# Predicting Short Term Stock Prices with News Data and Recurrent Networks

Brian Falkenstein

# Overview

- Want to incorporate features extracted from relevant news articles into a recurrent network to predict short term stock prices
- Specifically, using time steps *t-k* to *t* to predict price at *t+1*

# Short Term Trading / Active Trading

- Position held for no longer than a few days
- Short term market volatility
  - Often caused by political or economic events, key data releases
- Riskier than buy and hold

# Short Term Trading / Active Trading

- Position held for no longer than a few days
- Short term market volatility
  - Often caused by political or economic events, key data releases
- Riskier than buy and hold

**Passenger Files Class-Action Lawsuit Against American Airlines**

**Free Netflix: Petition asks streamers to stop charging due to coronavirus**

**Were Hedge Funds Right About Wells Fargo & Company (WFC)?**

# Dataset

- Want up to date time series stock data
  - **IEXCloud**
    - Limited number of queries
  - 39 companies, 1 month of historical data with 1 day resolution
    - [open, low, high, close]
- News articles from the same time
  - **NewsAPI**
  - Company name as search term
  - First paragraph of article used in embedding

IEX Cloud

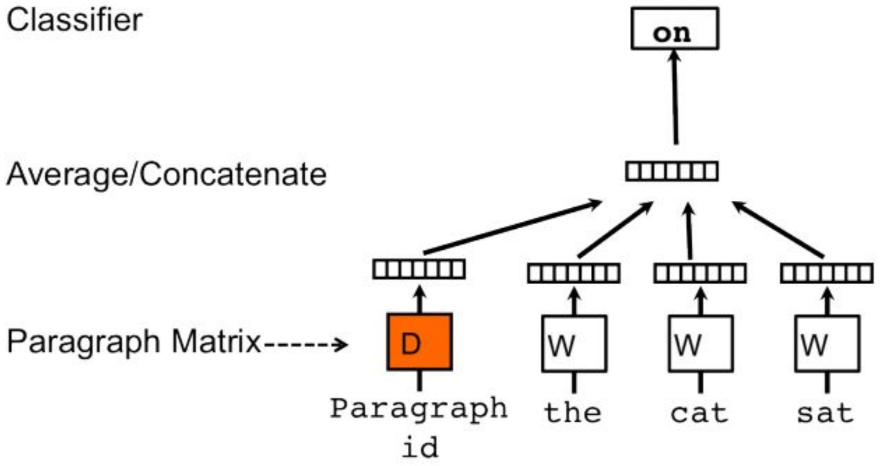
News API

.PHP

```
'As we all navigate life during the pandemic and do our part to flatten the curve by not partaking in non-essential travel, air
lines are updating their policies, providing free flight changes and cancellations for travel directly affected by the pandemi
c. For … [+4211 chars]'
```

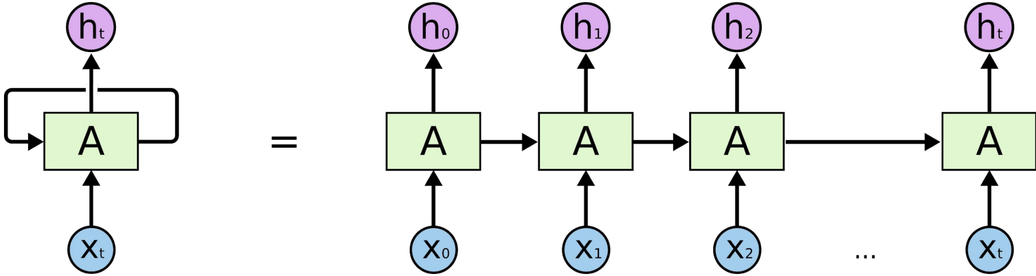
# Document Embedding

- **Doc2Vec**
  - *Distributed Representations of Sentences and Documents* (Le, Quoc and Mikolov Tomas 2014)
  - Learning document representations that aid in word context prediction
  - Success with sentiment analysis task
    - Goal - capture sentiment of relevant  news articles



# Recurrent Modules - RNN

- Daily stock data passed in recurrently
  - Take last output as prediction
- Parameters
  - Hidden dimensionality, sequence length, learning rate
    - Need low LR (exploding gradient)
- Implemented with PyTorch

