Kai Kit Lok (Brian) - 301108938
Kevin Estrada - 216179224
(Team Lead)Cody Wuco -301090621

CSC 180.02 - Inteligent System

Project1 : Yelp Business Rating Prediction using Tensorflow

Due Date: September 30, 2021

Problem Statement

In this project, we aim to predict a business's star rating based on all of the review text for that business using neural network implementations in TensorFlow. Consider this problem as a regression problem.

Methodology

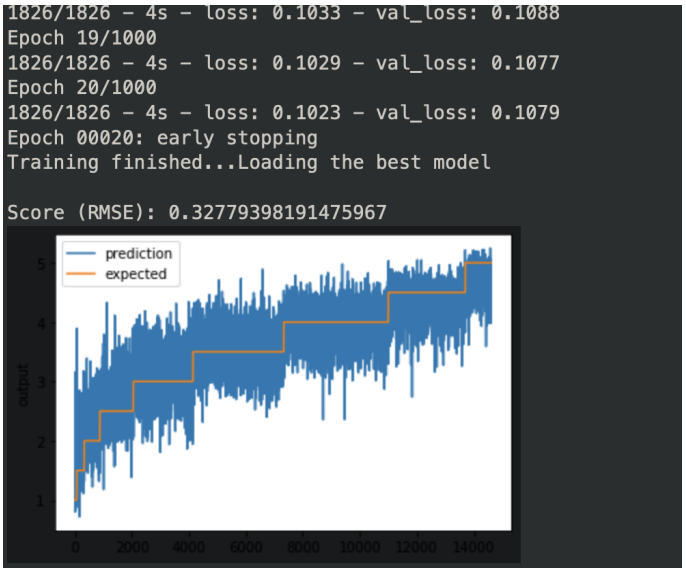
A dataframe was created from 2 JSON files which contained information about the business. Attributes like name, location, star ratings, reviews etc. The other JSON file was from the reviews that the users left on the business.

We then used the TF IDF method to extract extra information from the reviews file. We also created neural networks and early stopping for the training of the data

Tools: pandas, numpy, sci-learn, tensorflow, Jupyter notebook

Experimental Results and Analysis

	name	stars	categories	predict
4	Mr G's Pizza & Subs	4.0	Food, Pizza, Restaurants	3.758507
7	Pittock Mansion	4.5	Tours, Museums, Architectural Tours, Venues & ...	4.354261
10	Blake's On The Park	3.5	Nightlife, Bars, Gay Bars, Sports Bars	3.606768
13	ARGO	4.0	Food Delivery Services, Food, Restaurants, Med...	3.856143
16	Texas Roadhouse	3.5	Steakhouses, Restaurants, Salad, Barbeque, Ame...	3.875949



The prediction was fairly accurate, as you can see from the first picture it is not too far off from the actual rating. We also ran the RMSE score multiple times and that was the lowest we could get it.

Resources did affect the performance and we could have run it more times to get a smaller score. Making the prediction a lot more accurate.

Hyper Parameter tuning

Each run had a 5 model loop.

20 neurons in the first layer, 10 in the second layer

HH:MM:SS	Adam	SGD
Sigmoid	00:09:12	00:11:13
Tanh	00:08:53	00:09:50
Relu	00:06:36	00:09:52

It seems like the changes in efficiency are dependent on the type of information that is fed in. In testing on one of the labs, it seemed to favor SDG and flat line when Relu was used, but in our project, it seemed to favor Adam and only slow down without relu instead of flatlining.

Relu, Adam, and 20 Neurons on first layer

HH:MM:SS	1 Hidden Layes	2 Hidden Layers
10 Neurons per layer	00:06:36	00:06:00
20 Neurons per layer	00:05:56	00:05:16
100 Neurons per layer	00:04:11	00:05:02

The efficiency seems to change almost randomly with changes in neurons and layers.

Task Division and Project Reflection

Kevin and Cody were responsible for getting the data to work on Google Colabs

We were having issues getting the data to process correctly, since 2 of us are stuck on laptops, so we all worked through the beginning separately. Two of us (Cody and Kevin) worked on getting Google Colab to work properly with the code, while the other marched ahead trying to figure out how to process the data. Once we got to the same point, we shared solutions and combined our work, which really helped us all learn and kept us from getting stuck. After we got the data working correctly (Brian) got the TF-IDF and Tensorflow working, (Cody) got the best fit loop working, and (Kevin) worked on the Report/charts, however the report and charts were giving us a hard time. (Brian) ended up fixing our lift chart. Then (Kevin) worked on figuring out the report.

The range we have for the prediction is pretty large, so we try to increase the review count to at least 200. The range for prediction is reduced which means the model is more accurate. The reason for this might be because the model has more data to train for the business.

It was a rocky start for the beginning of the project due to the limited resources from our computers. Two of us had to use google colab which made it easier to work but was a bit confusing at the start. The communication was great during the project, everyone was available and helpful as well. There were some improvements with the data, the neurons and layers were optimized to run through the data quicker. Google collabs did crash a lot when we would run things so it took a lot of time to see where we were.