A Long-term Recurrent Convolutional Network for Stock Index Prediction

Xinpeng Yu1, and Dagang Li2
1School of Electronics and Computer Engineering, Peking University, Shenzhen, China
2International Institute of Next Generation Internet, Macau University of Science and Technology, Macau, China

*Email: yuxinpeng@pku.edu.cn

Abstract: Stock index prediction aims to predict the future price of stock indexes, which plays a key role in seeking the maximum profit from stock investment. However, it has been proven to be a very difficult task because of its innate complexity, dynamics, and uncertainty. With the rapid development of deep learning, more researchers have attempted to apply nonlinear learning methods such as long short-term memory networks (LSTMs) to capture the complex patterns hidden in market trends. In this paper, we propose a Long-term Recurrent Convolutional Network (LRCN), which combines convolutional layers and long-range temporal recursion and is end-to-end trainable. In the LRCN model, the two-dimensional convolutional neural network (2D-CNN) performs convolution on the most recent region to capture local fluctuation features, and the long short-term memory (LSTM) learns the long-term temporal dependencies to improve stock index prediction. To evaluate the effectiveness of LRCN, we collected real stock market data for stock indexes S&P 500 and DJIA, and the experimental results show that the proposed LRCN can significantly outperform several existing highly competitive methods.

Keywords: stock index prediction; long short-term memory; convolutional neural network; recurrent convolutional network; deep learning

1. Introduction
The stock price is unfixed and nonlinear time sequence, which makes it very difficult to forecast the future trend, the stock price fluctuations are often influenced by all kinds of information, for example, government policy, corporate performance, even the breaking news, thus, stock trend prediction becomes a challenging task. Most of the traditional efforts on stock prediction rely on time-series analysis models, such as autoregressive models [1]. Besides, the hidden Markov model (HMM) has also been used to make a nonlinear prediction of stock trend [2]. In general, these solutions create dynamic stock indicators based on stock prices and volumes as stochastic inputs and take the historical data of indicators to fit the stochastic trends. However, such traditional solutions yield apparent drawbacks as they lack the capability to model dynamic validity of indicators, highly volatile market, and the complex correlation between stocks and the market.

In recent years, more and more people try applying machine learning techniques, especially deep learning, to pursue a more promising stock prediction. Recurrent neural networks (RNN) have been introduced as a promising substitute since its ability to model the sequential nature and nonlinear structure within the stock trend prediction task [3]-[5]. Although the traditional RNN has the ability to
process nonlinear data, but not enough to model the long-term dependence on the time series. This motivates the use of the gated memory cells, thus the famous Long Short Term Memory (LSTM) network was proposed to memorize the context of the time series data for a long time. LSTM was introduced by [6], and it aimed for a better performance by tackling the vanishing gradient issue that recurrent networks would suffer when dealing with long data sequences. Additional gating units in the LSTM make it capable of maintaining the long-term memory of the trading patterns from the historical prices. Recently, several studies have introduced convolutional neural networks (CNNs) into the stock performance prediction domain [7]-[9], inspired by their remarkable achievements in other fields. A CNN is capable of directly extracting features of the input without sophisticated preprocessing and can efficiently process various complex data [10,11]. Chen et al. [8] used a 1D-CNN with an agent-based reinforcement learning algorithm to study Taiwan stock index futures. Hao et al. [9] integrated both 1D-CNN and LSTM to extract multiple time scale features for a more comprehensive learning of price sequences.

In the abovementioned stock price prediction models, all the convolutional neural networks are one-dimensional. In this study, we use a 2D-CNN to perform convolution on the recent region of historical time series for the purpose of capturing more local fluctuation features, which can potentially enhance these CNN-based approaches, and we use the LSTM to learn the long-term temporal dependencies to facilitate better prediction. Combining these two gives a Long-term Recurrent Convolutional Network (LRCN) with markedly better stock index prediction performance.

2. Long-term Recurrent Convolutional Network

2.1. Notation and Problem Statement

The goal of this work is to predict the closing price of the next day. Given the time series of all features denoted as \( X = (X_1, X_2, ..., X_T) \in \mathbb{R}^{T \times N} \) where \( T \) represents time window size and \( N \) specifies the number of features. Hence, \( X_t = (x_1^t, x_2^t, ..., x_N^t) \in \mathbb{R}^N \) is a vector of all the \( N \) features at time \( t \). The LRCN model aims to learn a nonlinear mapping function \( F(\cdot) \) as follows:

\[
\hat{y}_{t+1} = F(X_1, X_2, ..., X_T)
\]

The features used in this paper include open, close, high, low, adj_close, and volume in the granularity of the trading day. This study uses adj_close of the next day as the target \( y \).