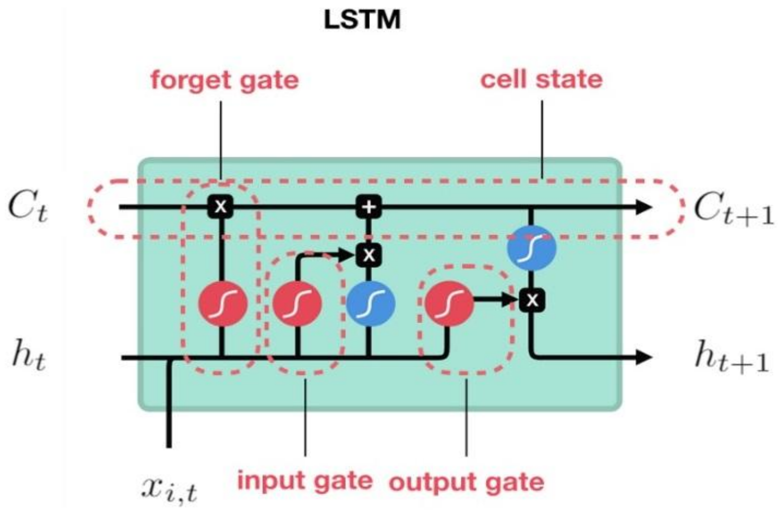



$$out_{t+1} = f(concat(X_{i,t}, h_t), w_1) + b_1$$

$$h_{t+1} = f(concat(X_{i,t}, h_t), w_2) + b_2$$

# Recurrent Modules - LSTM

- Improves on RNN by better capturing long term dependencies
  - More parameters
- Gates allow for control of how much old information is forgotten, and how much new information is transferred forward
- Used already implemented LSTM in pytorch.nn



