Stock Price Prediction using Recurrent Neural Network (RNN) Algorithm on Time-Series Data

Israt Jahan
Department of Computer Science and Operations Research
North Dakota State University
Fargo, ND 58105
i.jahan@ndus.edu

Sayeed Sajal
Department of Math and Computer Science
Minot State University
Minot, ND 58707
sayeed.sajal@minotstateu.edu

Abstract

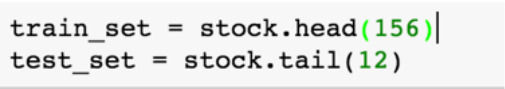
The stock market is generally very unpredictable in nature. There are many factors that might be responsible to determine the price of a particular stock, such as, the market trend, supply and demand ratio, global economy, public sentiments, sensitive financial information, earning declaration, historical price and many more. That's why accurate prediction is one the challenging works. But, with the help of new technologies like data mining, machine learning, we can analyze big data and develop an accurate prediction model without any human errors. In this work, we analyzed the closing price of a stock as sample data. We developed a prediction model using a supervised machine learning algorithm to predict the stock price. In particular, we used a recurrent neural network (RNN) algorithm on the time-series data of a stock. We also cross-checked the predicted closing price with the true closing price. Finally, we found that this model also can be used to predict other volatile financial instruments.

Introduction

Prediction is one of the difficult things where the future is very volatile. By nature, stock market is unpredictable. There are many factors that influences stock price [1, 2]. With the advancement of new technology and statistical tools, many scholars explored the possibility to predict stock price. In 1997, the prior knowledge and neural network was used to predict stock price [3]. Later, genetic algorithm approach and support vector machine were also introduced to predict stock price [4, 5]. Lee introduced stock price prediction using reinforcement learning [6]. In 2008, Chang used a TSK type fuzzy rule-based system for stock price prediction [7]. In 2009, Tsai used a hybrid machine learning algorithm to predict stock price [8]. Over the time, the scholars used to predict the stock price using different kinds of machine learning algorithms such as deep learning [9, 10], extreme machine learning [11] and applied econometric approach using machine learning [12]. In 2018, popular machine learning algorithms were used to predict stock price such as pattern graph [13], convolutional neural network [14], recurrent neural network [15]. Here, we used recurrent neural network on time-series data to predict stock price.

Experimental Results

Here, we used the closing price of Advanced Micro Device (AMD) of 168 working days. The start date is May 2nd, 2017 and end date is Dec 28th, 2017. First 156 days data was used as training data shown in Figure 1 and last 12 days data was used as test data. First of all, we trained the 156 days data to predict next 12 data points. Then we compared the true data and generated data using recurrent neural network. The data is separated in two parts for train and test data.



```
train_set = stock.head(156)
test_set = stock.tail(12)
```

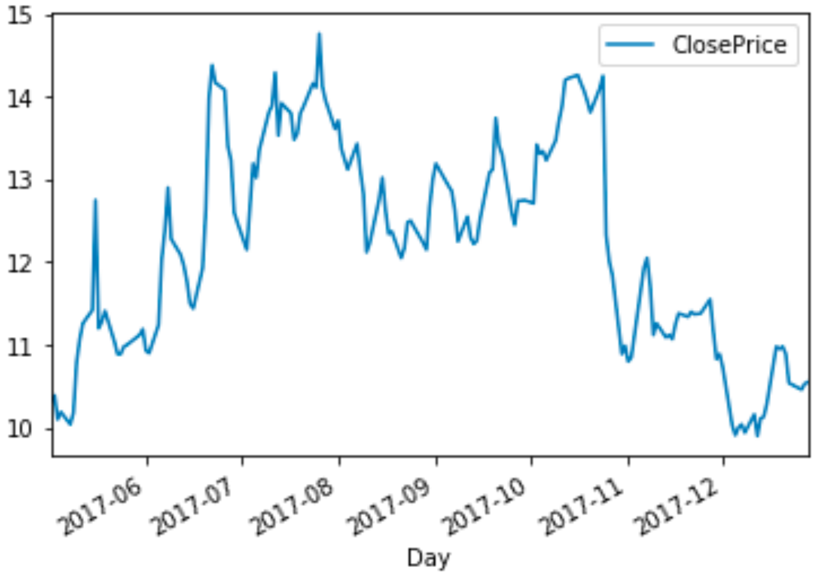
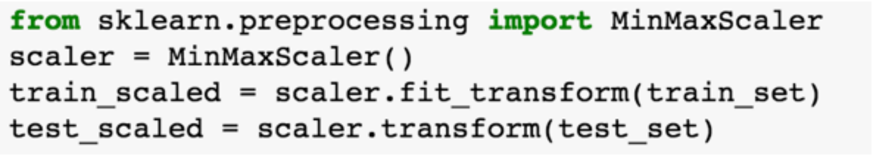


