DL Assignment 3: Recurrent Neural Networks for Stock Price Prediction

Kamalnain Taank
University of Adeliade
North Terrace Adelaide SA 5000
a1805326@student.adelaide.edu.au/

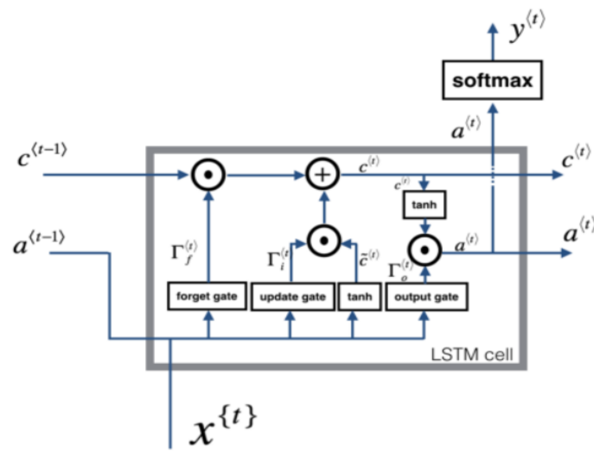
Abstract

Forecasting price of crypto-currency and stocks is a very challenging task. Interest of many professional investors have increased towards research relating to stock price forecast over the past few years. Every investor seek profit and wants a good investment and for that, they always try to figure out future situations to stock market. Prediction systems provide crucial information to people like predicted future trends of the stock market. In this assignment, we are going to a Long-Short-Term Memory(LSTM)[1] model which will tell investor whether to buy stocks or sell/don't buy them.

1. Introduction

Stock market prediction aim to predict the future trend of the stocks. There are complicated variables and methods used in real life to determine future changes in stock shares. These trend fluctuates vigorously. Factors like economic growth, company's reputation, etc make stock market volatile and predicting the future trend is very hard. However, stock market prediction models offer positive chances of profits. Advancement in technology helps investors to better analyze informative indicators and always stay at low risk. LSTM[1] is one of the most prominent architecture of Recurrent Neural Network (RNN)[2]. It has memory cells or feedback connections and it can process complete sequences of data such as speech or paragraphs. It can grasp the structure of data is very good in leaning time series patterns.

1.1. LSTM Architecture



One LSTM cell has 3 gates in it along with an activation function. These gates are nothing but just sigmoid layers which outputs numbers between 0 and 1, deciding how much of each sample to let through. 0 means nothing can be passed thorugh it and 1 means everything can pass. Forget gates decides what to forget when the new information comes. The decision about what to forget is taken by this equation

$$\Gamma_f^{\langle t\rangle} = \sigma(W_f[a^{\langle t-1\rangle}, x^{\langle t\rangle}] + b_f)$$

here W is weight, x is the input sample. If this outputs 1 then LSTM will remember it and if it outputs 0, it will forget it. After LSTM forget something, it updates itself by the equation below.

$$\Gamma_u^{\langle t\rangle} = \sigma(W_u[a^{\langle t-1\rangle}, x^{\langle t\rangle}] + b_u)$$

Output value of this equation also varies between 0 and 1. Now if the cell wants to update the new subject, it creates a new vector which is then multiplied by the previous cell state. Equation to create a new vector is below:

$$\tilde{c}^{\langle t\rangle} = \tanh(W_c[a^{\langle t-1\rangle}, x^{\langle t\rangle}] + b_c)$$

Here c is the cell state. Then the cell state is updated by this equation:

$$c^{(t)} = \Gamma_f^{(t)} * c^{(t-1)} + \Gamma_u^{(t)} * \tilde{c}^{(t)}$$

Finally at the output gate, LSTM decides the output using these two equations:

$$\Gamma_o^{(t)} = \sigma(W_o[a^{(t-1)}, x^{(t)}] + b_o)$$
$$a^{(t)} = \Gamma_o^{(t)} * \tanh(c^{(t)})$$

Among these two equations, first decides the output using sigmoid function and second equation multiply above value buy the activation function or tanh(c) function.