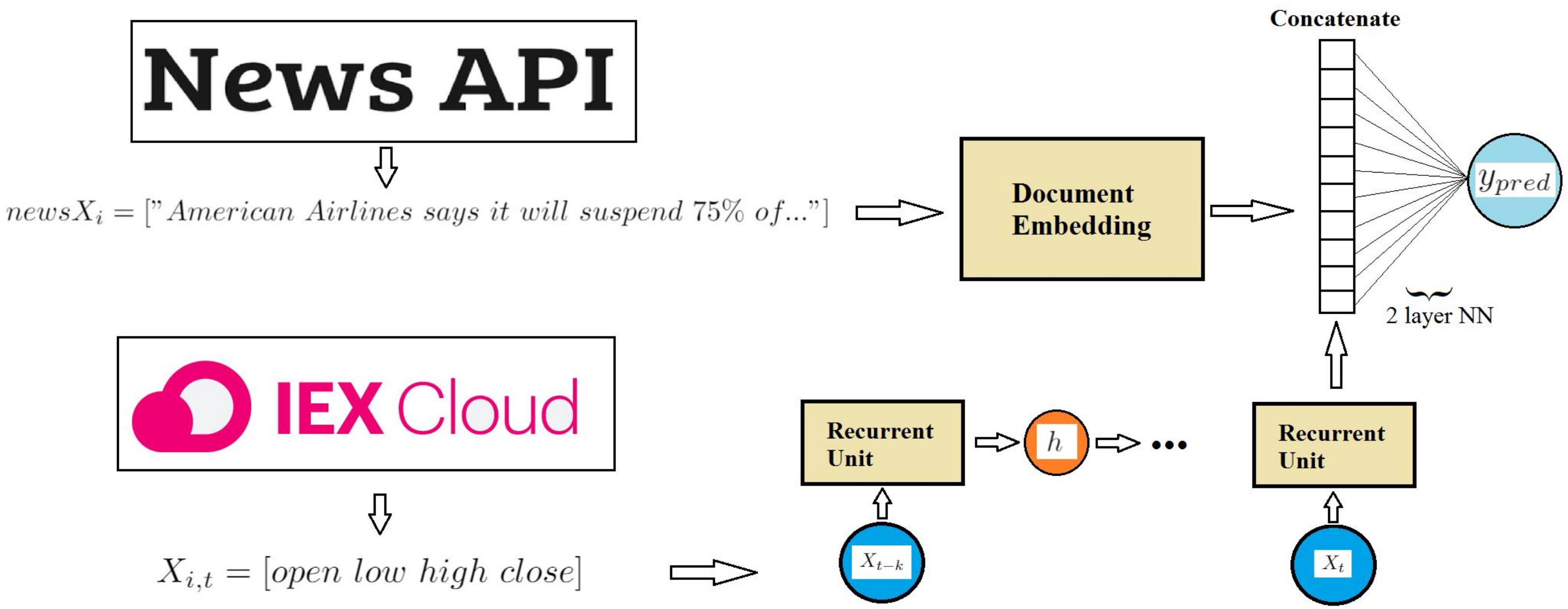

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

# Full Model




$$size(X) = [num\_companies,\ num\_time\_steps,\ num\_features]$$

# Results - RNN

Table 1: RNN Test Errors

| Hidden Dim | Sequence Length | MAE |
|---|---|---|
| 10 | 10 | 1.761 +/- 0.640 |
| **20** | **10** | **1.333 +/-0.536** |
| 30 | 10 | 1.538 +/- 0.822 |
| 50 | 10 | 1.791 +/- 1.215 |
| 20 | 1 | 1.710 +/- 0.8188 |
| **20** | **5** | **1.325 +/- 0.606** |
| 20 | 20 | 2.061 +/- 1.022 |

Table 2: RNN with News Features Test Errors, Hidden dim=20 Sequence length=5

| Document Feature Dim | MAE |
|---|---|
| 5 | 1.722 +/- 0.371 |
| **10** | **1.574 +/- 0.597** |
| 20 | 1.726 +/-0.580 |

# Results - LSTM

Table 3: LSTM Test Errors

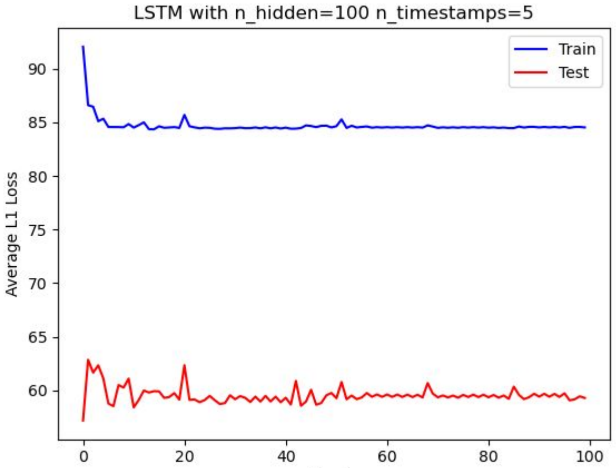
| Hidden Dim | Sequence Length | MAE |
|---|---|---|
| 10 | 10 | 107.14 +/- 68.427 |
| 50 | 10 | 62.037 +/- 44.274 |
| **100** | **10** | **55.155 +/- 25.839** |
| 100 | 1 | 72.409 +/- 65.199 |
| 100 | 5 | 66.954 +/- 60.184 |
| 100 | 20 | 88.994 +/- 67.606 |

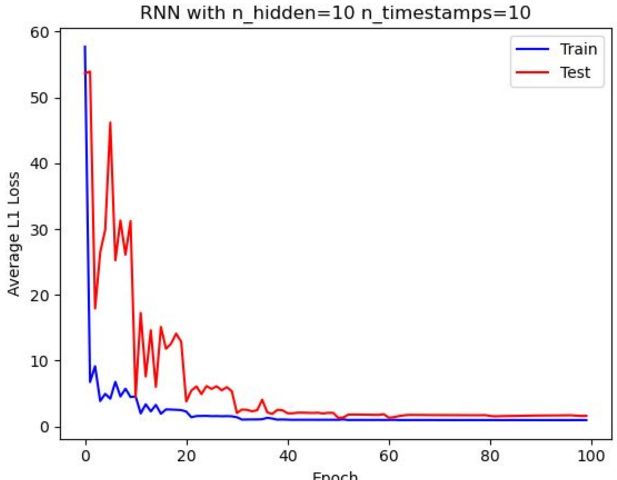
Table 4: LSTM with News Features Test Errors, Hidden dim=100 Sequence length=10

| Document Feature Dim | MAE |
|---|---|
| 5 | 56.633 +/- 19.001 |
| 10 | 49.607 +/- 11.433 |
| **20** | **48.870 +/- 10.467** |

# Conclusions

- LSTM results BAD
    - Likely need a much larger dataset
- Effects of news features inconclusive
    - Could try other embedding methods (LSA, BOW, ...)
- Better metric would be to train the model to make decisions on buying and selling, and tracking profit/loss
    - Compare to other methods (buy and hold)
    - Reinforcement learning



# Thank you!