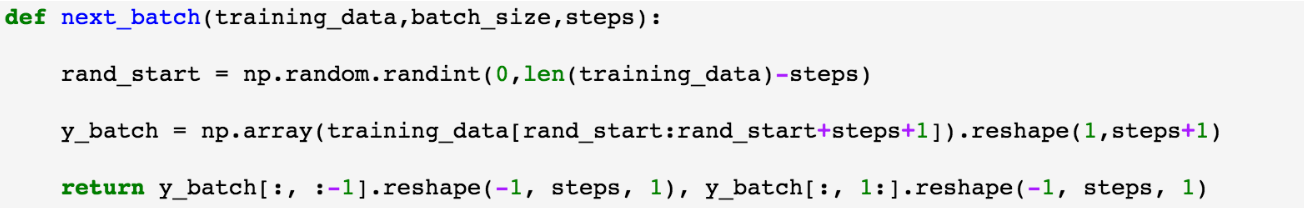
Figure 1: Train Data for first 156 days (AMD closing price).

Here, we used python 3.0 to develop the RNN model to predict stock price. Numpy, pandas, matplotlib, scikitlearn and tensorflow packages in Python were used in this work. Scikitlearn was used for pre-processing the data.



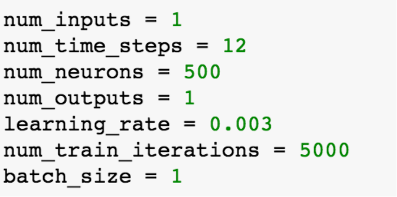
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
train_scaled = scaler.fit_transform(train_set)
test_scaled = scaler.transform(test_set)
```

Then we defined the next batch function to generate the batch.



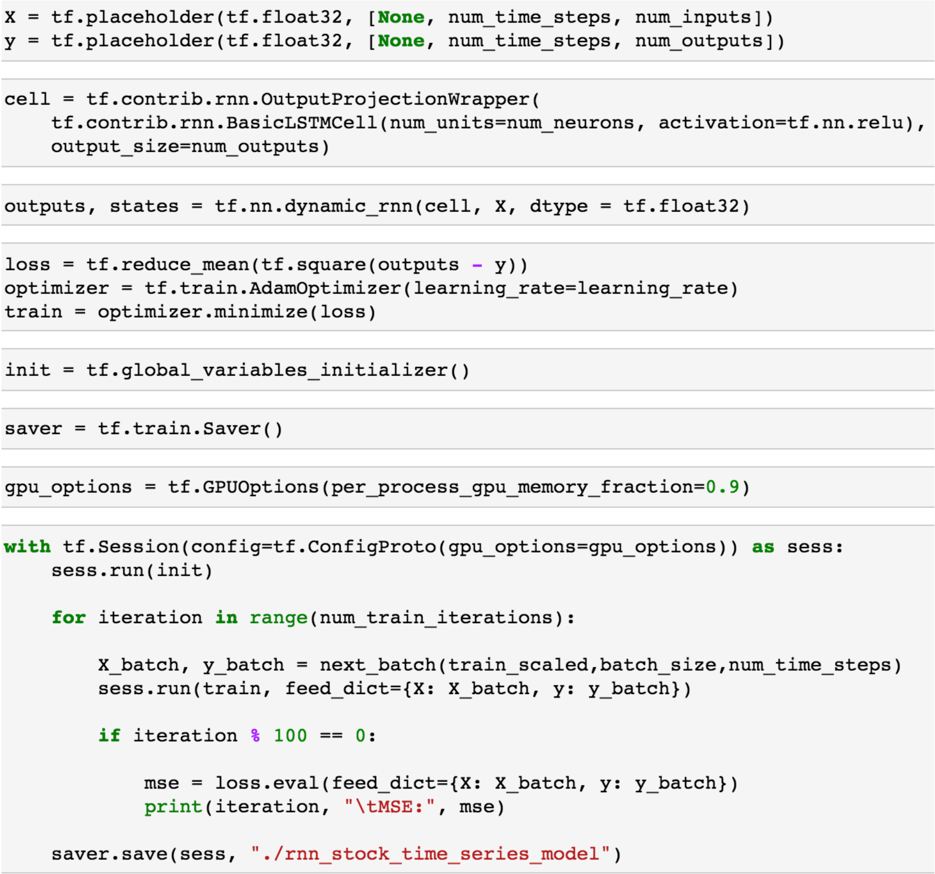
```
def next_batch(training_data,batch_size,steps):

