CS230

Predicting Stock Market Movements Using Global News Headlines

Jason N. Kurohara
Computer Science
Stanford University
jkuro@stanford.edu

Joshua R. Chang
Computer Science
Stanford University
jrchang@stanford.edu

Callan A. Hoskins
Computer Science
Stanford University
chosk@stanford.edu

Abstract

Financial markets are inherently volatile and reflectc diverse macroeconomic and microeconomic trends. In this research project, we use natural language processing in conjunction with a deep neural network to predict the daily change in price of the Dow Jones Industrial Average. After trying a variety of models with input data (previous days' stock price changes, news, and weather), our most successful model performed significantly better (61.31%) than random guessing for predicting whether the stock market would rise or fall on a given day. While the stock market is inherently stochastic, our study offers interesting insights into how global news headlines can influence stock price movements.

1 Introduction

Much of the world's wealth is located in large financial markets, which allow individuals and organizations to purchase shares of companies on a public exchange. The price of a company's shares is subject to a wide range of variables–some intuitive and some not. The task of using these inputs to predict a stock's price movement is a trillion-dollar industry.

For most of the history of the stock market, investment decisions have been made by educated and informed humans. Though the stock market has displayed a consistent overall trend for the duration of its existence, results of attempts to 'beat' the stock market have been mixed. It is unclear which variables influence stock prices and by how much. Microeconomic trends and consumer preferences often determine the price of a single stock, but these factors can be extremely difficult to predict; even experts sometimes perform this task very poorly.

For this research project, we focus on predicting the price of the Dow Jones Industrial Average (DJIA), a price-weighted index of thirty large American companies that represents the state of the domestic economy as a whole. Since significant changes in the DJIA are caused by factors that affect each of its constituent stocks, its price reflects macroeconomic trends. Previous research has shown that macroeconomic trends can be predicted based on daily news, and further research shows that news headlines alone can be used to achieve the same results.

We have constructed a neural network that predicts the float amount by which the DJIA rises or falls based on a textual input consisting of twenty-five news headlines concatenated into one string per day.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Related work

Past work has used text drawn from various sources to generate predictions about stock market movement. One team used Twitter [1] to extract public opinion and was able to use a neural network to transform these opinions into rather accurate predictions about the DJIA closing price. Other leading researchers have experimented with event detection algorithms [2] and other natural language processing techniques to create rich input for neural networks that aim to learn their relationship to stock prices.

Others have focused on developing a neural network architecture that is best able to convert these inputs into accurate predictions. Since macroeconomic trends are closely tied to public opinion and sentiment, effective sentiment analysis is key for our objective. One promising model uses a convolutional neural network in conjunction with a recurrent neural network to conduct sentiment analysis of short texts [3]. This work is applicable to our objective because it extracts sentiment from documents that are too brief to contain much context, like news headlines.

3 Dataset and Features

We use a publicly available dataset of DJIA prices and daily news articles available from Kaggle [4]. The DJIA data includes the opening and closing price of the DJIA, plus other metrics such as its daily trading volume, daily high, and daily low, for every market day from August 8, 2008 through July 1, 2016. This amounts to 1988 data points.

Date	Open	High	Low	Close	Volume	Adj Close
7/1/16	17924.2402	18002.3809	17916.9102	17949.3691	82160000	17949.3691

Figure 1: This is an example of all the DJIA metrics included for one day in our dataset.

The news data is comprised of the 25 most popular news headlines for every day (including weekends) in the same range, sourced from Reddit News. The 25 news headlines for each day are those that received the most votes from Reddit users.

For example, corresponding to the price change on July 1, 2016, there are 25 news headlines ranging from "The president of France says if Brexit won, so can Donald Trump" to "Spain arrests three Pakistanis accused of promoting militancy".

Our output is the normalized daily change in price of the DJIA. We calculate the change in DJIA price by subtracting opening price from the closing price, then normalize the data by subtracting the mean and dividing by the standard deviation of the DJIA daily price changes.

