Deep Learning
Amazon Stock Price

Goktug Cengiz

Roberto Fernandez

Universitat Politecnica de Catalunya

November 2019

Contents

1 Introduction

For this project we decided to focus on predicting Stock Exchange changes. This is a very interesting topic, although we know it is a very complicated one too. For starters, we just cannot predict accurately from looking only at historical stock changes.

A big factor for sudden changes are news around the world, for example:

- Some changes to a certain company can make their stocks either rise or drop. Changes can vary a lot, from a change in their leadership to a law suite.
- Some changes to a related company can impact them too. For example, if a transportation company is having any kind of problems, they can affect all other companies working with them and that may affect their stocks.
- Political issues are also known for directly affecting the stock market. An example would be Donald Trump nearly starting an economic war against China or even smaller events like here in Spain where two left-wing parties decided to form a pact and Ibex35 instantly went down.

For a more accurate prediction, we should use a service to capture recent global and local news, process them and figure out how they could affect the stock market, but this is out of scope of this project. Instead we will focus on finding the short and long term patterns that the stock market is known to have, which is the strength of neural networks. With this, we will not be able to achieve the maximum accuracy, but we can still get some predictions right.

To train and test our neural network, we will be using a dataset from Kaggle called New York Stock Exchange [**?**], which we will analyze in detail in the next section.

2 Data Preprocessing

2.1 Data Analysis

We used a large dataset with 851.264 entries, which has the following entries:

- Symbol: company identifier string
- Open: stock value at day start
- Close: stock value at day finish
- High: highest stock value during the day

- Low: lowest stock value during the day
- Volume: amount of transactions received during the day

There are 501 different stocks (symbols), each representing a different company. Most of the data spans from 2010 to 2016, although some companies were created after 2010 so they still appear but there is less data about them.

In this project, we will focus on one company, to keep things a bit simpler, but the same process could be applied for all of the companies.

Except 'Symbol' and 'Volume', all fields are really similar in the values they contain. Their minimums, maximums and means are nearly the same. But we can take a look at the differences through time.

At the start of 2010, Amazon had stock values of 136.25. It reached a minimum of 105.8 at some point in time, but keep raising until reaching a maximum of 847.21 and then descended up until the last row of data at the end of 2016 with a final value of 749.87.

From this we can observe that, while Amazon is constantly growing, they still have ups and downs that are important to predict.

