Application of Deep Learning to Algorithmic Trading

Guanting Chen[1], Yatong Chen[2], Takahiro Fushimi[3]

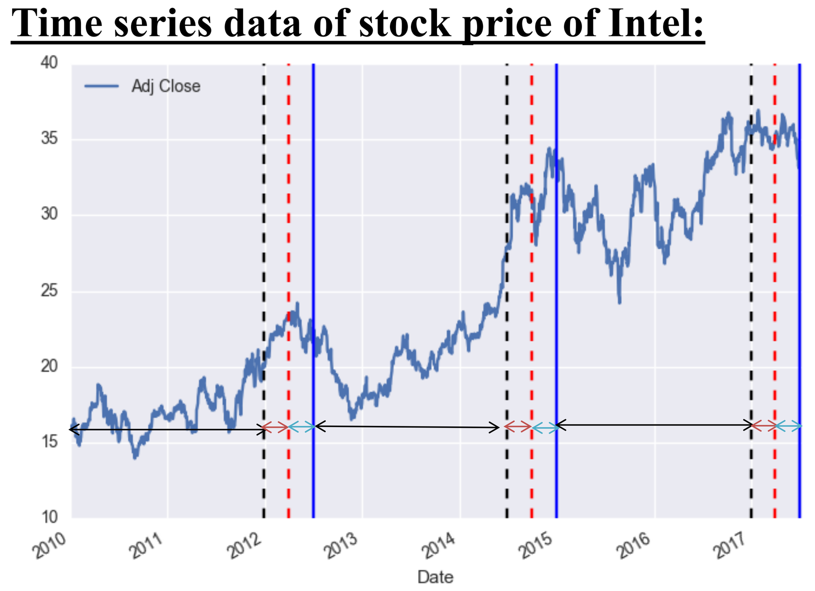
[1] Institutive of Computational and Mathematical Engineering, [2] Civil and Environmental Engineering, [3] Management Science and Engineering, Stanford University

Introduction

Deep Learning has become a robust machine learning tool in recent years, and models based on deep learning has been applied to various fields. However, applications of deep learning in the field of computational finance are still limited[1]. In our project, Long Short Term Memory (LSTM) Networks, a time series version of Deep Neural Networks model, is trained on the stock data in order to forecast the next day's stock price of Intel Corporation (NASDAQ: INTC): our model predicts next day's adjusted closing price based on information/features available until the present day. Based on the predicted price, we trade the Intel stock according to the strategy that we developed, which is described below. Locally Weighted Regression has also been performed in lieu of the unsupervised learning model for comparison.

Dataset

Time series data of stock price of Intel:



Data resource:
https://finance.yahoo.com/

Time periods:
- *Period I: 2010/01/04-2012/06/29*
- *Period II: 2012/07/02-2014/12/31*
- *Period III: 2015/01/02-2017/06/30*

Data set size:
- Train set: 2 years(⟷)
- Dev set: 3 months(⟷)
- Test set: 3 months(⟷)

Trading Framework

Input	Model	Output	Decision
1.Daily Trading Data 2.Technical Indicators 3.Indexes	LSTM Network LWR model	Predicted Next Day's Price	Trading Decision: Buy or Sell

Data Preprocessing and Features

The input features we choose consist of three sets of variables. The first set is historical daily trading data of INTC including previous 5 day's adjusted closing price and log returns, Open/Close price, High/Low price, and trading volume. These variables provide basic information about INTC. The second set is the technical indicators that demonstrate various characteristics of the stock behavior. The third set is index: S&P 500 (^GSPC), CBOE Volatility Index (VIX), and PHLX Semiconductor Sector (^SOX).

