Empirical Research on Futures Trading Strategy Based on Time Series Algorithm

Shi Yao[1(✉)], Yan Hongfei[1,3], Ying Siping[1], Chen Chong[2], and Su Qi[1]

[1] Peking University, Beijing, People's Republic of China
{shall,fyanhf,sukiag}@pku.edu.cn, spying0403g@gmail.com
[2] Beijing Normal University, Beijing, People's Republic of China
chenchong@pku.edu.cn
[3] National Engineering Laboratory for Big Data Analysis and Application Technology, Center for Big Data Research, Peking University, Beijing, People's Republic of China

Abstract. This article attempts to establish a trading strategy framework based on deep neural networks for the futures market, which consists of two parts: time series forecasting and trading strategies based on trading signals. In the time series forecasting task, we experimented with three types of methods with different entry points, namely recurrent neural networks with gate structure, networks combining time and frequency domain information, and network structures using attention mechanism. In the trading strategy part, the buying and selling signals and the corresponding trading volume are established according to the prediction results, and trading is conducted with the frequency of hours. In the empirical exploration part, we tested the prediction effect and strategic rate of return of various models on the copper contract. The data shows that in general, the best strategy can obtain a relatively stable income growth that has nothing to do with market fluctuations, but lacks countermeasures for rare external events with greater impact.

Keywords: Futures · Quantitative trading · Deep neural network · Long short-term memory network · Attention mechanism

1 Introduction

Futures are financial contracts that involve the sale of financial instruments or physical commodities for future delivery and are mainly divided into commodity futures and financial futures. A futures contract is a contract for the purchase and sale of futures, and is an agreement between two parties to trade at a specific time when the buyer needs to acquire a specified asset at a specific price

Peking University Grant 2020: "New Ideas for Teaching 2.0" Key Project; MSTC Grant 2019YFC1521203: research, development and demonstration of key technologies for knowledge organization and services for Antiques based on Knowledge Graph; NSFC Grant 61772044.

Z. Dou et al. (Eds.): CCIR 2020, LNCS 12285, pp. 111–123, 2020.
https://doi.org/10.1007/978-3-030-56725-5_9

(also known as the delivery date), the seller delivers the asset at that price, and the asset in exchange is called the underlying. For the same underlying, the futures exchange specifies multiple delivery months, each corresponding to a futures contract, while the main contract represents the highest volume contract. Futures trading is a two-way trading mechanism where buyers and sellers are called long and short respectively.

Through statistical and mathematical methods and computer programming, quantitative trading is based on a large amount of historical data to predict the future market, and follows the probability of formulating the corresponding trading strategy, according to the rules of automated buying and selling operations, in order to seek a stable and high return above the average return.

CTA (commodity trading advisor) generally refers to the investment in futures asset management products, mainly divided into trend strategy and arbitrage strategy, the former occupies the mainstream position. The trend strategy refers to tracking the market trend, going long or short, and is divided into long-term trend tracking strategy, medium-term trend tracking strategy and short-term intraday trend tracking strategy.

Currently, most of the work on financial time series forecasting using deep learning focuses on stock price and index price forecasting, and less work on commodity prices and futures prices [1]. This paper focuses on short-term intraday trend tracking strategies through the method of deep learning to predict prices based on historical data, so as to achieve the effect of tracking the market trend, and in this way to generate buying and selling signals to achieve trading strategies.

2 Time Series Prediction Methods

The futures trading strategy in this paper is based on time series single-step prediction results, and the specific methods used will be described in this chapter.

2.1 Long- and Short-Term Memory Networks

Recurrent neural networks (RNN) were first proposed by David E. Rumelhart et al. in 1986 [2] to apply deep neural networks to the processing of sequence data. The key idea of RNN is that different parts of the model can share parameters through the loop connection of hidden units in adjacent moments, so that the network can be conveniently extended to longer sequences, and also has the ability to process longer sequences, but the gradient disappears or the gradient explodes as the time series grows during training [3]. Hochreiter and Schmidhuber proposed the Long Short-Term Memory Network (LSTM) in 1997 [4], which alleviated the above problems to a certain extent, and the practical results showed very good results and robustness, with great success in tasks such as speech recognition [5] and machine translation [6].

