Moving Average Prediction of Financial Time Series Data based on a Two-Dimensional Convolutional Neural Network

Junyoung Sim  
*Korea International School Jeju*Republic of Korea  
jysim24@kis.ac

*Abstract*—This paper proposes a model predicting the long-term moving average of financial time series by using a convolutional neural network. Specifically, the proposed model having a fully connected deep neural network with stochastic gradient descent is trained to predict future moving averages based on the feature maps of past moving averages. The proposed model is able to obtain feature maps of past moving averages by encoding past moving averages with two-dimensional convolutions and two-dimensional max pooling for enhanced feature extraction of both sequential and periodic relationships within financial time series. Implementing the proposed model on the S&P 500, Dow Jones Industrial Average, and NASDAQ Composite resulted in predicting the moving average of 709 days (84.71%), 663 days (79.21%), and 807 days (96.42%), respectively, out of 837 backtesting days.

Keywords—moving average, market forecasting, feature extraction, neural network, deep learning

# Introduction

Despite the financial market being a noisy environment, its behavior has been predicted by numerous models. Diffusion indices [[4](#diffusionindex)], neural networks [[5](#neuralnet)], long-short-term memory models [[2](#lstm)], and support vector machines [[3](#svm)] are merely a few examples that have a variety of implementations. However, data processing of these methods compromises their efficacy and practical application.

A major technical issue with previous methods is financial time series being processed by one-dimensional encoding. Although intuitive, one-dimensional encoding of financial time series only reduces the noise and size of sequential data points without extracting meaningful features. Given that pattern recognition tasks require feature extraction for cutting-edge learning performance and prediction accuracy as suggested by prior research [[1](#featureextraction)], feature extraction with a higher dimensionality must be used to encode financial time series. Therefore, the proposed model having two-dimensional convolutions and two-dimensional max pooling encode financial time series into compact feature maps containing both sequential and periodic relationships that are learned by a deep neural network for enhanced long-term predictions.

Another issue with previous methods is their objective of predicting the return of financial assets. This is not the most robust investment strategy since buy-and-hold return predictions do not indicate uncertainty in market fluctuations one may take advantage of by optimizing long-term portfolios or alternating between long and short positions. To supplement this weakness, the proposed model aims to interpret and predict moving averages of financial time series because moving averages are effective in indicating long-term trends reflecting market fluctuations.

Thus, (with moving averages and two-dimensional encoding, this study expects models to make accurate predictions (gains, drop downs) on future moving averages without leaving certain time periods with uncertainty; will be beneficial for long-term portfolio management) \*\*\*

# Methodology

Figure 1 shows the architecture of the proposed model, which is designed to predict the moving average of the next 75 days based on the moving average of the past 121 days. Each process involved in the model, such as time series sampling, two-dimensional encoding, and training the deep neural network, will be provided a detailed explanation throughout this section.

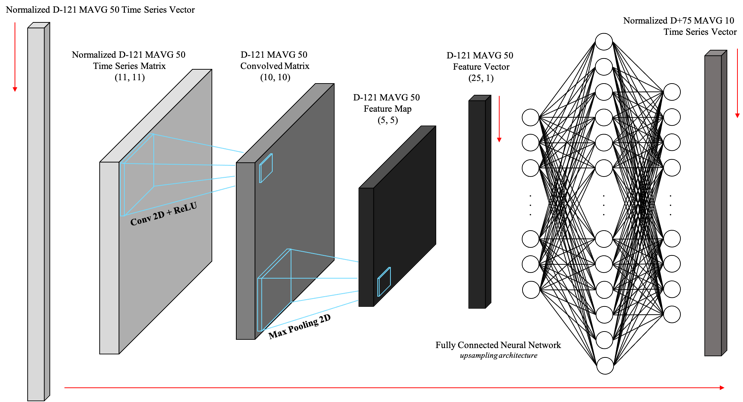


Fig. 1. Model Architecture for Predicting Moving Average of Financial Time Series

## Moving Average Time Series Sampling

Equation 1 was used for computing moving averages in which *t* represents the raw price data and *n* represents the number of periods being observed at a time.