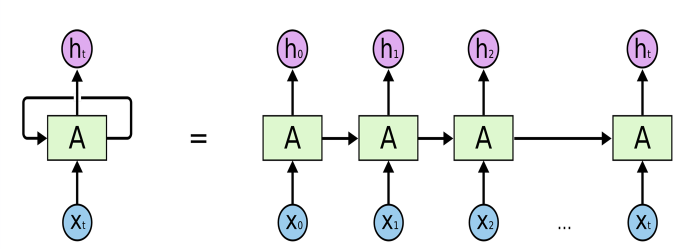
- **Daily Trading Data of INTC**
  - Previous 5 days' prices and log returns
  - Open/Close price, High/Low price, and Trading volume
- **Technical Indicators of INTC(computed based on the trading data)[3]**
  - Rolling Average/Standard Deviation with 5 and 10 days window
  - Bollinger Band: two standard deviations from a moving average
  - Average True Range: a measure to volatility of price
  - 1 month Momentum: the difference between current price and the price 1 month ago
  - Commodity Channel Index: an identification of cyclical trends
  - Rate of Change: the momentum divided by the price 3 months ago
  - Moving Average Convergence Divergence: a display trend following characteristics and momentum characteristics
  - Williams Percent Range: a measure of the buying and selling pressure
- **Index of the Market and Sector**
  - S&P 500, VIX, and PHLX Semiconductor Sector

**All of the variables are scaled between 0 and 1 before we feed them into the model.*

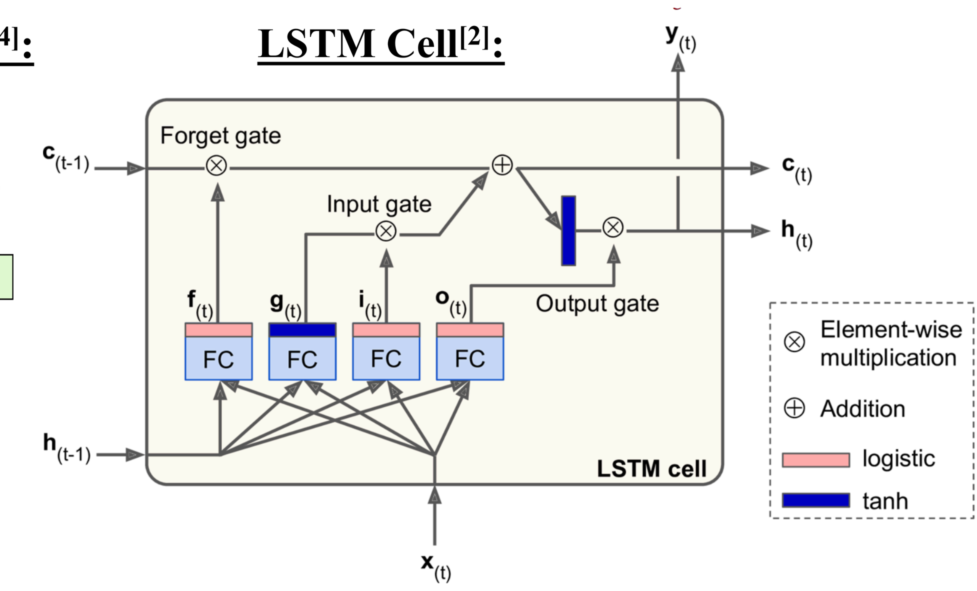
Models

- **Long Short Term Memory[LSTM] Networks:**

Recurrent Structure of LSTM[4]:



LSTM Cell[2]:



Parameters For LSTM:
- Optimizer: *AdamOptimizer*
- Library: *Tensorflow*
- Hidden layers #: 5
- Delay #: 10
- Training Step #: 5000

Mathematical Format for LSTM:

$$i_{(t)} = \sigma(W_{xi}^T \cdot x_{(t)} + W_{hi}^T \cdot h_{(t-1)} + b_i)$$
$$f_{(t)} = \sigma(W_{xf}^T \cdot x_{(t)} + W_{hf}^T \cdot h_{(t-1)} + b_f)$$
$$o_{(t)} = \sigma(W_{xo}^T \cdot x_{(t)} + W_{ho}^T \cdot h_{(t-1)} + b_o)$$
$$g_{(t)} = \tanh(W_{xg}^T \cdot x_{(t)} + W_{hg}^T \cdot h_{(t-1)} + b_g)$$
$$c_{(t)} = f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)}$$
$$y_{(t)} = h_{(t)} = o_{(t)} \otimes \tanh(c_{(t)})$$

- **Locally Weighed Regression[LWR] Model:**

Algorithm for LWR:

