**Application of RNN for Algorithmic Stock Trading**

Mohan Patnam, CSCIE89 (Deep Learning)(mpatnam@gmail.com)

**Problem Statement:**

Application of Recurrent Neural Nets for algorithmic stock trading with the focus on short-term price prediction using Tensorflow/Keras API for Deep Learning. A case study of single/multi-variate time-series analysis for a chosen tech stock and its industry sector (technology) and the stock market index (S&P500) as a whole. Perform various experiments ranging from prediction of stock price movements (up/down) in the near term to predicting stock price itself using multiple variables.

**Stock**: MSFT (Microsoft Inc.)

**Sector**: XLK (Tech Sector ETF) – MSFT is 11.80% of XLK portfolio weight

**Stock Index**: SPY (representing S&P 500 companies as whole)

**Technology Overview:**

A RNN based deep learning methodology for stock price prediction. Built using Python and Deep Learning API (TensorFlow). The system captures effects of sector and stock market performance as a whole on the performance of individual stocks. The software built is modular where one can input similar data from a different time-frame easily without changing rest of the code.

**Data Set:**

* Data Source: QuantQuote (<https://quantquote.com/>)
* Historical Intraday Stock Data with one-min resolution for 3 years (2014-2017)  
  <https://quantquote.com/historical-stock-data>
* Data downloaded for S&P 100 stocks, XLK (tech ETF) and SPY.  
  Total data size: 3.3G. Actual data used: ~15M (2014 year)
* Stock Minute Data Sample  
  <https://quantquote.com/sample/SPY_MINUTE_TRADE.csv>
* Data Sample with used columns highlighted -

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date** | **TimeBin** | **Open** | **High** | **Low** | **Close** | **Volume** | **Split Factor** | **Earnings** | **Dividends** |
| 20140102 | 948 | 33.1417 | 33.151 | 33.1323 | 33.1323 | 20582 | 1 | 0 | 0 |
| 20140102 | 949 | 33.1323 | 33.151 | 33.1323 | 33.151 | 6675.17 | 1 | 0 | 0 |
| 20140102 | 950 | 33.1417 | 33.1417 | 33.123 | 33.1323 | 42184 | 1 | 0 | 0 |
| 20140102 | 951 | 33.1314 | 33.1323 | 33.1146 | 33.1323 | 11203.4 | 1 | 0 | 0 |

**Software:** Python 3.5, Jupyter Notebook 4.3.0, Tensorflow API 1.7.0, Keras API 2.1.4,   
Cygwin 2.7.6

**Hardware:** Windows 10 on i5-4460 (4 cores @3.20GHz), 24GB memory

**References:**

1. Data: <https://quantquote.com/historical-stock-data>
2. MSFT and its relationship with XLK   
   <http://portfolios.morningstar.com/fund/holdings?t=xlk>
3. RNN for stock price prediction  
   <https://lilianweng.github.io/lil-log/2017/07/08/predict-stock-prices-using-RNN-part-1.html>
4. RNN Homework sample code.

**Lessons Learned & Pros/Cons:**

Presented a general pipeline of data preparation and normalization for single vs multi-variate time series of stock prices, trained appropriate RNN based models in each case with varying parameters. Performed multiple experiments ranging from predicting stock price direction (up/down) in the near future to the prediction of stock prices (or returns) themselves.

**Pros:**

1. Simple RNN model demonstrating single vs multi-variables as input.
2. RNN based model performed reasonably well to predict the stock price direction (up/down). Loss (MSE) is minimal.
3. RNN model can be easily extended to have multiple inputs to see the effects of related variables. This was demonstrated using multi-variate time series prediction.

**Cons:**

1. Predicting stock prices is not an easy task! RNN based models typically suffer from amnesia as it cannot capture longer term patterns satisfactorily.
2. Need more sophistical learning models involving LSTMs etc to capture longer term relationships that affect stock prices.
3. Stock price movements depends on many other factors including company’s earnings, market sentiment, geo-political and macro-economic events etc. RNN based model can perform better if additional data representing above factors can be provided as input.

**YouTube URLs:**

1. 2-min: <https://youtu.be/u5YZMcV-QVY>
2. 15-min: <https://youtu.be/DrUFj00Sky4>