Predicting US Stock Market Movement from Political Tweets

Chenghao Peng: chpeng@staford.edu; Brian Wai: brianwai@stanford.edu

Fall 2019, CS 230: Deep Learning

What's the issue?

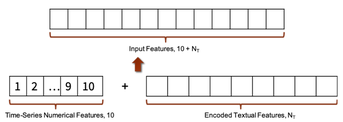
Due to the unpredictable nature of the current POTUS, stock market often sees high volatility. We took tweets coming out of account ***@realdonaldtrump*** and mapped it with intraday 1-min S&P 500 index value since the day Mr. Trump won the 2016 election, trained a few different models to predict how his words on twitter may affect US stock market, using S&P 500 as an indicator.

Where's the data come from?

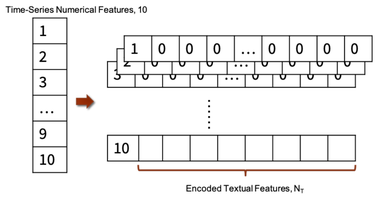
- Twitter Data: We acquired all tweets (original and retweet) from account ***@realdonaldtrump*** between **11/08/2016** and **12/31/2018** through API calls we built under the Twitter Developer API Tools. The dataset contains the following information: Date/Time of the tweet, Type (Retweet/Not), Content, Number of likes and Number of retweet.
- Stock Market Data: We acquired S&P intraday 1-min movement data from Github Repo: *https://github.com/FutureSharks/financial-data*. Dataset consists the following info: Date/Time, Open, Close, High, Low. All indicating the corresponding index value of the minute.

Feature Selection

We selected the previous 10 index close values (time-series) combined with encoded tweet at the time of the event as input features. Features were normalized using MaxMin. **GloVe** with 200 dimension and **One-Hot** were selected as encoding method. We also used two different feature manipulation to accommodate different models.



Feature Manipulation – Method 1



Feature Manipulation – Method 2

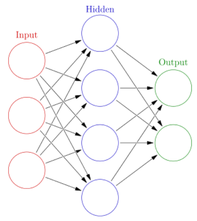
What are the models?

- **Linear Models:**

Served as baselines for both classification and regression. Linear models are the basics of machine learning. Serving as a baseline in this case.

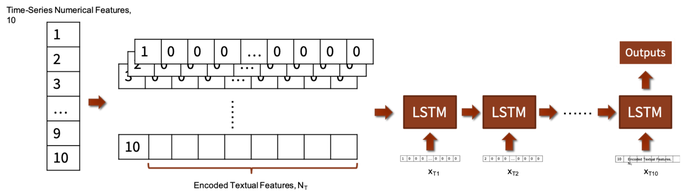
- **Artificial Neural Network (ANN):**

Fully connected neural networks are neural networks where neurons are connected to each other in various patterns, to allow the output of some neurons to become the input of others.



- **Long Short-term Memory (LSTM):**

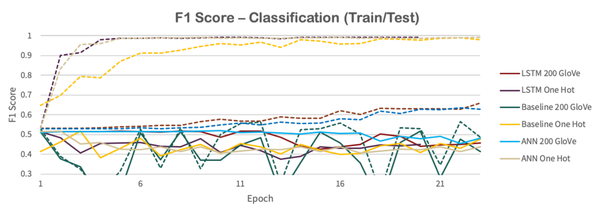
Long short-term memory(LSTM) algorithm is a special type of Recurrent Neural Network(RNN). The LSTM/RNN model is a sequential model that address the strong correlation between inputs such as time series.



What are the results?

- **Classification Models:**

None of the classification models turned out to have produced valuable results. While most of the model we tested converged after 10 epochs, we observed huge biases between training and test samples. The model with the least amount of over-fitting: LSTM and ANN with GloVe 200 Embedding.



- **Regression Models:**

In contrast, the model handles the prediction exceptionally well given the evaluation metrics we picked us for this project, which are Mean Absolute Error (MAE) and Root-Mean-Square Error (RMSE).

Model	R-Squared		RMSE		MAE	
	Train	Test	Train	Test	Train	Test
Baseline-OneHot	0.99	0.99	0.33	7.36	0.32	2.04
Baseline-GloVec200	0.99	0.99	2.94	3.81	1.08	1.26
ANN-OneHot	0.99	0.98	87.3	423.9	7.22	15.98
ANN-GloVec200	0.99	0.99	90.25	100.26	7.08	7.82
LSTM-OneHot	0.99	0.99	4.14	6.82	1.41	1.78
LSTM-GloVec200	0.99	0.99	27.68	21.2	2.63	2.01

What did we learn?

- Deep learning regression models can be directly applied to NLP related problems, the outcomes turn out to be MUCH better than creating labels from numerical values.
- In more complex system, LSTM for example, One-Hot encoding turns out to have better result than GloVe embedding. One possible reason is that the complex reason can fully utilize the full feature set One-Hot maintains.
- In the regression problem, the baseline linear model did exceptional well in term of performance, especially with the GloVe features. This could possible mean that our labeling algorithm has some bias that we did not consider thoroughly.
- In all three models, One-hot variable tends to over-fit the model vs. GloVec. This is a known benefit of the word embedding techniques. And we verified this conclusion.

What would we do next?

- Gathering more tweets from other politicians from both sides of the government.
- Training word2Vec embeddings with the collected/sample tweet to gain more accurate relations between words.

Acknowledgment

We thank Conor Smith, our project TA in CS230 at Stanford University for comments and suggestions along the way.