2.2 Methodology Incorporating Frequency Domain Characteristics

LSTM has alleviated the problem of long-distance dependence to some extent, but has not solved it. In order to obtain information about different trends, a class of methods attempts to obtain data of different granularity representing trends across different spans through hierarchical modeling. Koutník et al. [9] divide the hidden layers of the RNN into different modules that are responsible for extracting information only for fixed periods. Chang et al. [10] freely combine different RNN units based on multi-resolution jump connections. Cui et al. [11] propose a CNN network structure at multiple scales. Chung et al. [12] propose an RNN network structure at multiple scales, but also mention that there is no evidence that models of this class can actually capture information across long time spans.

The frequency domain is a coordinate system that describes the fluctuating characteristics of the cycle of things. In general, time series predictions are made based on the time domain. In this paper, we have used the discrete Wavelet transform to modify RNN for the futures price prediction task(wLSTM).

Wavelet transform was proposed by S.G. Mallat in 1989 [17]. Figure 1 gives the network structure of mLSTM after three decompositions. After three decompositions, the final result is a three-stage high-pass filtering result and a final low-pass filtering result. The sequence length after each decomposition is reduced by half due to downsampling, the former is mainly used to obtain frequency domain information, while the latter is used to represent time domain details. Finally, the decomposition results are passed through the LSTM and the output is concated together.

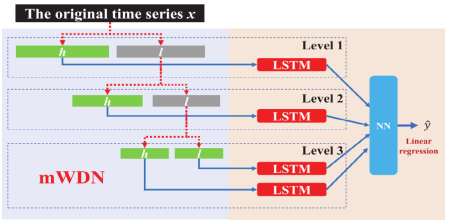

Fig. 1. mLSTM [16]

2.3 Methods of Introducing Attention Mechanisms

The attention mechanism, proposed by Dzmitry Bahdanau et al. in 2015 [18], breaks the limitation of fixed intermediate vectors in end-to-end structures, called bahdanau attention, and is widely used in various types of networks, yielding many variants [19]. During the step-by-step processing of the input sequence by the encoder, all intermediate output results are retained in the context vector. In each step of the encoding, the similarity between the input of the step

and the output of the encoder is calculated separately, which is weighted to obtain the context vector of the step and participate in the calculation together. Attentional mechanisms can likewise be introduced in single-step time series prediction tasks, and long-distance dependencies can be captured directly through this mechanism.

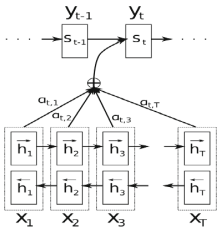

Fig. 2. Bahdanau attention [18]

Ashish Vaswani et al. proposed the Transformer model in 2017 [20], which ditches the traditional RNN structure and deals with sequence problems based entirely on attentional mechanisms, more directly acquiring information over long distances, while improving parallelism. The model employs an encoder-decoder structure.

Considering that the task of this paper is single-step time series prediction, which does not require the use of end-to-end structures, and the amount of data is insufficient to support an overly complex model, a variant of the decoder in Transformer will be used. Shiyang Li et al. experimented with artificially generated data and showed that this variant has a better effect in capturing long sequence information than LSTM [21]. Thus, all references to the Transformer in the following text refer to variants of its decoder.

3 Empirical Analysis

3.1 Data Pre-processing

The subject matter of this article is the copper contract from the Shanghai Futures Exchange, which is a large Chinese futures exchange. Data covers the period from 1 January 2005 to 18 February 2020.

After identifying the subject matter, we need to select the granularity of the data. There's a certain trade-off in terms of data granularity. For data volatility, measured by calculating the standard deviation of the percentage change in the closing price, the statistics are given in Table 1, the model used is considered in a comprehensive manner, the 60-min level of data is selected, and closing price is used.

Table 1. Data characteristics at different data granularities

Frequency	15 min	60 min	1 day
Standard deviation (%)	0.45	0.80	1.24
Data volume	78244	19561	3675

On the dataset division, considering the problem of information leakage after time series disruption, we use the HOLDOUT method to divide the dataset chronologically. To better measure the performance of the algorithm under different scenarios, we used 3 contracts as the test set, during which the market has successively gone through three phases: uptrend, sustained volatility and downtrend. The validation set is also 3 contracts.

Data from the remaining contracts were added to the experiment for pre-training. However, some screening of varieties is required because of the potentially very significant differences in markets between varieties, which violates the independent homodistribution assumptions. Table 2 gives the Pearson correlation coefficients between copper contracts and individual contracts. Through this indicator, the aluminum contract (Al) and rebar contract (Rb) were finally selected as the pre-training dataset.