2.2. Proposed Model

The overall structure of our proposed LRCN model is shown in figure 1. The forecast of stock performance is affected by both long-term temporal dependencies and local fluctuation features. Due to its memory blocks, the LSTM network has a strong capability of capturing the long-term memory of sequential data with high prediction capacity on chaotic time series. Hence, we adopt LSTM to learn long-term temporal dependencies from stock data time series. In addition, a two-dimensional convolution (Conv2D) is introduced to extract local fluctuation features, which can help to enhance the prediction performance. This particular neural network learns filters to study the mapping relationship between any input and output from the training on known patterns.

Given a time series of historical data for a stock index \( X = (X_1, X_2, ..., X_T) \top \), for each input time step \( X_t \), the proposed model performs convolution on the recent region from time \( t-M \) to \( t-I \) \((x_{t-M}, ..., x_{t-I-3}, x_{t-I-2}, x_{t-I-1})\) to capture local fluctuation features. In order to fit the input of the Conv2D, we transform the input sequences into a matrix, where the rows represent features and the columns represent time. For easier description, it is defined as Conv. Then, the output of Conv2D layer Conv, and raw input \( X_t \) are concatenated to form a vector, and such a joint feature is fed to the LSTM layer in order to learn long-term temporal dependencies. Formally, the LSTM can be formulated as follows, suppose the hidden state at the previous time step \( t-I \) is \( h_{t-I} \):

\[
\begin{align*}
\tilde{i}_t &= \text{sigmoid} (W_i \cdot [\text{Conv}_t, X_t, h_{t-I}] + b_i) \\
\tilde{c}_t &= \text{tanh} (W_c \cdot [\text{Conv}_t, X_t, h_{t-I}] + b_c)
\end{align*}
\]

2
\[ f_t = \text{sigmoid} \left( W_f \cdot \left[ \text{Conv}_t, X_t, h_{t-1} \right] + b_f \right) \]  
\[ c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \]  
\[ o_t = \text{sigmoid} \left( W_o \cdot \left[ \text{Conv}_t, X_t, h_{t-1} \right] + b_o \right) \]  
\[ h_t = o_t \odot \tanh \left( c_t \right) \]

where \( \odot \) denotes the elementwise product, the three types of gating units use the \( \text{sigmoid}( \cdot ) \) as the activation function, and the hyperbolic tangent \( \tanh( \cdot ) \) is adopted as the activation function for input modulation and output. The input gate \( i_t \) determines the allowed number of candidate hidden values \( \tilde{c}_t \) updated into the memory cell. The forget gate \( f_t \), controls how much previous information should be kept in the new cell. The output gate \( o_t \) defines the proportion of information that can be output. Then, the output of the LSTM layer \( h_t \) is fed to the fully connected layer for prediction, the overall output of the LRCN model is expressed as:

\[ \hat{y}_{t+1} = W_2 h_t + b_2 \]  

where \( W_2 \) is a weight matrix and \( b_2 \) is the bias vector, \( \hat{y}_{t+1} \) is the prediction value of the next day.

![Figure 1](image_url)

**Figure 1.** Structure of the proposed LRCN model for stock index prediction, where detailed layer connections are indicated.

### 3. Experiments

#### 3.1. Datasets and Setup

We collected real-world historical data of two stock indices: S&P 500 and DJIA, which traded from Jan 3, 2000, to Dec 30, 2020 at a daily frequency, for a total of 21 years. For a fixed time window of size \( T \) and a stride of 1, each sample incorporated input sequences of \( T-1 \) time steps and a target index value for model training and evaluation. Then, we divide each dataset into 7:1.5:1.5 ratios in the time dimension as the training set, validation set, and test set.

#### 3.2. Parameter Settings and Evaluation Metrics

In the experiments of the previous study [12] for DA-RNN, the length of time window 10 yielded the best results. In this study, all the compared models use the same length of time window as 10. The length
of the recent region is set as 6, which is the same as the number of input features in order to facilitate calculation. In addition, the learning rate is 0.001, the batch size is 128, the loss function is MSE, the neuron number in LSTM is 64, and we train all the models for 1000 epochs. To measure the effectiveness of various methods for stock index prediction, we consider three evaluation metrics: root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination ($R^2$).

### 3.3. Results

To evaluate the effectiveness of the proposed LRCN, we conduct experiments to compare our results with those of the compared models, including a standard long short-term memory neural network (LSTM), the encoder-decoder network (Encoder-Decoder) proposed in [13], we change it to perform stock index prediction as Qin et.al. did in [12], and the dual-stage attention-based recurrent neural network (DA-RNN) proposed in [12]. The comparison results of all the models over the two datasets are shown in table 1.