Moving averages are normalized into a float value between 0 and 1 based on its minimum and maximum value as written in equation 2.

The historical data of the S&P 500 (SPY), Dow Jones Industrial Average (DIA), and NASDAQ Composite (QQQ) are used to create models on each of them. Once the historical data is downloaded via [Alpha Vantage](https://www.alphavantage.co/), a time frame observing 205 data points, *t*, was slid through. Within *t*, one input and its corresponding output is sampled; the normalized moving average of the first 170 data points in *t* withreturns the input, whereas the normalized moving average of the last 84 data points in *t* withreturns the output. 5,080 samples from each index between January 1, 2000 and January 1, 2021 were acquired by applying this method of sampling. The sampled time series dataset is preserved in chronological order for backtesting purposes.

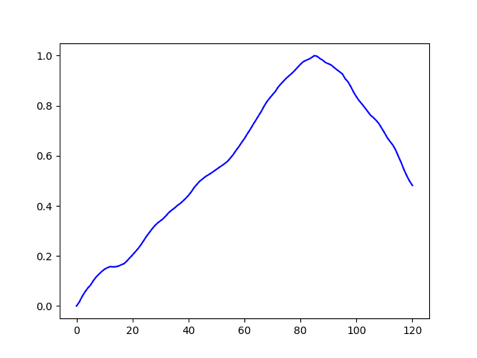


Fig. 2. Normalized D-121 MAVG 50 Time Series Input

Note that when sampling the moving average input to convey the long-term trend of the past and omit unnecessary noise as shown in Figure 2. On the other hand, when sampling the moving average output to allow some noise that convey information about minor corrections and major drop downs as shown in Figure 3.

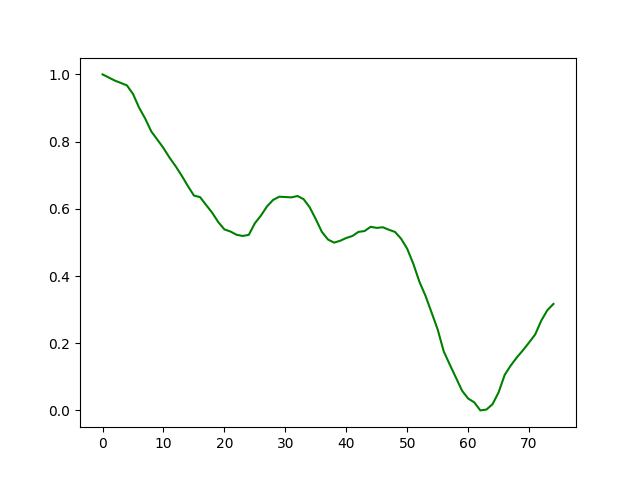


Fig. 3. Normalized D+75 MAVG 10 Time Series Output

## Feature Extraction of Moving Average Time Series

After sampling the dataset, each input, which is originally a vector with 121 data points, is reshaped into a 11 x 11 matrix. Subsequently, one layer of two-dimensional convolutions and max pooling are used as written in equation 3 and 4. Specifically, convolutions with a 2 x 2 kernel and stride value of 1 and max pooling with a 2 x 2 window encode each reshaped input. This process of two-dimensional encoding would be effective in extracting features representing both sequential and periodic relationships between past and future data points of financial time series that is reshaped into a matrix.

As a result, each input is condensed into a feature map consisting of 25 data points that can be visualized into a grayscale image as shown in Figure 4. Note that the feature map of each input is flattened into a vector before training the deep neural network.