(1) Fit θ to minimize

$$J(\theta) = \frac{1}{2}\sum_i W^{(i)}(y^{(i)} - \theta^T x^{(i)})^2$$

which can also be written as:

$$J(\theta) = (X\theta - y)W(X\theta - y)$$

(2) Output $\theta^T x$

Mathematical Format for θ in LWR:

$$\theta = (X^T W X)^{-1} X^T W y$$

Weights for LWR:

$$w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$$

Trading Strategy

Every day in the test set, we use the trained model to compute the predicted price. If the predicted price of the next day is higher than the current actual price, we buy one share of INTC. In contrast, we sell one share of INTC if the predicted price is lower than the current price. The strategy can be described as follow. At time t, buy one share of INTC if $\hat{y}_{t+1} > y_t$ or sell one share of INTC if $\hat{y}_{t+1} \leq y_t$, where $\hat{y}_{t+1}$ is the predicted price by the model and y_t is the current adjusted closing.

Outcome from Experiments

- **Mean Square Error(MSE) of LSTM and LWR:**

Period	Train Error		Dev Error		Test Error	
	LSTM	LWR	LSTM	LWR	LSTM	LWR
I	0.000901216	/	0.005116574	0.106168594	0.006486553	0.053617312
II	0.000504966	/	0.009841842	0.044881932	0.007717514	0.015474667
III	0.000638235	/	0.01662774	0.033309036	0.011428699	0.021470895

- **Daily Returns of Strategy based on LSTM, LWR and Buy-and-Hold for Test Set:**

Period	LSTM Network	LWR Model	Buy and hold
I	0.489266	0.255020	-0.083255
II	0.285829	0.152761	0.103849
III	0.190888	0.054216	-0.065555

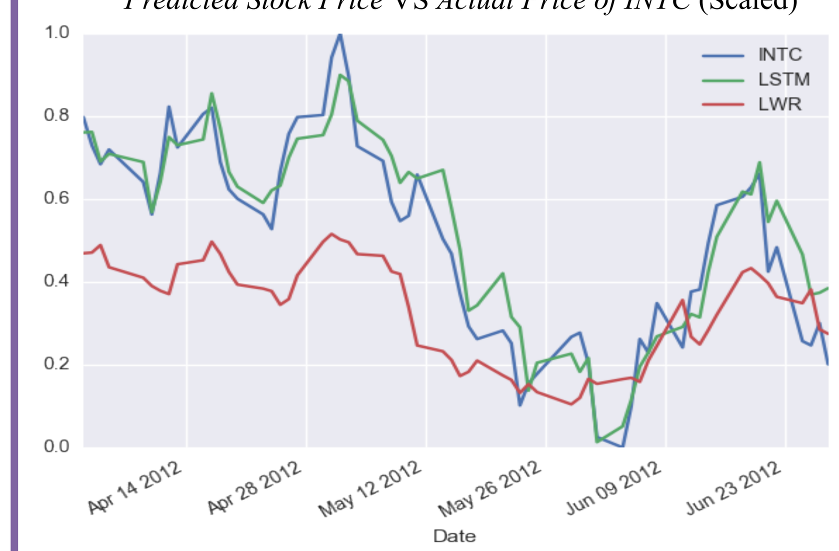
- **Daily Sharpe Ratio of Strategy based on LSTM, LWR and Buy-and-Hold for Test Set:**

Period	LSTM Network	LWR Model	Buy and hold
I	0.538356	0.284026	-0.096499
II	0.253281	0.144540	0.102692
III	0.316494	0.092015	-0.118043