**Table 1. Stock index prediction results over the S&P 500 Dataset and DJIA Dataset**

| Models       | S&P 500 Dataset | DJIA Dataset |
|--------------|-----------------|--------------|
|              | MAE ($\times 10^{-2}\%$) | RMSE ($\times 10^{-2}\%$) | $R^2$ ($\times 10^{-1}\%$) | MAE ($\times 10^{-2}\%$) | RMSE ($\times 10^{-2}\%$) | $R^2$ ($\times 10^{-1}\%$) |
| LSTM         | 0.96            | 1.41         | 9.77         | 1.81          | 2.19          | 9.18          |
| Encoder-Decoder | 1.28          | 1.75         | 9.65         | 2.92          | 3.34          | 8.09          |
| DA-RNN       | 1.02            | 1.47         | 9.75         | 2.57          | 3.22          | 8.22          |
| LRCN         | **0.92**        | **1.41**     | **9.78**     | **1.09**      | **1.64**      | **9.54**      |

As illustrated in table 1, our proposed LRCN model shows better performance than LSTM. This suggests that using the 2D-CNN to perform convolution on the recent region of historical time series can enhance the prediction performance. With the integration of the 2D-CNN as well as the LSTM, our
LRCN achieves the best MAE, RMSE, and $R^2$, that increase of 9.80%, 4.08%, and 0.31% and 57.59%, 49.07%, and 16.06% for the S&P 500 and DJIA datasets, respectively, compared to the DA-RNN model, indicating the effectiveness of our overall model structure. For visual comparison, we show the prediction results of Encoder-Decoder, DA-RNN, and LRCN over the DJIA dataset in figure 2. We observe that LRCN generally fits the ground truth much better than Encoder-Decoder and DA-RNN, indicating the effectiveness of our overall model structure.

4. Conclusion
In this paper, we propose a Long-term Recurrent Convolutional Network (LRCN) for stock index prediction. Specifically, we use a two-dimensional convolutional neural network (2D-CNN) to perform convolution on the most recent region to capture local fluctuation features. Then, we use the long short-term memory (LSTM) to learn the long-term temporal dependencies to improve stock index prediction. Therefore, the LRCN can take advantage of the powerful ability of 2D-CNN to extract features and thus provide more accurate predictions. Extensive experiments on the S&P 500 dataset and the DJIA dataset demonstrated the superior performance of the proposed LRCN relative to both Encoder-Decoder and DA-RNN, indicating the LRCN model has broad application prospects and is highly competitive. In summary, this work provides new insight into stock index prediction research and can help to develop better predicting models.

5. References
[1] L. Li, S. Leng, J. Yang, and M. Yu, “Stock market autoregressive dynamics: a multinational comparative study with quantile regression,” Mathematical Problems in Engineering, vol. 2016, 2016.
[2] M. Zhang, X. Jiang, Z. Fang, Y. Zeng, and K. Xu, “High-order hidden markov model for trend prediction in financial time series,” Physica A: Statistical Mechanics and its Applications, vol. 517, pp. 1–12, 2019.
[3] R. Akita, A. Yoshihara, T. Matsubara, and K. Uehara, “Deep learning for stock prediction using numerical and textual information,” in 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS). IEEE, 2016, pp. 1-6.
[4] Q. Gao, “Stock market forecasting using recurrent neural network,” Ph.D. dissertation, University of Missouri–Columbia, 2016.
[5] A. M. Rather, A. Agarwal, and V. Sastry, “Recurrent neural network and a hybrid model for prediction of stock returns,” Expert Systems with Applications, vol. 42, no. 6, pp. 3234–3241, 2015.
[6] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[7] Zhan, X.; Li, Y.; Li, R.; Gu, X.; Habimana, O.; Wang, H. Stock Price Prediction Using Time Convolution Long Short-Term Memory Network. International Conference on Knowledge Science, Engineering and Management. Springer, 2018, pp. 461–468.
[8] Chen, C.T.; Chen, A.P.; Huang, S.H. Cloning strategies from trading records using agent-based reinforcement learning algorithm. 2018 IEEE International Conference on Agents (ICA). IEEE, 2018, pp. 34–37.
[9] Hao, Y.; Gao, Q. Predicting the Trend of Stock Market Index Using the Hybrid Neural Network Based on Multiple Time Scale Feature Learning. Applied Sciences 2020, 10, 3961.
[10] Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. Communications of the ACM 2017, 60, 84–90.
[11] Widiastuti, N. Convolution Neural Network for Text Mining and Natural Language Processing. IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2019, Vol. 662, p. 052010.
[12] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, “A dualstage attention-based recurrent neural network for time series prediction,” arXiv preprint arXiv:1704.02971, 2017.
[13] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” arXiv preprint arXiv:1406.1078, 2014.