WEP Warwick Evans Publishing

Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 5
AIDML 2024

Deep Learning for Temporal Stock Prediction: A Comparison

Bohan Xuan *

Trinity College, University of Toronto, Toronto, Canada

* Corresponding Author: johan.xuan@mail.utoronto.ca

Abstract. Deep learning has emerging and numerous applications for most areas, including finance, physics and medical science, etc. The deep-based models achieve satisfying performance on those tasks. In this paper, we aim to provide a summary of deep learning-based temporal stock prediction. Specifically, we first categorize the models into three aspects, including CNN-based models, RNN-based models, and hybrid models, by combining CNN and RNN. Then, we detail the preliminary knowledge for deep-based models, including the components of CNN and RNN. Furthermore, we provide an in-depth review of those methods. Finally, we provide a perspective discussion on the stock prediction tasks for further research. We hope our method can be useful for future research and provide a brief introduction to beginners.

Keywords: Deep Learning; Temporal Stock Prediction; Convolutional Neural Network; Recurrent Neural Network.

1. Introduction

The paper will provide a state-of-the-art snapshot of the development of stock prediction using the CNN network, RNN network, and combined CNN and RNN network. The stock prediction problem is as follows: Given some information about the stock market, construct a potential model to predict its future trend. The provided information includes but is not limited to, open price, closing price, highest price, lowest price, and trading volume. The model also needs to predict these values in the future stock market. Such a model helps investors prepare for future risk management, analyze the company's operation status, and, most importantly, give additional suggestions for future investment. Stock prediction is vital in data analysis and time series data prediction, and interest is still strong. Numerous overviews exist for this field. Scholars are combining deep learning and machine learning models with the financial industry and making viable solutions for financial problems based on the models developed[1-4]. The financial industry has applications of genetic algorithms, rule-based systems, evolutionary Algorithms, and artificial neural networks. Deep learning has enormous potential in the financial industry. There might be applications of NLP, semantics and text mining-based models, and hybrid models based on spatio-temporal data representations.

In this paper, we aim to summarize the existing deep-based methods for temporal stock prediction. We first provide the basic concept of deep-based methods, including CNN and RNN. Then, we describe each paper briefly according to its category. The following sections can be arranged as follows. In section 2, we first describe the basic knowledge of Convolutional Neural Networks and Recurrent Neural Networks. In section 3, we introduce those methods sequentially and provide an in-depth analysis of this task.

2. Preliminary

2.1. Convolutional Neural Network

As shown in Figure 1, the structure of CNN has several layers, which can be divided into four different layers: convolution, ReLU layer, pooling layer, and fully connected layer.

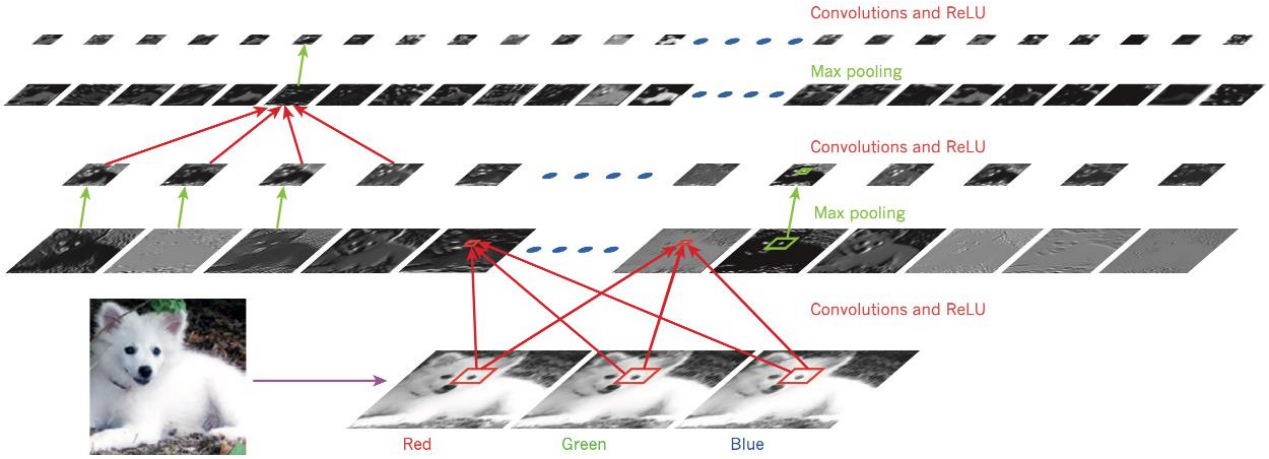

Figure 1. The framework of CNN models. The figure is from [6].

