Stock Prediction with LSTM/SVR

Time-Series Financial Analysis with TensorFlow

Derrek Gass  
Engineering & Computer Science  
CSU-Sacramento  
 Sacramento, California USA  
 derrekgass@ecs.csus.edu

Alexander Lee  
Engineering & Computer Science  
CSU-Sacramento  
 Sacramento, California USA  
 alexwlee@ecs.csus.edu

ABSTRACT

There has been much development and interest in the application of machine learning to solve problems of pattern recognition in the analysis of signals such speech and images, and in recent years, has extended past this domain into many other fields. One such application is the understanding of patterns in financial time-series data. Based upon the past closing values for a stock, we would like to predict the value of the stock in the future with some degree of certainty so that traders may be informed about the state of their investment and respond with the proper course of action. Current methods rely on traditional financial indicators such as ARIMA, linear regression, moving averages, and other methods. For the purposes of risk assessment, anomaly detection, and optimization of asset portfolios, the implementation of machine learning proposes solutions to aide in this complex quantitative decision-making process. In this research project, we propose two different predictive models based upon previous research and developed with machine-learning tools; a support vector regression implementation, and a long short-term memory model. Both models performed with a RMSE of <0.05, and provide a baseline prediction of the next day value of an asset, indicating that further research may prove beneficial for solving the problem at hand.

CCS CONCEPTS

• Computing Methodologies • Applied Computing

KEYWORDS

Machine Learning, Artificial Intelligence, Finance, Computers in other domains

1 Introduction

Predicting the future would always be of value to researchers in every field. For economic analysis, even just a few percent improvement in prediction quality would allow for huge performance gains over time. Stocks are highly volatile, and patterns within difficult to recognize. Recently the field of machine learning has proven its contribution in understanding this particular problem. At the moment, applications of this technology can provide insight into the near future value of an asset, and help traders make informed decisions.

Our approach includes the deployment of two machine learning solutions that predict the next day value based upon a sliding time window. We propose a SVG model, and a LSTM model implemented in TensorFlow with Python that performs better than traditional models alone. The rest of this paper is divided into discussing the problem formation, algorithm design and implementation, experimental evaluation, and related research topics.

2 Problem Formation

We would like to take a particular stock history and predict the next day value of that asset. Then, we produce performance metrics which asses how close this value was to the actual value, and update the model by moving the time window. This network can be retrained for any asset, providing the data wrangling has been handled appropriately.

3 System/Algorithm Design

The input into our model is a sliding time window represented by an pandas 1D X array of length 36, representing the previous 36 days adjusted closing price. The adjusted price is normalized and the network trained against a y value representing the next day’s adjusted closing price. This output from our trained network is a floating-point Y value, which is inverse scaled to produce the current price in US dollars. We would like to optimize our functions to minimize the standard error as they train, producing a quality predictive indication tool.

3.1 System Architecture

We adjusted the history of values to be arrays representing segments of time. This allows us to deploy our predictions on a sliding time scale, and update the neural network weights in the ways specific to each algorithm. Both SVR and LSTM are sequential models, yet SVR is implemented with scikit-learn, and LSTM with TensorFlow. Both models are well formulated to handle regression problems, yet only one of the models is capable of producing predictions further into the future than a single day. Let’s explore each algorithm in detail.

3.2 Support Vector Regression



Figure 1: SVR Network Architecture

*3.2.1 Support Vector Regression*

Support vector regression is a way to normalize data and make predictions based upon a hyperplane that exists in parallel to an imaginary band that projects from the regression line with a width of +/- 1 standard deviation.



Figure 1: SVR Graphical Representation



Figure 1: SVR Kernel Functions

This algorithm is great in theory, as it attempts to minimize error within some preset tolerance, yet when the future happening is taken into account, the hyperplane shifts, and the current projection is updated. We can’t know how future updates will be applied until the present becomes the past, so in truth, this algorithm will only be useful for near future projections such as a 1 day or 7 day lookahead. The farther out in time, the more the uncertainty compounds and the RMSE explodes.

3.3 Long Short-Term Memory

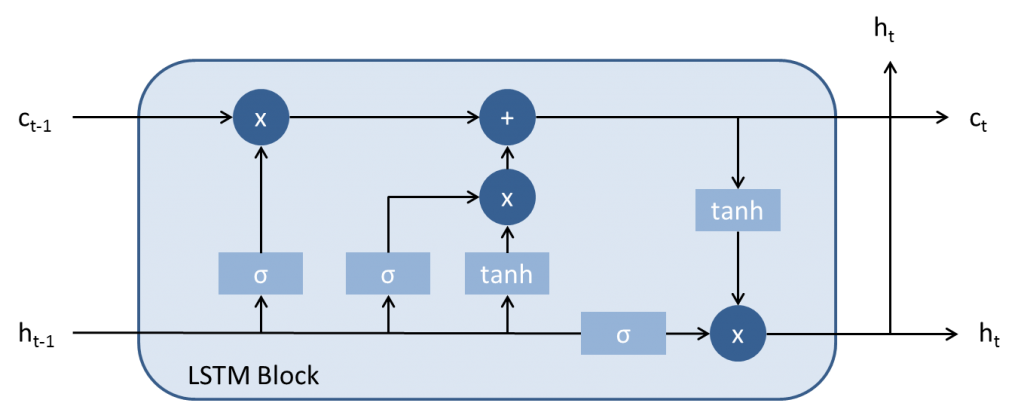


Figure 3: LSTM Overview

*3.3.1 Long Short-Term Memory*

This algorithm is great, except that it takes a moment for the effect of the current day to have an impact in the model. RMS lag.