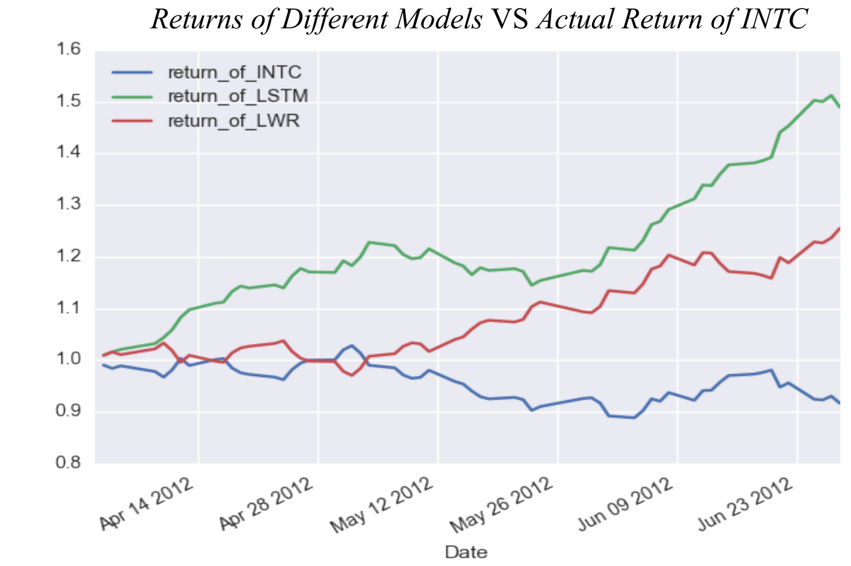
Price Predictions and Return Plots

- **Period I:**

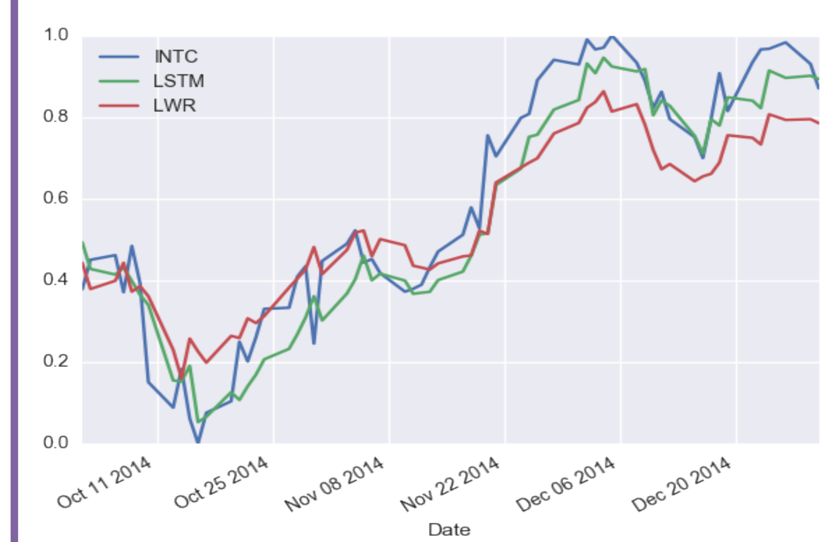
Predicted Stock Price VS Actual Price of INTC (Scaled)