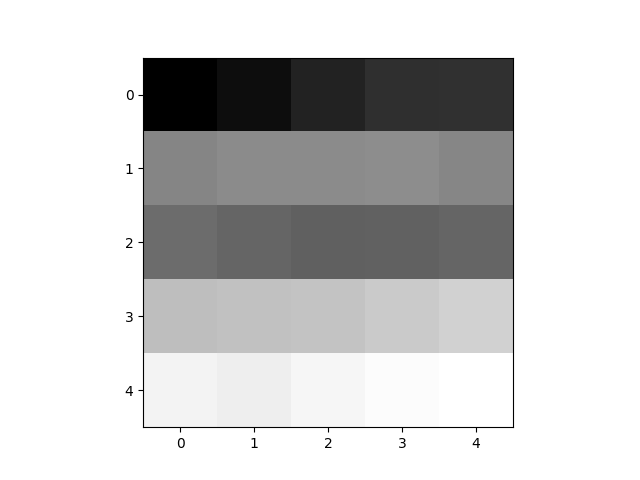


Fig. 4. Feature Map of Encoded Time Series Input

## Deep Neural Network

Since each encoded input and its corresponding output has 25 data points and 75 data points, respectively, a deep neural network with fully connected layers in an upsampling architecture is implemented as shown in Figure 5.

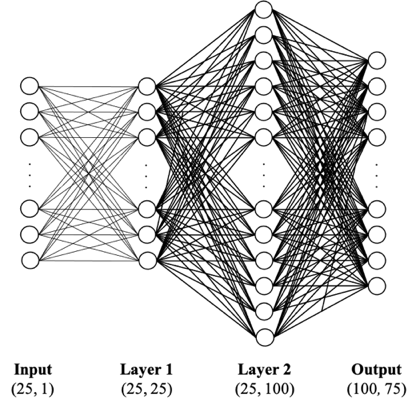


Fig. 5. Deep Neural Network Architecture

Equation 5 shows the activation of each node in the neural network, which is the ReLU dot product of and ; is a placeholder for either the input or the previous layer’s activation and is the node’s weight.

Equation 6 shows the model’s cost function, which is the mean squared error (MSE) between the supervised output and the prediction. Stochastic gradient descent was used to train the deep neural network, which updates weights according to their gradients and learning rate as written in equation 7.

## Implementation

To model on SPY, DIA, and QQQ, the first 85% of their dataset (4318 samples) was used for training with an epoch of 10,000 and a learning rate of 0.01. The remaining 15% (762 samples) were used for backtesting. Since each prediction forecasts the moving average of 75 days, there are a total of 838 backtesting days.

The total number of predicted days out of 838 backtesting days were calculated to quantify the performance of each model. In order to so, predictions with a MSE lower than 0.06 are selected to identify a set of days the models were able to predict if they have a MSE lower than the training MSE during the first 10 days. These predictions will be referred as correct predictions throughout the remaining sections of this paper.

# Results

The following table shows the training and backtesting MSE of the models on SPY, DIA, and QQQ.

1. Training and backtesting MSE

| Symbol | MSE | |
| --- | --- | --- |
| Training | Backtesting |
| SPY | 0.0267 | 0.1419 |
| DIA | 0.0312 | 0.1447 |
| QQQ | 0.0266 | 0.1360 |

Figures 6 through 8 show several correct predictions. Note that the green represents the supervised output, whereas the red represents the prediction.

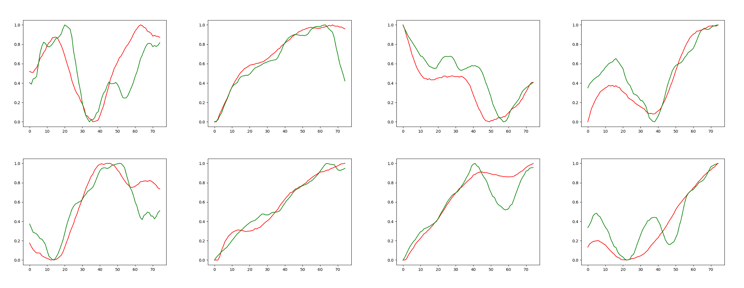


Fig. 6. Correct Backtesting Predictions (Red) on SPY vs. Actual (Green)

