Sacramento State University

Project 1 - Report

CSC 180 Intelligent Systems

Due: February 26, 2021 @ 11:00am

Logan Hollmer (**ID:** 301559973)
Quinn Roemer (**ID:** 301323594)

Problem Statement:

In project 1 we were tasked with creating a neural network that predicted the star rating of a business based on all the reviews for that business. In this project, we used a dataset provided by Yelp for academic purposes. The first step in this project is extracting the data from the downloaded JSON files into a Pandas data frame. After this, all the businesses must be grouped on the same row as the business ID. This is because the neural network created will ingest all the reviews for a business as input. After this, the reviews were vectorized using TF-IDF. After doing so, several models will be created with varying parameters such as optimizers, activation functions, and layer/neuron count. The best model created is then saved, graphed, and some example output is shown.

Methodology:

After understanding the problem statement the group discussed the best way to approach the problem. There was a unanimous agreement that the best approach would be setting up an automatic function for building, training, and testing models. This function would randomize the activation function, the number of layers/neurons, and the optimizer. Since both group members wanted to try doing the whole process, it was decided that each person would create a dataset and train their own models. In the end, the best model from each person would be compared, and the final model chosen between those two. When preprocessing the data the businesses with less than 20 reviews were removed. Logan chose to keep the total number of reviews as a feature which he linearized. He also decided to use a total of 1k features for the TF-IDF vectorizer thinking that 1k would be enough to capture any important words. On the other hand, Quinn decided to only use the reviews as his input and set the max features for TF-IDF to 2k. Quinn's function to build models was designed so the first hidden layer would have between 10 and 100 neurons and then up to 2 subsequent hidden layers with each one having between 10 and 100 neurons. Each model would be run with both the Adam and SGD optimizers. In Logan's, the first hidden layer had between 10 and 200 neurons and then would have up to 4 subsequent hidden layers each with statistically fewer neurons between 10 and 109. In this set up the optimizer was randomly chosen. Each setup was allowed to run overnight creating and testing between 100 and 200 different types of models. There was no particular pattern to the models that were created, the random function came up with models of several different types and configurations.

Experimental Results and Analysis:

A total of 300 models were created, trained, and tested. Logan found that his top model had an RMSE of 0.2515 with only one hidden layer with 51 neurons utilizing the relu activation function and the adam optimizer. Quinn's top model had an RMSE of 0.2499 and consisted of 3 hidden layers of 96, 13, and 15 neurons respectively. All the layers utilized the tanh activation function and the optimizer was adam.

Of interest, all the models trained by Logan that were close to his best model also had a single layer. This seems to suggest that there was not much interaction between the different features that were fed in and one layer seemed to be optimal. On the contrary, all the models trained by Quinn that were close to his best model used several hidden layers. This seems to suggest that there was interaction being captured between the different features. When comparing the two models it appears that capturing more words using TF-IDF provides better insight into the business's ratings than the number of reviews. It also suggests that there is some complex interplay between the words that are used.

When examining Quinn's top models (Table 1 on the next page) it appears that adam was the best optimizer. It appeared the most when looking at the top models and also was the optimizer for both Logan's and Quinn's top model. One area of interest is that for all the top models, the activation function is the same for every layer. Since the activation function for each layer is chosen at random, this seems to suggest that the best results are achieved when the activation function is consistent across the layers. Also, the tanh function appeared the most and seemed to be on average the best for this problem set. However, other activation functions were used as seen in Table 1. It also seems that having more than one hidden layer is optimal; having 2 to 3 hidden layers with a decreasing amount of neurons performed the best. However, model number 1 has very similar results with only one hidden layer so further research is required. Logan had a few models that performed well with only one hidden layer which may have been the result of having fewer features as inputs.

Table 1

Model ID	RMSE	Number of hidden Layers	Neurons per layer	Activation function	Optimizer
1	0.26034	1	15	relu	sgd
2	0.25064	3	38, 77, 16	tanh, tanh, tanh	adam
3	0.25053	3	95, 20, 70	tanh, tanh, tanh	adam
4	0.25031	2	66, 38	sigmoid, sigmoid	adam
5	0.24992	3	96, 13, 15	tanh, tanh, tanh	adam

Table 1 displays Quinn's top 5 models, including the RMSE value, number of hidden layers, neurons per layer, activation functions, and the optimizer.

Task division and project reflection:

As previously stated, both group members did the entire project. Each member processed, vectorized, and trained their own models. The writing of the report was split between each member with Logan writing the sections on *Methodology* and *Experimental Results and Analysis* and Quinn writing the section on the *Problem Statement* and the section on *Task Division and Project Reflection.* The presentation was scripted and recorded by Quinn.

Reflecting on the project, if we had more time we would have liked to have added the number of reviews to the 2k TF-IDF array to see how it would change the result. We also felt that more data from the reviews by increasing the max number of features would have been extremely useful. We believe that using around 5k features would have given much better results.

Also, we realized that we are making a very general model for reviewing many types of businesses, a more accurate model could have been achieved by focusing on one business category. However, even if a model could not be created for each business category at the very least including the category as a feature would have increased the accuracy.

Upon examining the model produced by Quinn, it seemed to capture that ratings should not go above 5, but it did not seem to capture that results should be in one-half increments. In regards to this, we believe that more data would have allowed for a more accurate result.

One of the biggest challenges we faced was memory size. We often had to save data to the disk and load it back in at a later stage to save space. Another challenge was having the time to train enough models. Due to the random selection of our model architecture, we had to build and train many different models to ensure we were getting an optimal solution. We ended up testing about 300 different models, however, we felt as if we could have run the program for longer to potentially find a more optimal model.