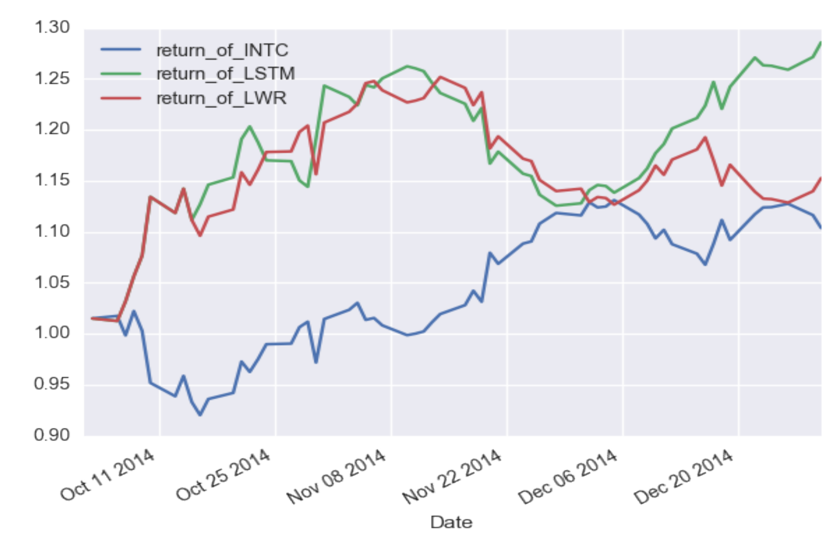


Returns of Different Models VS Actual Return of INTC

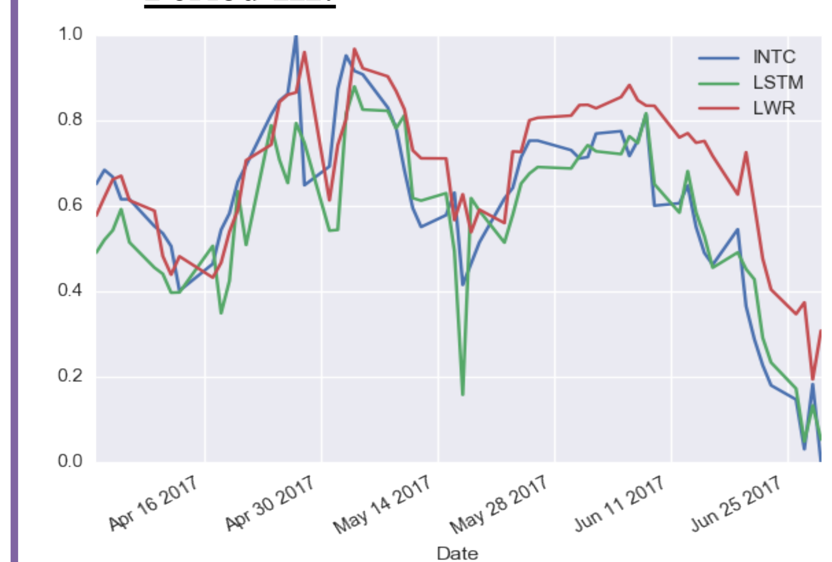


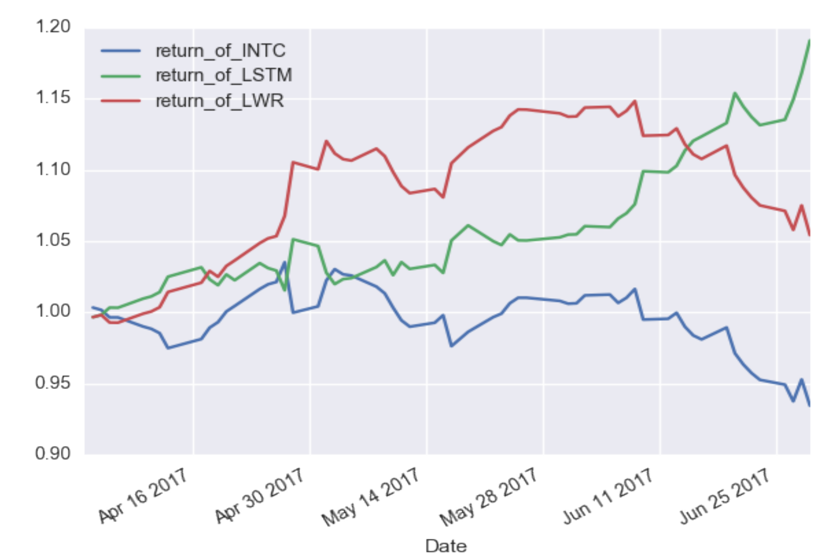
- **Period II:**





- **Period III:**





Conclusions

- Both LSTM Network and LWR model can predict the general trend of INTC stock price, and LSTM outperforms LWR in terms of accuracy and profitability for three periods.
- The performance of LSTM is more robust than LWR. LSTM has smaller MSE than LWR for both Dev Set and Test Set, and it has less deviation in the prediction price plot.
- The strategy based on LSTM yields higher returns and Sharpe Ratio than LWR-based strategy and simple Buy and Hold Strategy.
- However, the prediction by these models become inaccurate when the price changes dramatically.

Outlook

- Tuning hyper parameters and adding regularization term would improve the performance of LSTM.
- Trading strategy with reinforcement learning could generate more stable and higher returns.

References

[1] Andres et al.: High-Frequency Trading Strategy Based on Deep Neural Networks(2016)
[2] Hands-On Machine Learning with Scikit-Learn and TensorFlow
[3] A deep learning framework for financial time series using stacked autoencoders and long short term memory
[4]Understanding LSTM Networks, Colah's blog, http://colah.github.io/posts/2015-08-Understanding-LSTMs/