Convolutional and activation layers. Considering the input of the CNN network as an $\mathrm{N} \times N$ matrix and a $\mathrm{V} \times V$ filter matrix, the convolution layer will apply these two matrices to produce a new matrix with the size of $(\mathrm{N}+\mathrm{V}-1 \times N+\mathrm{V}-1)$, whose entry is given by $k_{i,j}=\sum_{p=0}^{V-1}\sum_{l=0}^{V-1} v_{p,l} \cdot n_{i+p,j+l}$, where $v_{p,l}$ is the entry of the $\mathrm{V} \times V$ matrix at row p and column l, and $n_{i+p,j+l}$ is the entry of the $\mathrm{N} \times N$ matrix at row $\mathrm{i}+\mathrm{p}$ and column $\mathrm{j}+\mathrm{l}$. The filter matrix in this layer will extract a particular "feature" from the input matrix by setting the filter matrix's higher weight to some specific entries. There might also be multiple filter matrices in a CNN model whose output will be possessed layer by layer. The matrix M will be the input of the ReLU layer. In this layer, the activation function $\mathrm{f(x)}=\max(0,\mathrm{x})$ will be applied entry-wise to the M. The primary purpose of using the function is to introduce non-linearity. The result from the Relu layer will be fed to the pooling layer.

Pooling layer. The pooling layer extracts some information from the matrix given by the ReLU layer. The pooling layer can have different sizes of filters and pooling methods. Suppose the matrix is of size $\mathrm{K} \times K$, and the pooling method is a function $\mathrm{g(x)}$, where x is a matrix.

The entry of the resulting matrix from the pooling layer is given by $q_{i,j}=\mathrm{g}([x_{u,v}])$, where $x_{u,v}$ is the sub-matrix of the matrix given ReLU layer from row $(\mathrm{i}-1) \times k+1$ to row $\mathrm{i} \times k$ and column $(\mathrm{j}-1) \times k+1$ to the column $\mathrm{j} \times k$. The result of the pooling layer will be fed to the fully connected layer. Function $g(\cdot)$ usually extracts the most significant entry from x (max pooling) or calculates the average of all entries (average pooling). The incentive of having a pooling layer is primarily to reduce the calculation's complexity. It also reduces the risk of over-fitting the model.

Fully-connected layer. The convolutional and pooling layers extract the input's local features, while the fully connected layer produces an output by considering all previously extracted features. A CNN network might have multiple fully connected layers, each taking an output from the result of the previous fully connected layers. All inputs for fully connected layers are obtained by flattening a matrix from multiple pooling layers and making a high-dimension vector. Each fully connected layer has activation functions, varying depending on the purpose. A fully connected layer will output a vector obtained by a function in the form of $\mathrm{z(x)}=\phi(W \cdot x+b)$, where W and b are learnable matrix and bias vector; each component of the vector is called a neuron. Mathematically, a neuron is obtained by $a_j=\phi\left(\sum_{i=0}^{n} v_i \cdot w_{j,i}+b_j\right)$, where ϕ is the activation function and $w_{i,j}$ is the entry of the matrix in this fully connected layer of row i, column j, b_j is the j-th component of the bias vector $\boldsymbol{b}$. The matrix of a fully connected layer is produced during the training process of a CNN model to minimize the loss function.