We allocate 80% of our total data to the train set, 10% to the development set, and 10% to the test set.

Another key feature to each day in our dataset is the previous 5 days' stock price changes. The intuition behind this feature is that stock prices depend not only on the daily news headlines, but also recent previous stock price movements. The number of previous stock price changes is a hyperparameter that we tuned appropriately.

We preprocess the news headlines by converting them to lower case, replacing contractions, removing unwanted characters, and removing stop words that are irrelevant for this natural-language processing task. Next, we create embeddings for each sentence using pre-trained, 300-dimensional GloVe embeddings for all words with corresponding GloVe embeddings, and random embeddings of the same size for words not found in GloVe's vocabulary. The GloVe algorithm is used to create a vector space representation of words to capture semantic and syntactic meaning. For more information on GloVe, see [5].

To structure our previous 5 days' stock price changes, we create a two dimensional array with rows that represent each day in our dataset. Every row then contains the previous 5 days' stock price changes. Thus, we end up with a (1988,5) array. For the first 5 days, we zero-pad the non-existent entries. For instance, from August 8-13, 2008, this input data would be:

0	0	0	0	0
-1.47363277	0	0	0	0
-1.47363277	-1.79395714	0	0	0
-1.47363277	-1.79395714	-1.85602318	0	0
-1.47363277	-1.79395714	-1.85602318	1.18877929	0
-1.47363277	-1.79395714	-1.85602318	1.18877929	4.208866

Figure 2: This is a sample of the first 5 rows of our two-dimensional array. The first row corresponds to the previous stock price changes on August 8, 2008 which is all zeros since this is the beginning of our dataset. The sixth row corresponds to August 13, 2008 which contains the five previous stock price changes.

4 Methods

As our baseline, we simplified our model into a sentiment analysis task. Because we are working with brief news headlines, we used Wang et al's approach to sentiment analysis [3], feeding sentence embeddings into a CNN to learn high level features, and then feeds these results into a RNN for sentiment analysis of short texts or news headlines. The output of the RNN is fed into a fully-connected network, which aims to learn its relationship to sentiment. We used the mean squared error as our loss function and Adam optimization algorithm. We found an implementation of this network on Github [6] and were able to use it as our base model.

More specifically, for our RNN, we use a Long Short-Term Memory (LSTM) that learns long-term dependencies. Rather than using a traditional feed-forward network, an LSTM detects features from an input sequence and carries the information over a long distance.

In addition to the sentiment analysis on the textual inputs, we also included the change in price of the stock market for the previous five days. We feed the previous five days' price changes into an LSTM, which then becomes another input to the fully connected network. Our model is built on the Keras [7] and scikit-learn [8] machine learning frameworks. The full model architecture is shown in *Figure 3*.

