Abby Fisher

August 2, 2016

Professor Nicholas Flann

Independent Study

Full Report

News and Neural Networks (Jasper, 2015)

This program was built to help predict stock prices by scraping Google News with certain queries and dates. These dates and queries would be the building blocks for the neural networks that would be trained to gain better predictions every time there was a new query. I went through the code in a very thorough way in order to gain an understanding of how it works and see if there would be a possibility in changing the queries to not just predict stocks but other things.

There are five main files that I spent most of my research in. Each had a very distinct purpose. Every file within the code was a building block for the next step. The first would scrap the data from google and store it. That data would then be processed and used to build the neural network that was then trained and used to build the predictions of the stocks.

Starting with scrape-data.js, this is the file that contains the code that scrapes Google News. How it works: there is a list of queries which include the ten highest stock names and creates a date range to search through. It creates a URL for each of these queries and each date to gain a dataset that has all of the articles from that day that include the query word within it. From there it stores each of these data sets onto a database. The database that was used is Firebase, it is cloud based web database that anyone can easily include into either a web application or a program like this one (Wiki, n.d.). From Firebase, the data can then be exported into JSON format to create an export.js file that the next file (process-data.js) will use to perform its functionality.

Process-data.js takes the export files and processes the data. It takes every word from the headlines and bodies of each article and coverts them into one large string. It takes all the stop words, words which are taken out of searches that do not put much meaning into the search (Wiki, Stop Words- Wikipedia, n.d.), out of this large string of words. It then counts how often each word is used within all the headlines and bodies and sorts the words by how often they are used. Once that is finished it eliminates the highest and lowest used words to get a better range to use against the original data to see where these words are showing up. If the word is within the same spot as the original data, then an object is pushed into an array specifying that the word is there. It then uses clean-change-data.js to see if the symbol and date of that article is present. If it is found and the corresponding number is 0 or 1 then it will be a part of the training.js file.

Clean-change-data.js is the only file I could not figure out exactly how it was created except for it being the dates of historical high stock prices for each of the top ten stock markets. I never found where it is created within the program.

The training.js file is the core file that is needed for this project and it is used within the nn.py file. It is the vectorized dataset that PyBrain uses to train and then predict the outcome. “PyBrain is a modular Machine Learning Library for Python. Its goal is to offer flexible, easy-to-use yet still powerful algorithms for Machine Learning Tasks and a variety of predefined environments to test and compare your algorithms.” (Tom Schaul, n.d.) The PyBrain function that is used here is Supervised Regression Training. It takes in the length of the training.js data and then a target so it can then do the task of inferring a function from labeled training data. (Wiki, Supervised Learning, n.d.) With this a training dataset and a testing dataset are created. It then builds the network using training.js data that will train until it converges on the testing and training datasets. From there is finds the accuracy and outputs it to the console. This accuracy is the prediction of the stocks for that dataset that was created clear back with scrap-data.js.

# Works Cited

Jasper, M. (2015, December 16). *News and Neural Networks* . Retrieved from http://mdjasper.github.io/google-news-neural-network/

Tom Schaul, J. B. (n.d.). *Pybrain*. Retrieved from Pybrain: http://pybrain.org/

Wiki. (n.d.). *Firebase-Wikipedia*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Firebase

Wiki. (n.d.). *Stop Words- Wikipedia*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Stop\_words

Wiki. (n.d.). *Supervised Learning*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Supervised\_learning