    rand_start = np.random.randint(0,len(training_data)-steps)

    y_batch = np.array(training_data[rand_start:rand_start+steps+1]).reshape(1,steps+1)

    return y_batch[:, :-1].reshape(-1, steps, 1), y_batch[:, 1:].reshape(-1, steps, 1)
```

After successful tuning on the parameters in RNN model, we successfully developed the model which generate the predicted price which is very close the true price. The parameters are given below.



```
num_inputs = 1
num_time_steps = 12
num_neurons = 500
num_outputs = 1
learning_rate = 0.003
num_train_iterations = 5000
batch_size = 1
```

We also have the define the basic LSTM cell and run the session to measure mean-square error (MSE). The python code is given below.



```
X = tf.placeholder(tf.float32, [None, num_time_steps, num_inputs])
y = tf.placeholder(tf.float32, [None, num_time_steps, num_outputs])
```

```
cell = tf.contrib.rnn.OutputProjectionWrapper(
    tf.contrib.rnn.BasicLSTMCell(num_units=num_neurons, activation=tf.nn.relu),
    output_size=num_outputs)
```

```
outputs, states = tf.nn.dynamic_rnn(cell, X, dtype = tf.float32)
```

```
loss = tf.reduce_mean(tf.square(outputs - y))
optimizer = tf.train.AdamOptimizer(learning_rate=learning_rate)
train = optimizer.minimize(loss)
```

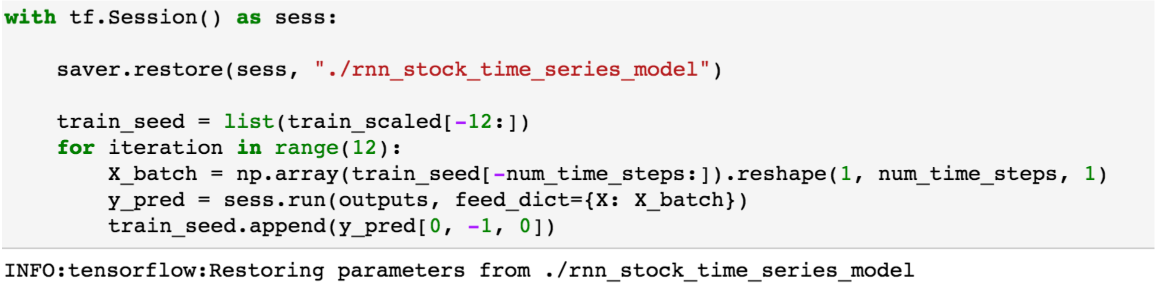
```
init = tf.global_variables_initializer()
```

```
saver = tf.train.Saver()
```

```
gpu_options = tf.GPUOptions(per_process_gpu_memory_fraction=0.9)
```

```
with tf.Session(config=tf.ConfigProto(gpu_options=gpu_options)) as sess:
    sess.run(init)

    for iteration in range(num_train_iterations):

        X_batch, y_batch = next_batch(train_scaled,batch_size,num_time_steps)
        sess.run(train, feed_dict={X: X_batch, y: y_batch})

        if iteration % 100 == 0:

            mse = loss.eval(feed_dict={X: X_batch, y: y_batch})
            print(iteration, "\tMSE:", mse)

    saver.save(sess, "./rnn_stock_time_series_model")
```

At the end, we generate the predicted values from the seed and compare the generated values with the true value to evaluate our model shown in Figure 2.



```
with tf.Session() as sess:

    saver.restore(sess, "./rnn_stock_time_series_model")

    train_seed = list(train_scaled[-12:])
    for iteration in range(12):
        X_batch = np.array(train_seed[-num_time_steps:]).reshape(1, num_time_steps, 1)
        y_pred = sess.run(outputs, feed_dict={X: X_batch})
        train_seed.append(y_pred[0, -1, 0])
```