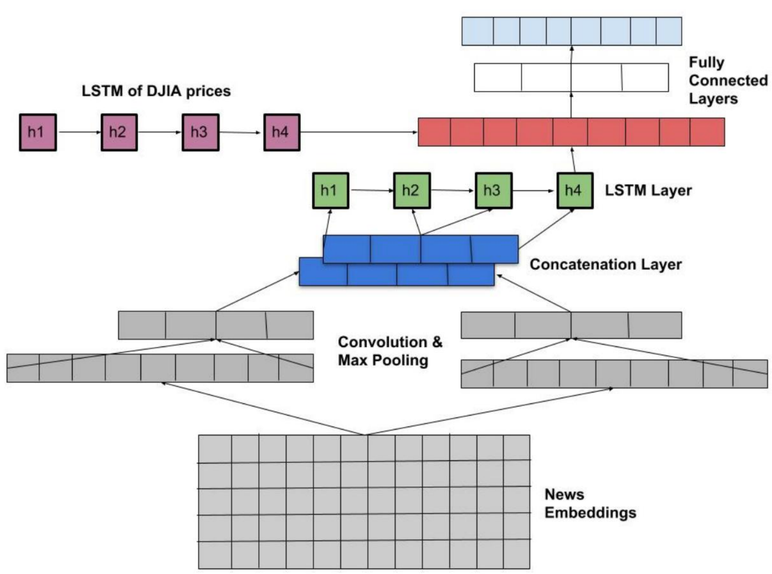

Figure 3: Model architecture

5 Experiments

Our baseline model relied on news data as the sole source of input. With this model and the optimal hyperparameters found by [6], we were able to correctly predict whether the stock market would rise or fall 54% of the time. This was our starting point.

The baseline model's hyperparameters include different learning rates, the value of p in dropout, the number of layers to include in the fully-connected layer, the number of elements in the LSTM, and the number of convolution layers on the textual data. We used both grid search and random search to find the best possible values of these hyperparameters.

In an attempt to put the performance of our model in perspective with that of other models, we created a simple logistic regression model that uses NYC weather data to predict whether the price of the DJIA rises or falls. With this simple (and logically unrelated) data, we were able to predict positive/negative price movements in the DJIA with 52% accuracy–slightly better than flipping a coin.

On our first iteration of a model including both previous stock prices and global news headlines, we achieved 55.28% accuracy with a mean absolute error of 71.40. This model predicted price changes more accurately than our baseline did, which indicated that adding previous stock prices improved our model. We then tuned our hyperparameters using grid search, restricted our hyperparameter scope using random search, and finally implemented a last iteration of grid search on an even narrower scope of hyperparameters gained from the previous random search. On this last grid search, we iterated through four different depths of our network (1 to 4 fully connected and convolutional layers), two different hidden dimensions (128 and 256) and five dropout rates (0.35, 0.37, 0.39, 0.41, 0.45) for a total of 40 models in order to determine the one with the best performance.

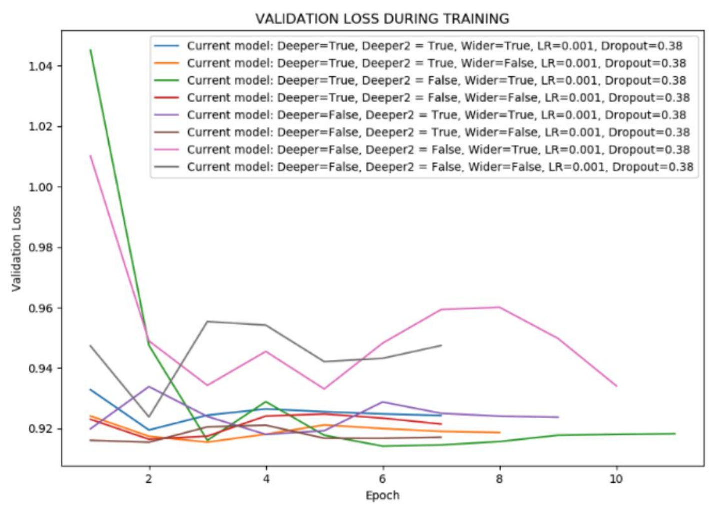

Figure 4: By varying the number of layers and hidden dimensions, we tested our models and observed their validation loss. After training on less than 20 epochs using early stopping, our validation loss still remained relatively high.

6 Results

Our best model used 4 fully connected layers, 4 1D convolutions for the textual news data , and 128 hidden dimensions in the fully connected layer, achieving 61.31% accuracy in predicting the direction change in the DJIA. This was a 7.03% increase over our baseline model. Our model was well-fit to the data, since the training accuracy for our our best model was 58.29% while our test accuracy was 60.31%. The results are summarized in *Figure 5* and the confusion matrix for our best model is shown in *Figure 6*.