2. Related Work

According to Efficient market hypothesis[9],[10]: In the financial market, opportunities are exploited as soon as they arise. On the one hand, plenty sources of the information such as the stock prices, historical data and company's information make it extremely hard to predict accurately. but on the contrary as said in [11] and [12], it is possible to predict the the stock market trend. It is more practical to predict whether to but a share or don't but it or sell it instead of predicting the future stock prices. In paper by Minh Dang[13], SVM model is used in this study. This study uses time series analysis and text mining techniques to predict the future trend of the stock market. This study collects online articles from websites and extracts news about stock market trends to form a dataset. Here 3 classes were used as labels, positive, negative and neutral. They got quite high accuracy of about 73%.

In another study conducted by Jingyi Shen M. Omair Shafiq[14], they collected chineses stock market data of 2 years. They used multiple feature engineering techniques and a Long-Shirt-Term memory (LSTM) model to predict stock price trends. This study also uses principal component analysis (PCA) as a part of feature engineering.

3. Methodology

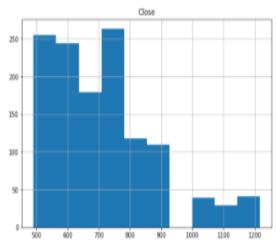
3.1. Raw Data

The dataset which we used here is Google stock price. This dataset has 1258 training samples and 20 testing samples in it. Since the dataset is a bit small, we concatenated both training and testing samples and later separated them in training and validation set for evaluation purposes.

	index	Date	Open	High	Low	Close	Volume
0	0	2012-01-03	325.25	332.83	324.97	663.59	7,380,500
1	1	2012-01-04	331.27	333.87	329.08	666.45	5,749,400
2	2	2012-01-05	329.83	330.75	326.89	657.21	6,590,300
3	3	2012-01-06	328.34	328.77	323.68	648.24	5,405,900
4	4	2012-01-09	322.04	322.29	309.46	620.76	11,688,800

3.2. Data Preprocessing

Thare are few substeps in data preprocessing. A) Date is parsed while importing the dataset using pandas B) Values were in string format so they were converted into float values along with the removal of 'commas(,)' from the strings. C) Date is converted into Unix timestamp to decrease the hassle while scaling and normalization steps. D) Upon analyzing histograms, we can see that there are some abrupt values in Close feature:



These are 1000 shares which were split into 2002 shares so to rectify that, we divided Close values upto index 560 by 2.002. E) Now we shifted close index upwards by 1 day and made a new column names 'Future' which helped us to get a target value. Meaning that 'Future' column will tell us whether prices will increase or decrease tomorrow. So if the price increases meaning 'Target' value 1, we will say that we can Buy share because it will profit us and if 'Target' is 0 then don;t buy the shares. F) Later we dropped the column so that model does not know about the future values, because in real life we don't know them and that's why we want to predict them. G) Last 10 % values are separated which is validation set, for evaluation purpose. H) Data is scaled by change in its value using percent_change() and scale() function. I) We then created sequences of past 30 days or past 30 samples. These 30 days sequences will be the basis of predicting whether to buy to buy' or not to buy the share. J) Data is balanced on the basis of 'Target' feature because unbalanced data will make our model biased and. K) Then the data is split into feature set and target set.

3.3. Model Training

4 LSTM layers were used with 50 input neurons each and 1 dense layer with 2 output neurons with softmax activation function because it gives vector which have probabilities of the classes we want to predict as our classes are mutually exclusive. For loss function we used 'Sparse Categorical Crossentropy' because our two class 1 and 0 are mutually exclusive i.e. each data sample belongs to a single class. As we are doing classification, we used accuracy metric. For optimization we used Adam Optimizer[3] as our data has noise and Adam optimizer combines the best properties of AdaGrad and RMSProp algorithms, proving an optimizing algorithm that can handle gradients on noisy data. We used a bit slow learning rate of 0.0001 because our model started

to somewhat over fit at values 0.01 and 0.001. Over fitting was observed with validation loss, although validation accuracy kept on increasing but so does validation loss. We also used decay of 10(̂-6) so that when model starts to reach minima, learning rate start decreasing so that model will take small steps towards minima which will prevent loss function from skipping the minima. Dropout of 0.2 is used used to prevent over fitting. Dropouts prevents the model from depending too much on specific neurons by which model considers all neurons instead of a any group. It makes the model more robust and model learns trend without giving special attention to specific set of neurons.