2.2. Recurrent Neural Networks and Long Short-Term Memory Network

Recurrent Neural Network. As shown in Figure 2, a recurrent neural network processes a sequence of inputs where the previous input might influence the current output. Mathematically, the input can

be viewed as a sequence of vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$, where $\boldsymbol{x}_i \in R^k$. Each vector is linearly dependent on the ones that precede it. At the same time, the emergence of the entire sequence follows an identical and independent distribution P(X), where $\mathrm{X} \in R^k \times \ldots \times R^k$.

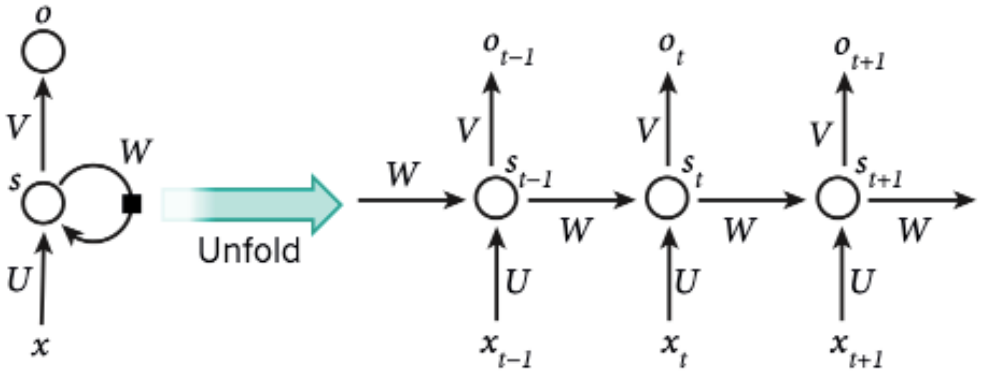

Figure 2. The framework of RNN models. The figure is from [6].

There are three main design patterns of the recurrent neural network in general [5,6]: (i) Recurrent neural networks produce an output after inputting elements of the input sequence. (ii) Recurrent neural networks whose hidden state only has a connection with the output of the previous time. (iii) Recurrent neural networks produce an output after passing all of the sequence of inputs. Generally, a design of (i) requires more calculation expense but is strictly more powerful than the one design under (ii), which makes less connection with the previous stage.

A recurrent neural network generates output based on time series. For any input x_t, which denotes the input at time t, RNN has a hidden state a_{t-1} generated based on the previous input. The new state is generalized by $a_t = \phi_1(a_{t-1}, x_t)$. a_t will be passed to the activation function h(x) before applying to the function ϕ_2 to produce the output $\mathrm{y} = \phi_2(\mathrm{h}(a_t))$. $a_t$could also serve as the state for the input x_{t+1}.

For a model designed under (i), the function $\phi_1$is of the form,

$$\phi_1(a_{t-1}, x_t) = [w_{11} \quad w_{12}] \cdot \begin{bmatrix} a_{t-1} \\ x_t \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{1}$$

Where w_{11}, w_{12} are weight-assigned matrices, and $\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$is the bias vector.

The function ϕ_2 is of the form,

$$\phi_2(\boldsymbol{x}) = \boldsymbol{V} \cdot \boldsymbol{x} + \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{2}$$

Where $\boldsymbol{V}$is a pre-trained matrix that maps the state vector $\boldsymbol{x}$to potential output before adding the bias vector $\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$.

An RNN network is usually trained using the backpropagation through time (BPTT) algorithm. The problem with the recurrent neural network under this algorithm is that information cannot be retained over the long term. This is because of the vanishing/exploding gradient problem while training. This problem happens because the training considers too many steps, which will result in the model's constant/unstable results. The side effect of the limited training step is the loss of information in the long term. The problem emerges for certain in RNN trained under the BPTT algorithm.

A long short-term memory (LSTM) model addresses this problem. The main difference is LSTM introduces different gates to the previous RNN model: forget gate, input gate and output gate, which

controls the impact of the previous state in the model, reassigning the input weight contributing to the current state a_t, reassigning the newly produced $\mathrm{h}(a_t)$ before applying to ϕ_2. These gates efficiently filter out the impact of previous information on the current state and output, which enables the model to retain the memory of previous information. The training of LSTM usually takes longer time than other CNN and RNN networks.