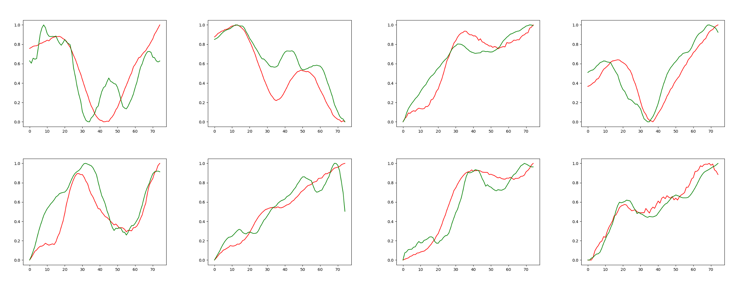


Fig. 7. Correct Backtesting Predictions (Red) on DIA vs. Actual (Green)

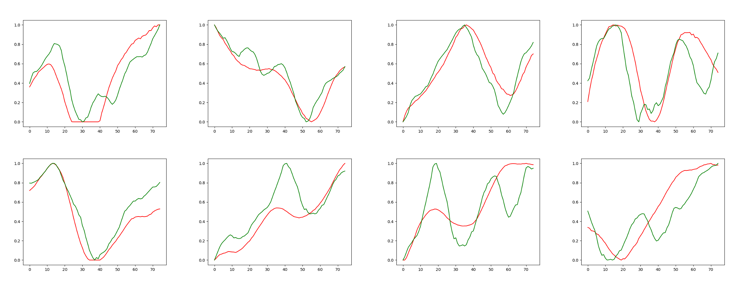


Fig. 8. Correct Backtesting Predictions (Red) on QQQ vs. Actual (Green)

Table 2 shows the number of correct predictions and the percentage of the days that were predicted by each model to evaluate their performance.

1. performance evaluation

| Symbol | Correct Predictions | Predicted Days (%) |
| --- | --- | --- |
| SPY | 50 | 84.71 |
| DIA | 61 | 79.21 |
| QQQ | 56 | 96.42 |

# Discussion

# Conclusion

# Insights

## Hyperparameters

Although the hyperparameters used in this study were acquired through rigorous testing, additional testing with different time series size, kernel dimensions, and neural network architectures must be conducted to implement different market forecasting perspectives.

## Wider Applications

This paper only introduced models on SPY, DIA, and QQQ. Testing on other economic indicators and stocks would be necessary to validate the efficacy of the proposed model in a wider spectrum. Moreover, further research would be able answer whether the proposed model can be applied to fields of study other than finance requiring time series forecasting.

##### References

1. S. Khalid, T. Khalil, and S. Nasreen, “A survey of feature selection and feature extraction techniques in machine learning,” in *2014 Science and Information Conference*, 2014, pp. 372–378. doi: [10.1109/SAI.2014.6918213](https://doi.org/10.1109/SAI.2014.6918213).
2. T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018, doi: <https://doi.org/10.1016/j.ejor.2017.11.054>.
3. K. Kim, “Financial time series forecasting using support vector machines,” *Neurocomputing*, vol. 55, no. 1, pp. 307–319, 2003, doi: <https://doi.org/10.1016/S0925-2312(03)00372-2>.
4. J. H. Stock and M. W. Watson, “Macroeconomic Forecasting Using Diffusion Indexes,” *Journal of Business & Economic Statistics*, vol. 20, no. 2, pp. 147–162, 2002, doi: [10.1198/073500102317351921](https://doi.org/10.1198/073500102317351921).
5. T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka, “Stock market prediction system with modular neural networks,” in *1990 IJCNN International Joint Conference on Neural Networks*, 1990, pp. 1–6 vol.1. doi: [10.1109/IJCNN.1990.137535](https://doi.org/10.1109/IJCNN.1990.137535).