================================================================
Total params: 5,981
Trainable params: 5,981
Non-trainable params: 0
_________________________________________________________________
```

Were you able to achieve the target model performance?: No, the preprocessing and the shape of the neural network resulted in a accuracy of 0.5324

What steps did you take to try and increase model performance?: The accuracy of the original model with the preceding stages was 0.5324, but when applied to the test variables, it was reduced to 0.4625. After examining the features, I chose to remove the 'CLASSIFICATION' feature since it included too many unique values that may alter my model's weights. To minimize the number of categories, I altered the binning intervals for the 'APPLICATION TYPE' and 'AFFILIATION' variables. Increase the number of neurons in your brain. I also raised the number of neurons in the first hidden layer to 90 and 45 in the second hidden layer for the first try, while keeping the same activation functions. Then I created a third layer with 15 nodes and the same activation functions as the previous two layers. Finally, I went back to the two hidden layer model but altered the activation function of the first layer to the leaky-relu to test whether allowing for negative values may increase the accuracy. I improved my model's performance, but not enough to hit the 75% mark.

Image 5. Shape of the optimized model (3rd attempt)

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 90)                3960

 dense_1 (Dense)             (None, 45)                4095

 dense_2 (Dense)             (None, 1)                 46

=================================================================
Total params: 8,101
Trainable params: 8,101
Non-trainable params: 0
_________________________________________________________________
```

3. SUMMARY OF THE ANALYSIS

As can be seen from the optimization attempts, I was able to improve on the initial findings. Increasing the number of neurons improved accuracy and reduced loss marginally. Increasing the number of neurons reduced the loss and improved accuracy slightly, but not much. Finally, modifying the activation functions resulted in somewhat improved accuracy but a significant loss, suggesting that the model was overfitted. I conclude that Attempt 2 was the best one, with more neurons, two hidden layers, and the activation function relu' in both levels. Because the data contains numerous categorical variables, it would be smart to use a random forest model boosting or perhaps a mixture of logistic regression models as a proposal.

Table 1. Summary of the analysis

MODEL	HIDDEN LAYERS	NEURONS	ACTIVATION FUNCTION	LOSS	ACCURACY
Original	Two	80 and 40	relu, relu, sigmoid	81.11%	46.25%
Attempt 1	Two	90 and 45	relu, relu, sigmoid	82.63%	68.04%
Attempt 2	Three	90, 45 and 15	relu, relu, relu, sigmoid	67.24%	54.39%
Attempt 3	Two	90 and 45	relu, relu, sigmoid	264.34%	69.21%