3. Methods of Temporal Stock Prediction

3.1. CNN-based Methods

In [7], the research explores potential variables that can enhance the extraction of critical features from the stock market besides technical indicators and historical data. The study also investigates the correlation between different stock markets.

In specific, the proposed method consists of two variants, including 2D-CNNpred and 3D-CNNpred, to model input data from different dimensions. The network then predicts the trend of closing prices. The method has been validated on various public data, such as the S&P 500, NASDAQ, DJL, etc. It demonstrates that CNN has great potential for stock trend prediction, which can be used in real-world applications. This paper aims to predict Apple's stock price over the past ten years. In this method, the data is first scaled to the range [0,1]. The paper uses a traditional RNN with one hidden layer to handle this task. In experimental results, the RNN with proper timestep shows favourable performance, which states that not all data is helpful for current data prediction. This paper aims to build a model-independent approach that can be deployed in most existing methods. The method first uses the NlFTY-IT index and NIFTY-Pharma index to identify the stock data. Then, those indicators are sent to the network for training. This paper introduces three different architectures: RNN, CNN, and LSTM. From the experiments, CNN performed best in handling time-series stock data. Also, the most current history is more critical in the prediction process.

In [8], the author also improved the CNNpred method and proposed a U-CNNpred. The author claims that the stock trend has similar characteristics over time, which can be modelled by transfer learning. To this end, this paper compares three versions with different training strategies, including base predictor without fine-tuning, partially fine-tuned predictor and complete fine-tuned predictor. Those predictors have different numbers of layers to be fine-tuned. The proposed layer-wise training method produces superior results compared to the supervised training method.

3.2. RNN-based Methods

In [9], given the data from the previous day, the article addresses the following problem: How can the prediction of the Chinese stock price direction change the next day to be more precise. The research also provides a potential solution for choosing the sample size for training. The stock market is sensitive to environmental factors, especially economic and political trends. The choice of the sample size is relatively unclear. The research uses the five-minute bars of one day as the input for the model, including open price, high price, low price, and close price. According to the Chinese stocking market regulations, the input shape is 48*5= 240. The data was normalized. To test the model on Year Y, the data on years Y-2 and Y-3 are used as training data and the data from Year Y-1 is used for validation. The LSTM gives a binary output, 1 for up, 0 for down. The research model combines a deep belief network(DBN) and a long short-term memory model (LSTM). In particular, a 2-layered DBN is used to extract the stock features from the given input and provide the input for LSTM to predict the stock price direction. The DSN is pre-trained using the unsupervised method. After that, DSN and LSTM are trained using the backpropagation(BP) method. The proposed DBN-LSTM model results are compared with the multi-layer perception and pure LSTM models. The DSN-LSTM model performs better than other models (MLP, LSTM). Extracting key features of input data before proceeding to the neural network is a newly developed method for stock price prediction, which leads to better performance of the DSN-LSTM architecture in stock price direction prediction

than other models. Additional data, such as news articles, can be incorporated into the model to make better predictions.

In [10], the researcher wants to predict the closing price of a stock. Stock prediction is hard to conduct because of plenty of affecting factors. It is easy to make human errors when conducting a prediction. The researchers used the closing price of Advanced Micro Devices for 168 working days. The model is trained using the first 156 data points and predicts the next 12. The researcher used the RNN model to predict training using the back-propagation(BP) method. They also develop the LSTM cell. The generated values with the true values have the same trend, and the prediction error is within 5%.

3.3. Combining CNN and RNN for Stock Prediction

In [11], the author improves the general model and generates a CNN-BiLSTM-AM architecture. In this method, the data was processed first. Then, the data was sent to CNN to extract the key features of the data. BiLSTM and AM calculated the result from the previous layer. The paper compared the results with MLP, CNN, RNN, LSTM, BiLSTM, CNN-RNN, BiLSTM-AM. In general, CNN-BiLSTM-AM produced a model with the lowest error but the highest percentage of the explained data, which indicates this method is the best model among all the models.