One notable trend throughout our experiments is the reasonably high mean absolute error, which provides insight into the average magnitude of error in our predictions regardless of direction. Our

Model	Accuracy	RMSE	MAE
Weather Logistic Regression	51.80%	1.0948	62.63
News with CNN & LSTM *(Baseline)*	54.52%	0.1612	169.32
News and Previous Stock Price *(1 FC layer, 1 CNN, 2x Hidden Dims)*	55.28%	1.0068	71.40
News and Previous Stock Price *(2 FC layers, 2 CNN, 1x Hidden Dims)*	58.28%	0.9776	68.27
News and Previous Stock Price *(3 FC layers, 3 CNN, 2x Hidden Dims)*	61.31%	0.9721	68.67
News and Previous Stock Price *(4 FC layers, 4 CNN, 1x Hidden Dims)*	61.31%	0.9707	69.87

Figure 5: Weather model, baseline model, and best model from the four architectures adjusting the number of layers.

model is best suited for a binary classification task of predicting a rise or fall in the DJIA and struggles with determining the actual magnitude of DJIA price changes.

N = 199	**Predicted: Fall**	**Predicted Rise**
Actual: Fall	TN=89	FP=23
Actual: Rise	FN = 54	TP=33

Figure 6: Confusion matrix for binary classification task

The most similar academic paper we found that performs a similar binary classification task is by Manuel R. Vargas, Beatriz S. L. P. de Lim, and Alexandre G. Evsukoff, who published a paper on a similar topic at the 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA) [9]. They achieved a test accuracy of 64.21%, which we will treat as the state of the art model. In comparison, we achieved 61.31% for a similar binary classificaiton task.

7 Conclusion

Our model was able to outperform our baseline at the task of predicting whether the DJIA would rise or fall each day. We started at 54% accuracy and eventually got to 61.31%–a significantly better difference than a random coin flip.

Our model reacted positively to the addition of historical market price data, showing that a basic model with textual news input can be bolstered by incorporating additional relevant input sources. This makes sense because the stock market is subject to many factors, any of which can influence the market's price. Historical price data is just one, and there are many others we could incorporate into our model. Some interesting ideas include tweets from the presidential Twitter account and price information from foreign markets. It could also be interesting to apply this model on a company-specific basis, feeding it news headlines focused on a single company. We suspect that this data would be even more relevant because it is directly relevant to the status of the company (which should be reflected in its stock price). From an architecture perspective, we would like to refine our architecture to test deeper models with more layers, and adapt our evaluation metrics to analyze the magnitude of the stock prices rather than just their change.

Contributions

Josh: Tested models and performed hyperparameter searches. Implemeted error analyses. Final report and poster.

Callan: Baseline model, plots and figures. Assisted with model development. Final report.

Jason: Implemented changes for architecture in Keras. Implemented random search and testing features for hyperparameter tuning. Assisted in dataset cleaning of news headlines and DJIA prices. Final report and poster.

Github Repository

https://github.com/jasonkurohara/cs230project

References

[1] Xiaojun Zeng Johan Bollen, Huina Mao. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[2] Ting Liu Junwen Duan Xiao Ding, Yue Zhang. Deep learning for event-driven stock prediction. 2014.

[3] Zhiyoung Luo Xingyou Wang, Weijie Jiang. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. *COLING*, 2016.

[4] Aaron7sun. Daily news for stock market prediction, 2016.

[5] Christopher D. Manning Jeffrey Pennington, Richard Socher. Glove: Global vectors for word representation, 2014.

[6] David Currie. Predicting-the-dow-jones-with-headlines. `https://github.com/Currie32/Predicting-the-Dow-Jones-with-Headlines`.

[7] Open Source Francois Challet. Keras deep learning framework.

[8] Open Source. scikit-learn machine learning framework.

[9] M. R. Vargas, B. S. L. P. de Lima, and A. G. Evsukoff. Deep learning for stock market prediction from financial news articles. In *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pages 60–65, June 2017.