Figure 4: LSTM Layer

4 Experimental Evaluation

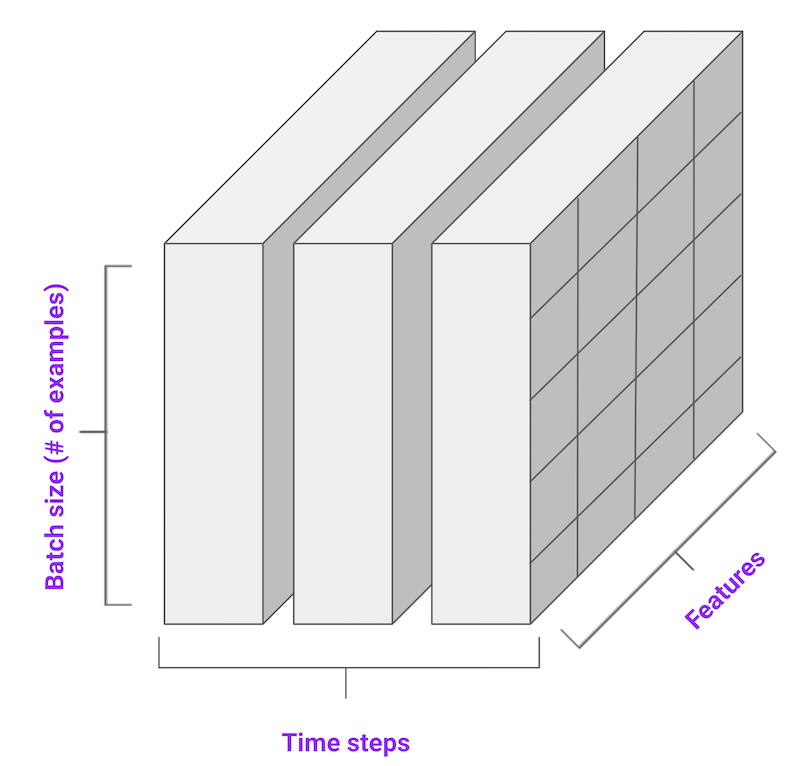


Figure 5: Encoding of Time-Series Data

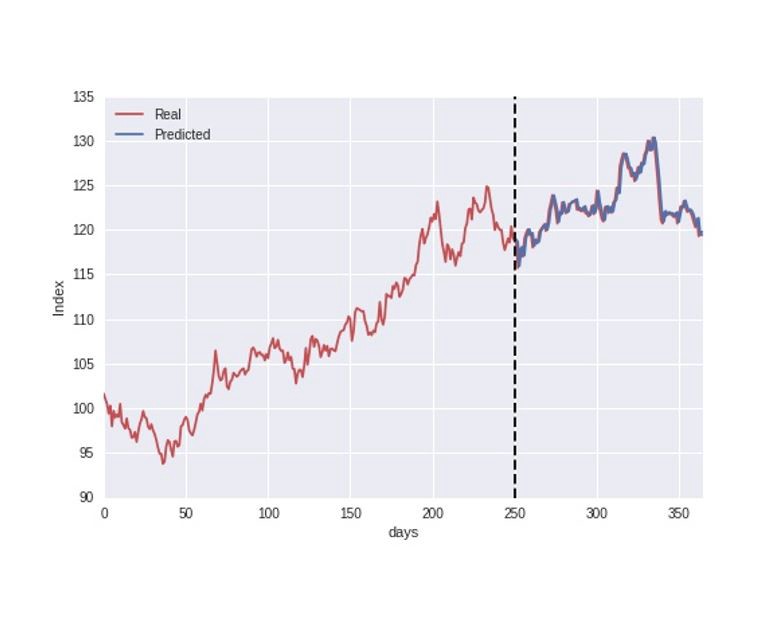
4.1 Methodology

• What was data used? How you split data to training and testing?

• What was the experimental setting?

• What metrics were used to compare different methods?

• What methods were implemented and compared? Make sure you include the competing methods that address the same problem as comparison baseline.

 Figure 4: Train/Test Split

1222 datum. Split 70 for train, 30% test.

the beginning of 2014 to the end of 2018.

X is a 36 x 1 array made up of the adj. closing prices of the last 36 days, y is the next day’s value.

Trained on 855 samples, validated on 367 samples.

20 epochs, batch size 10

Loss: mean squared error

Optimizer: adam

Dropout layers set to 0.2

One dense output layer predicting the next day’s asset value.

4.2 Results

Present the quantitative results of your experiments. Figures such as charts or histograms are frequently better than tables. For each figure, explain the result. What conclude we can draw from each figure?

5 Related Work

Answer the following questions for each related work that addresses the same or a similar problem.

• What is their problem and method?

• How is your problem and method different?

• Why is your problem and method better?

6 Conclusion

Briefly summarize the results and conclusions.

7 Work Division

A paragraph stating how the work is divided over all team members in your project.

8 Learning Experience

One or two paragraphs stating what you (and your partners) have learn from this project.

ACKNOWLEDGMENTS

Special thanks to Dr. Haiquan Chen. Without your guidance through the machine learning class, this endeavor would remain elusively out of reach to undergraduate students.

REFERENCES

[1] FNM Surname (2018). Article Title. Journal Title, 10(3), 1–10.

[2] F.N.M Surname, Article Title, https://www.acm.org/publications/proceedings-template.

[3] F.N.M Surname and F.N.M Surname, 2018 Article Title, The title of book two (2nd. ed.). Publisher Name, City, State, Country.

[4] Ian Editor (Ed.). 2018. *The title of book two* (2nd. ed.). University of XXX Press, City, Chapter 100. DOI: <http://dx.doi.org/10.1000/0-000-00000-0>.

Conference Name:ACM Woodstock conference

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00