In [12], this paper uses the previous 5 day data to predict the next day trend of Shenzhen Component Index. The paper proposed a new model by using CNN-BiSLSTM, which is a special variation of the general BiLSTM by replacing last function of producing the result of the model, which made the model more suitable for predicting the stock market. The model was trained by the data from 1991-2021 of Shenzhen Stock component Index. The model was compared with other models including CNN-BiLSTM, RNN, LSTM, while the model shows the most favorable results.

In [13], this paper proposed an easier way to extract the information from the data. This paper incorporates the PCA into the data processing before training the CNN and LSTM models to predict the future trend of stock market. From the experiment results, the combination produced by PCA with highest variation of the dataset takes the lowest time for training, while remaining the lowest error for training compared to the process training using other combination produced by PCA.

In [14], in this paper, the authors generated a CNN-LSTM model to predict the stock price in the next day. The data for the training was pre-processed and as an input for the CNN. As CNN can extract key feature of the input data, the output of CNN is passed to LSTM for training. The paper also produces a comparison of the model including RNN, LSTM and CNN-RNN. The CNN-LSTM shows the most desirable result.

4. Conclusion

In this paper, we aim to provide a board comparison on different models for temporal stock prediction. First, the models can be categorized into three parts, including CNN-based models, RNN-based models and Hybrid models by combining CNN and RNN. We also detail the pre-knowledge on deep learning techniques, such as the components of CNN, RNN and LSTM. Then, we provide a brief introduction on those models and compare them in terms of accuracy, complexity and training and inference time. We conclude that the RNN based models using temporal coherence obtain better performance but has inferior results on inference time. The increasing cost mainly comes from the BPTT method, which needs to train the model sequentially. We claim that our paper can be widely used for the beginners who are interested in the stock prediction and can help the beginners to select the proper methods for their specific goals.

References

[1] Jiang, W. (2021). Applications of deep learning in stock market prediction: recent progress. Expert Systems with Applications, 184, 115537.

[2] Hu, Z., Zhao, Y., & Khushi, M. (2021). A survey of forex and stock price prediction using deep learning. Applied System Innovation, 4 (1), 9.

[3] Shah, J., Vaidya, D., & Shah, M. (2022). A comprehensive review on multiple hybrid deep learning approaches for stock prediction. Intelligent Systems with Applications, 16, 200111.

[4] Rouf, N., Malik, M. B., Arif, T., Sharma, S., Singh, S., Aich, S., & Kim, H. C. (2021). Stock market prediction using machine learning techniques: a decade survey on methodologies, recent developments, and future directions. Electronics, 10(21), 2717.

[5] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

[6] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521 (7553), 436 - 444.

[7] Hoseinzade, E., & Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. Expert Systems with Applications, 129, 273 - 285.

[8] Hoseinzade, E., Haratizadeh, S., & Khoeini, A. (2019). U-cnnpred: A universal cnn-based predictor for stock markets. arXiv preprint arXiv: 1911. 12540.

[9] X Zhang, N Gu, J Chang, H Ye. (2021) Predicting stock price movement using a DBN-RNN.In Applied Artificial Intelligence.

[10] Jahan, I., & Sajal, S. (2018). Stock price prediction using recurrent neural network (RNN) algorithm on time-series data. In 2018 Midwest instruction and computing symposium. Duluth, Minnesota, USA: MSRP.

[11] Lu, W., Li, J., Wang, J., & Qin, L. (2021). A CNN-BiLSTM-AM method for stock price prediction. Neural Computing and Applications, 33 (10), 4741 - 4753.

[12] Wang, H., Wang, J., Cao, L., Li, Y., Sun, Q., & Wang, J. (2021). A stock closing price prediction model based on CNN-BiSLSTM. Complexity, 2021 (1), 5360828.

[13] Rasheed, J., Jamil, A., Hameed, A. A., Ilyas, M., Özyavaş, A., & Ajlouni, N. (2020, October). Improving stock prediction accuracy using cnn and lstm. In 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI) (pp. 1-5). IEEE.

[14] Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-based model to forecast stock prices. Complexity, 2020 (1), 6622927.