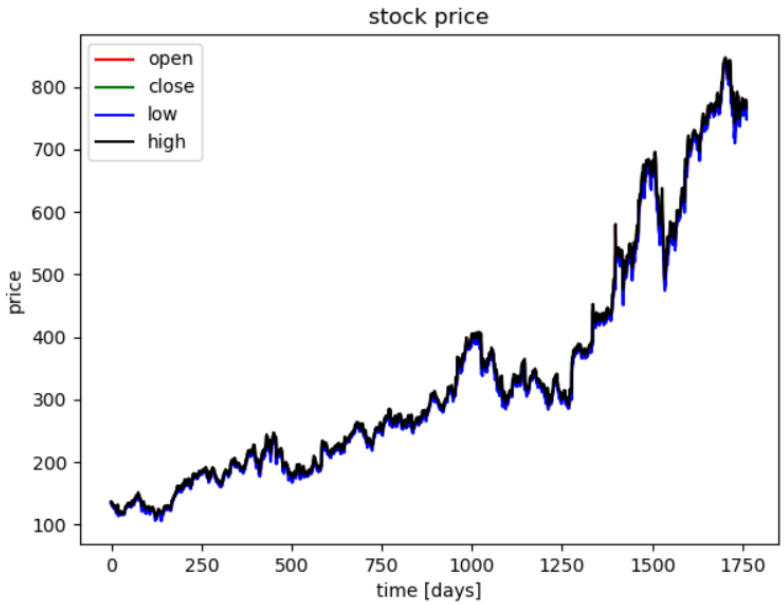

Figure 1: Amazon - Stock price change by days

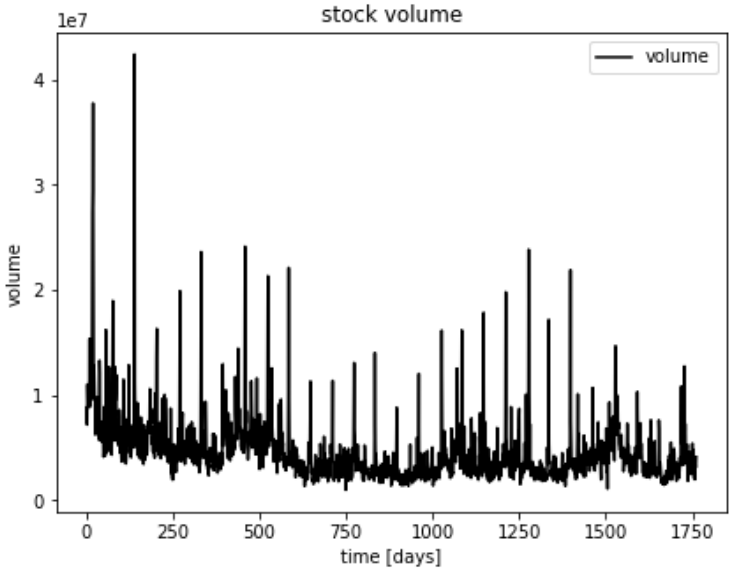

Figure 2: Amazon - Stock volume change by days

2.2 Normalization

We choose one company, Amazon is one of the most famous company in the world and analyze the movements of that company. Thus we delete the fields 'Symbol' and 'Volume', since it's not very useful.

We used normalization on fields 'Open', 'Close', 'High' and 'Low', turning them into the range of [0, 1]. Because this has the effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train deep networks.

Figure 3: After Feature Scaling

After feature scaling, we can see the values in the range of 0-1 with the graph above.

2.3 Splitting the data into train and test

We divided the data set into 2 parts using the hold out method: train and test. % 80 of our total 1762 were distributed as train(1463), % 20 test (353). **train set** is the sample of data used to fit the model. Our model will see and learn from this data. **Test set** is sample of data used to provide an unbiased evaluation of a final model fit on the training data set. We will evaluate our model thanks to test set.

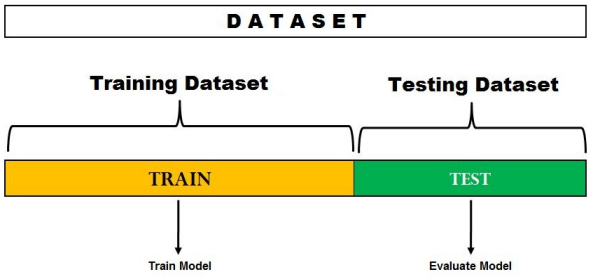

Figure 4: Data set splitting

We created a data structure with 10 timesteps and 1 output. Therefore, for each element of train set, we have 10 previous training set elements.

2.4 Reshaping

The np.reshape() function when called on an array takes one argument which is a tuple defining the new shape of the array. We cannot pass in any tuple of numbers; the reshape must evenly reorganize the data in the array. Once reshaped, we can print the new shape of the array. Finally, we have to reshape for keras library, we did all.

3 Modeling

3.1 LSTM

Long Short-Term Memory (LSTM) is a specific recurrent neural network (RNN) architecture that was designed to model temporal sequences and their long-range dependencies more accurately than conventional RNNs.

