Using Deep Learning to Predict Stock Market Movements

Evan Rosenman (*rosenman@stanford.edu*)

Task

Goal: Predict how stocks move over a ten-day time horizon, using historical performance data and relevant news articles.

Background: Predicting stock performance, and incorporating outside information like tweets, is a growing problem in the machine learning literature [1, 2, 3].

Methods: We make use of feed-forward and sequence-based models, incorporating summary statistics about relevant news articles in some of our models.

Results: Best performance is achieved by a regularized feed-forward neural network with two hidden states, though sequence-based models show future promise.

Data + Features

Kaggle "Two Sigma: Using News to Predict Stock Movements"

Market Data *(4072956 x 16)*

Day	Asset	10-day Previous Returns	Trading Volume	...	*10-day Future Returns*

News Data *(9328750 x 35)*

Day	Subjects	Length	Sentiment	Novelty	...

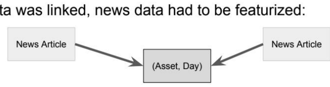
Goal was to predict 10-day future market-residualized returns for each asset and each day within the dataset.

Data was split as follows:

2007	2008	2009	2010	**2011**	**2012**	**2013**	**2014**	***2015***	***2016***
Dropped *(financial crisis + aftermath)*				Training Set				Test Set	

This was the result of early experiments demonstrating that models trained on the crisis years performed worse.

When data was linked, news data had to be featurized:



This was done using mean/min/max/std features, resulting in a total of 94 potential features. Only present for ~30% of asset-days.

Performance Evaluation

Used scoring scheme from Kaggle competition:

- Generate prediction $\hat{y}_{ti}$ in [-1, 1] representing confidence in market-residualized movements of asset *i* over the ten days following *t*.
- For r_{ti} the true value and u_{ti} a relevancy variable, compute:

$$x_t = \sum_i \hat{y}_{ti} r_{ti} u_{ti},$$

- Lastly, "sigma score" is computed as:

$$\text{score} = \frac{\bar{x}_t}{\sigma(x_t)}.$$

We evaluated on this non-standard loss, but trained our algorithms using a variety of "proxy" loss functions.

Results

Exploring Loss Functions

We initially explored which standard loss function was the best proxy for sigma score, finding cross-entropy performed best.

Model	Features	Loss	Test Sigma
1-Layer NN	Market Data Only	Cross-Entropy	**0.436**
"	"	Sigma-numerator	0.395
"	"	Hinge Loss	0.375
"	"	Squared error	0.250

Exploring Models

Model	Features	Details	Training Sigma *(1.7MM)*	Test Sigma *(0.9MM)*
2-layer NN	Market data + news data	128/16 nodes Dropout = 0.8	0.594	**0.507**
GRU	Market data	32 nodes 50-day windows Dropout = 0.8	**1.840**	0.488
LSTM	Market data	32 nodes 50-day windows Dropout = 0.8	1.770	0.483
1-layer NN	Market data	32 nodes	0.675	0.482
1-layer NN	Market data + news data	32 nodes	0.745	0.430
Logistic Regression	Market data + news data		0.383	0.368

Models

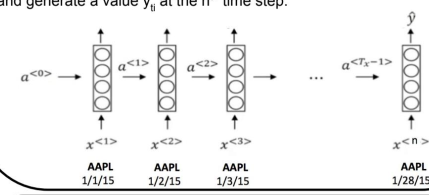
Feed-Forward Models

Covariates x_i put through 1+ layers of non-linearities and linear combinations to yield predictions

$$a_i^{[1]} = \sigma(W^{[1]} x_i + b^{[1]})$$
$$\hat{y}_i = \sigma(W^{[2]} a_i^{[1]} + b^{[2]})$$

Sequence Models ("many-to-one")

Use a sliding window to turn the outcome into a sequence $x_i^{<1>}, \ldots, x_i^{<n>}$ and generate a value $\hat{y}_{ti}$ at the n[th] time step.



Discussion

- News features add some predictive value, but relative sparsity of these features makes modeling challenging
  - Careful featurization and null-filling may be useful
  - Could also model "newsy" stocks and other stocks separately
- Sequence models heavily overfit to training data, and regularization only partially addresses issue
- Cross-entropy appears to be a good proxy for sigma-score loss, but it's possible that other losses would track better
- Stability of Kaggle kernels is a major bottleneck!

Future Directions

- Alternative featurizations of the news data, including using word embeddings with the headlines
- Better regularization schemes to help get predictive value out of the sequence models
- Stacking or ensemble-type approaches that make use of benefits of different models in different situations

[1] Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of computational science*, 2(1), 1-8.
[2] Schumaker, R. P., & Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news. *ACM Transactions on Information Systems* , 27(2), 12.
[3] Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market. In *Big Data*, 2015 IEEE International Conference on (pp. 2823-2824). IEEE.