4. Experiments

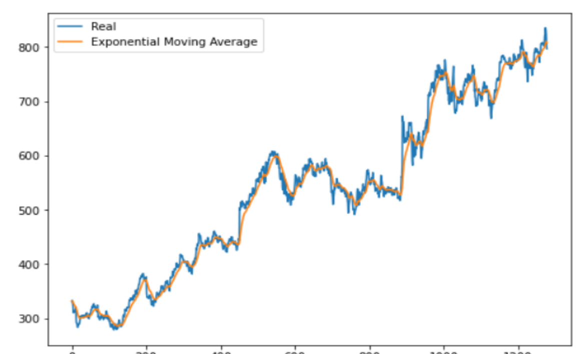
There are some things we tried to improve the accuracy of the model and to help it learn the pattern more efficiently without much loss.

4.1. Exponential Moving Average

Exponential moving average(EMA)[4] is a very popular method used to filter out noise and helps in identifying smoother trends. EMA gives more importance to the most recent sample value i.e. it give more weight to it. Since in stock market we assume that recent data is more important. It is sensitive to sudden changes in the trends.

$$EMA_t = \begin{cases} x_0 & t = 0 \\ \alpha x_t + (1-\alpha) EMA_{t-1} & t > 0 \end{cases}$$

Here xt is observation at period t. alpha(a) is the smoothing factor which is nothing but the weight assigned to the most recent data point.[5] We have given alpha = 0.1 to reduce noise and get a smoother trend of the data sample.



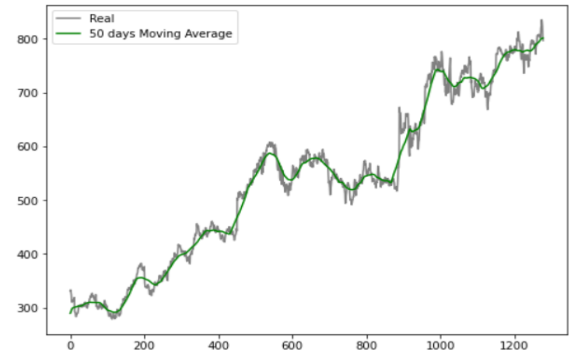
4.2. Rolling Moving Average

Rolling moving average also known as simple moving average is the unweighted mean of the last M data samples in the sample space. Value of M depends on how much smoothing we want but more smoothing result in less accuracy.

$$SMA_t = \frac{x_t + x_{t-1} + x_{t-2} + \ldots + x_{M-(t-1)}}{M}$$

Simple moving average at time period t

We took M as 50 days. We tried 4, 16 and 32 for the value of M but still had noise in it.



4.3. Commodity Channel Index

Commodity Channel Index (CCI) is a momentum-based oscillator used to help determine when an investment vehicle is reaching a condition of being overbought or oversold. It is also used to assess price trend direction and strength[6]. This gives information helps investors whether to buy a share or sell a share. The formula for CCI is given below:

The Formula For the Commodity Channel Index (CCI) Is:

$$\text{CCI} = \frac{\text{Typical Price} - \text{MA}}{.015 \times \text{Mean Deviation}}$$

where:

$$\text{Typical Price} = \textstyle\sum_{i=1}^{P}((\text{High} + \text{Low} + \text{Close}) \div 3)$$

$$P = \text{Number of periods}$$

$$\text{MA} = \text{Moving Average}$$

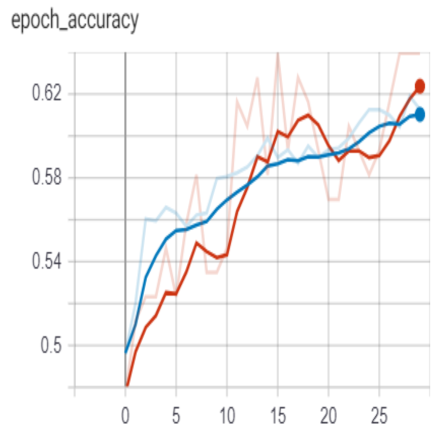
$$\text{Moving Average} = (\textstyle\sum_{i=1}^{P} \text{Typical Price}) \div P$$

$$\text{Mean Deviation} = (\textstyle\sum_{i=1}^{P} \mid \text{Typical Price} - \text{MA} \mid) \div P$$

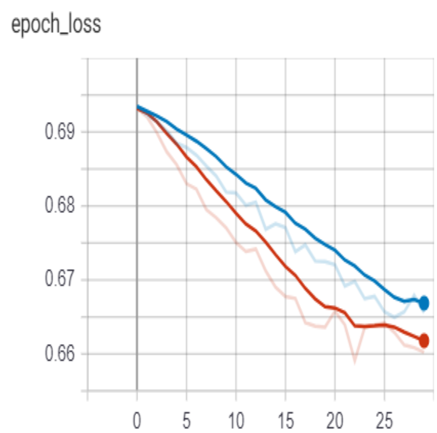
In most cases, the value of P or number of periods commonly used are 20 and so we did.