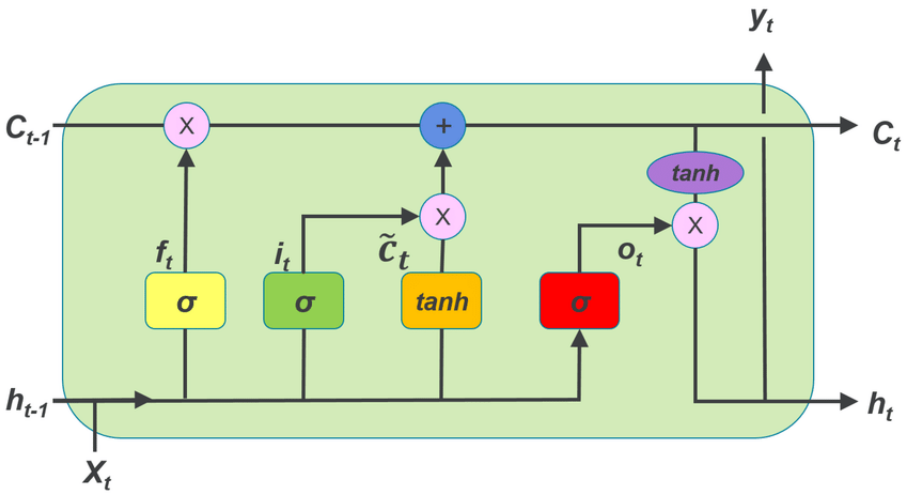

Figure 5: Long Short-Term Memory (LSTM) Structure

The LSTM contains special units called memory blocks in the recurrent hidden layer. The memory blocks contain memory cells with self-connections storing the temporal state of the network in addition to special multiplicative units called gates to control the flow of information. Each memory block in the original architecture contained an input gate and an output gate. The input gate controls the flow of input activations into the memory cell. The output gate controls the output flow of cell activations into the rest of the network. Later, theget gate was added to the memory block. This addressed a weakness of LSTM models preventing them from processing continuous input streams that are not segmented into subsequences. The forget gate scales the internal state of the cell before adding it as input to the cell through the self-recurrent connection of the cell, therefore adaptively forgetting or resetting the cell's memory. In addition, the modern LSTM architecture contains peephole connections from its internal cells to the gates in the same cell to learn precise timing of the outputs.

In our case, we did not built complex lstm structure. Firstly, we initialized lstm model. Secondly, we added a layer which has 10 units. After that, we completed model by output layer. So as to compile the model, we used Adam Optimizer, an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. Finally, we fitter our model by 200 epochs and 50 batch size.

4 Scientific and personal conclusions

After building an executing our system, we got the following chart, which shows how our training and testing predictions correlate to the real values.

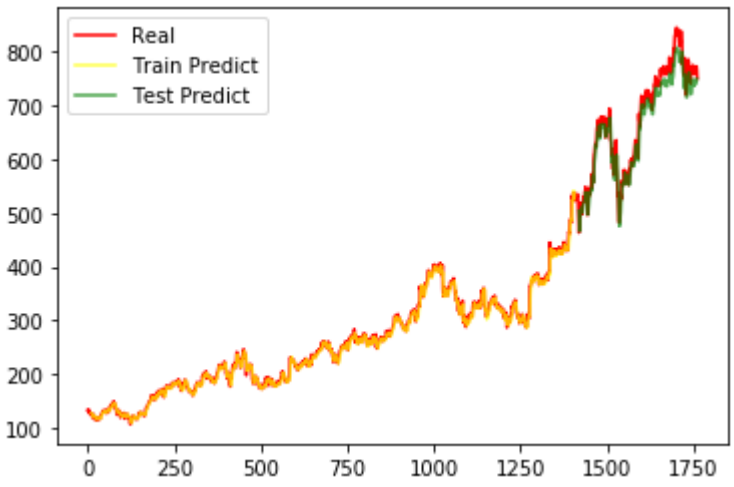

Figure 6: Amazon Stock Price Prediction

In this case we got a very accurate model, only deviating a bit at the end, going under the real price and it would start to deviate more and more given more time without the possibility to reconnect with real data and reset its error.

As we said at the start, stock price prediction is a complex topic that cannot be fully predicted based only on historical data, but using neural networks and time series we can extract some patterns that help us predict accurately during certain periods of time.

In further projects we would like to add a way to retrieve global and local news that may affect the stock market and include it inside our system, so it can determine if each news report will increase or decrease any stock in the market. We believe that having both of those ways to determine the stock price would increase the accuracy and robustness of our predictions significantly.

References

[1] https://www.kaggle.com/dgawlik/nyse

[2] Sak, Haşim / Senior, Andrew / Beaufays, Françoise (2014): "Long short-term memory recurrent neural network architectures for large scale acoustic modeling", In INTERSPEECH-2014, 338-342.

[3] Sundermeyer, Martin / Schlüter, Ralf / Ney, Hermann (2012): "LSTM neural networks for language modeling", In INTERSPEECH-2012, 194-197.
@miscdataset, title = New York Stock Exchange, howpublished = https://www.kaggle.com/dgawlik/nyse, note = Accessed: 2019-10-30,