```
INFO:tensorflow:Restoring parameters from ./rnn_stock_time_series_model
```

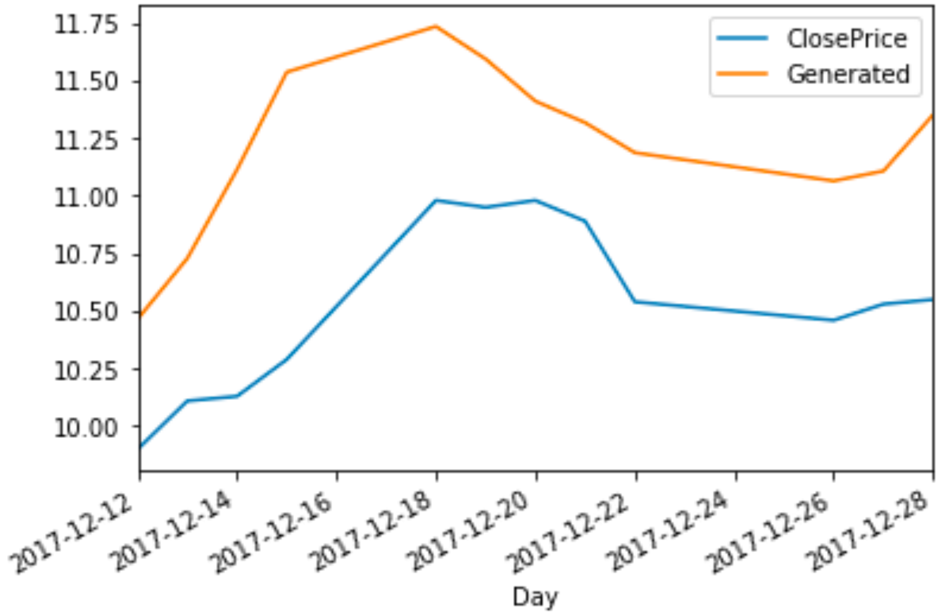


Figure 2: Comparison between true and generated stock price.

Conclusion

Here, we successfully developed a model to predict the stock price in future. The predicted stock price is very close to the true price. We only showed a stock as a sample in this experiment. The percentage of error is below 5 % based on this model. We can use the same model to predict other stock prices. In our future work, we planned to introduce the investors' sentiments on sensitive financial disclosure to incorporate the uneven price changes which is often very difficult to justify based on the historical prices only.

References

[1] King, B. F. (1966). Market and industry factors in stock price behavior. the Journal of Business, 39(1), 139-190.
[2] Chen, N. F., Roll, R., & Ross, S. A. (1986). Economic forces and the stock market. Journal of business, 383-403.

[3] Kohara, K., Ishikawa, T., Fukuhara, Y., & Nakamura, Y. (1997). Stock price prediction using prior knowledge and neural networks. Intelligent systems in accounting, finance and management, 6(1), 11-22.

[4] Kim, K. J., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. Expert systems with Applications, 19(2), 125-132.

[5] Trafalis, T. B., & Ince, H. (2000). Support vector machine for regression and applications to financial forecasting. In Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on (Vol. 6, pp. 348-353). IEEE.

[6] Lee, J. W. (2001). Stock price prediction using reinforcement learning. In Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on (Vol. 1, pp. 690-695). IEEE.

[7] Chang, P. C., & Liu, C. H. (2008). A TSK type fuzzy rule-based system for stock price prediction. Expert Systems with applications, 34(1), 135-144.

[8] Tsai, C. F., & Wang, S. P. (2009, March). Stock price forecasting by hybrid machine learning techniques. In Proceedings of the International MultiConference of Engineers and Computer Scientists (Vol. 1, No. 755, p. 60).

[9] Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. Expert Systems with Applications, 42(4), 2162-2172.

[10] Heaton, J. B., Polson, N. G., & Witte, J. H. (2016). Deep learning in finance. arXiv preprint arXiv:1602.06561.

[11] Li, X., Xie, H., Wang, R., Cai, Y., Cao, J., Wang, F., ... & Deng, X. (2016). Empirical analysis: stock market prediction via extreme learning machine. Neural Computing and Applications, 27(1), 67-78.

[12] Mullainathan, S., & Spiess, J. (2017). Machine learning: an applied econometric approach. Journal of Economic Perspectives, 31(2), 87-106.

[13] Jeon, S., Hong, B., & Chang, V. (2018). Pattern graph tracking-based stock price prediction using big data. Future Generation Computer Systems, 80, 171-187.

[14] Lee, M. S., Ahn, C. H., Kwahk, K. Y., & Ahn, H. (2018). Stock Market Prediction Using Convolutional Neural Network That Learns from a Graph. World Academy of Science, Engineering and Technology, International Journal of Business and Economics Engineering, 5(1).

[15] TORRES, D. G., & QIU, H. (2018). Applying Recurrent Neural Networks for Multivariate Time Series Forecasting of Volatile Financial Data.