5. Results

We used tensorboard to track our model's accuracy and loss for both training samples and testing samples. We trained our model for 30 epochs with batch size of 20 because AdamOptimizer uses mini batches while updating. Accuracy curve for training and testing is given below:

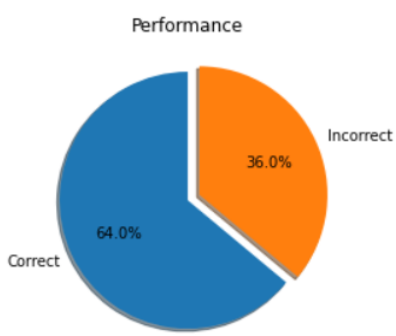


and loss curve for training and testing set is given below:



Evaluation of our model on validation set also used as testing set is:

```
Test loss: 0.6602492928504944
Test accuracy: 0.6395348906517029
```



6. Code

Code for this assignment can be found on: https://github.com/kamalkacode/Deep-Learning-Assignment.git

7. Conclusion

We explored some indices of stock market like Commodity Channel Index to help our model learn trends well. Exponential moving average and rolling moving average helped in de-noising the dataset and provided clear trends. After analyzing the accuracy and loss curve, we can say that our model performed well and could run for more epochs before over fitting. Here we can use early stopping to get the best performance out of this model. For later work, we can use Fourier Transformation on our dataset as it very good in de-noising the dataset and gives a very good overall pattern in data. We can also use Ichimoku cloud as it helps in identifying trends and provides traiding signals[7].

7.1. References

[1]. Klaus Greff; Rupesh Kumar Srivastava; Jan Koutník; Bas R. Steunebrink; Jürgen Schmidhuber (2015). "LSTM: A Search Space Odyssey". IEEE Transactions on Neural Networks and Learning Systems. 28 (10): 2222–2232.

[2]. Hochreiter, Sepp; Schmidhuber, Jürgen (1997-11-01). "Long Short-Term Memory". Neural Computation. 9 (8): 1735–1780

[3]. https://arxiv.org/abs/1412.6980

[4]. https://www.jstor.org/stable/3213263?seq=1

[5]. https://towardsdatascience.com/moving-averages-in-python-16170e20f6c

[6]. https://www.investopedia.com/terms/c/commoditychannelindex.asp

[7]. https://school.stockcharts.com/doku.php?id=technical_indicators:ichimoku_cloud [9]. Y. Gao, L. Zhou, Y. Zhang, C. Xing, Y. Sun, and X. Zhu. Sentimentclassification for stock news. In Pervasive Computing and Applications(ICPCA), 2010 5th Int Conference on, pages 99–104. IEEE, 2010. [10]. Duc Duong, Toan Nguyen and Minh Dang. Stock Market Predictionusing Financial News Articles on Ho Chi Minh Stock Exchange. In Pro-ceedings of the 10th International Conference on Ubiquitous InformationManagement and Communication, pages 71–76. ACM, 2016

[11].M. Y. Kaya and M. E. Karsligil. Stock price prediction using financialnews articles. In Information and Financial Engineering (ICIFE), 20102nd IEEE Int Conference on, pages 478–482. IEEE, 2010

[12].S. Lauren and S. D. Harlili. Stock trend prediction using simple movingaverage supported by news classification. In Advanced Informatics: Con-cept, Theory and Application (ICAICTA), 2014 International Conferenceof, pages 135–139. IEEE, 2014. [13]. https://www.researchgate.net/publication/309588518_Improvement_

methods_for_stock_market_prediction_
using_financial_news_articles [14].
https://journalofbigdata.springeropen.
com/articles/10.1186/s40537-020-00333-6