Table 2. Correlation of each contract with copper contracts

Contract	Ag	Al	Au	Bu	Fu	Hc
Correlation factor	−0.19	0.71	1.24	−0.30	0.52	−0.06
Contract	Ni	Pb	Rb	Sn	Wr	Zn
Correlation factor	−0.06	−0.57	−0.80	0.01	−0.07	−0.44

3.2 Trading Strategy Setting

Based on the predicted closing price for the next hour and the current price, a buy signal is generated if the prediction is to rise and vice versa. The difference between the two prices is recorded as volume (rounded). A set of buy and sell logic can be set based on the buy and sell signals and spreads: if a buy signal is given and the current account is long or open, then the contract with the corresponding volume units is bought; if it is short, then a certain percentage of the holding contract is sold. For sell signals, a reverse treatment using the same rules is sufficient.

In the event of a change in the main contract, the strategy empties the original contract holdings and buys the same number of new contracts. During the buying and selling process, trading-related fees and margin mechanisms are not taken into account due to the low trading frequency of the strategy. In addition, in order to better observe the accuracy of the buy and sell signals, no maximum position is set, taking into account the leverage effect of the margin system in the actual trading of futures.

3.3 Trading Strategies Based on LSTM

LSTM Model Setting. Since the input feature dimension is 1, the number of hidden layer units is set to 32. the length of the observation interval and the number of layers of the LSTM are used as hyperparameters, which are selected by the verification set effect. The loss function uses the most common mean square error for regression tasks [22] and the optimizer uses the RMSProp algorithm.

LSTM Experimental Analysis. Since in a practical quantitative strategy, the accuracy of predicting ups and downs as well as the accuracy of the price can affect its effectiveness. Therefore, two main indicators are used in evaluating the predicted outcome, the root mean square error and the F1 score.

Five sets of experiments were conducted based on the combination of different observation interval lengths P and number of layers L, and the performance of the indicators on the validation set was obtained (Table 3).

Table 3. LSTM validation set indicator statistics

Evaluation indicators	RMSE	F1 score	Precision	Recall
L = 1, P = 32	95.28	0.13	0.47	0.07
L = 1, P = 64	95.03	0.60	0.49	0.76
L = 1, P = 128	94.66	0.65	0.51	0.88
L = 1, P = 256	94.58	0.41	0.49	0.36
L = 3, P = 128	**94.51**	**0.68**	**0.52**	**0.98**

By analyzing the first four rows of data, it can be seen that LSTM can improve the prediction results with a limited lengthening of the observation interval length, but if the observation sequence is too long, it will affect the prediction results. At the same time, increasing the number of layers of LSTM is helpful for the model, but the gap is not significant.

Considering the larger number of participants in the multilayered LSTM model, this paper continues to include the comparison of pre-training (Table 4).

Table 4. Comparison of pre-training effects at L = 3, P = 128

Evaluation indicators	RMSE	F1 score	Precision	Recall
No pre-training	**94.51**	0.682	0.522	0.98
With pre-training	94.97	**0.688**	**0.524**	**1.0**

It can be seen that the addition of pre-training has little effect on the model.

Finally, a 3-layer no-pre-training model with an observation interval of 128 was finally selected as the best model to use for the test set (Table 5).

Table 5. LSTM test set indicator statistics

Evaluation indicators	RMSE	F1 score	Precision	Recall
LSTM	173.19	0.57	0.54	0.62

3.4 Trading Strategies Based on wLSTM

Modelling. The model used in this section is a wLSTM combining wavelet transform and LSTM model which requires a larger space to store information, so the hidden layer unit is set to 64, which is experimentally proven to improve the performance of the model. The model uses several different single-layer LSTMs to process the generated sequences separately. The required hyperparameters are the observed interval length P (32 or 128), the number of transformations D (2 or 3) and whether pre-training X is required.

Experimental Analysis. Based on different combinations of parameters, five sets of experiments were finally conducted and their performance on various indicators on the validation set was obtained. From the first four rows of Table 6, it can be observed that increasing the length of the observation sequence significantly increases the prediction error of the model, and the addition of pre-training is helpful for the model, especially for models with longer observation intervals. For the better performing third model, adding a transformation will get worse results, considering that the observed interval length is only 32, so too much decomposition will not get more trend information, but will increase noise.

Table 6. wLSTM validation set indicator statistics

Evaluation indicators	RMSE	F1 score	Precision	Recall
D = 2, P = 32, X = False	97.84	0.67	0.53	0.92
D = 2, P = 32, X = True	**96.24**	**0.68**	**0.53**	**0.97**
D = 2, P = 128, X = False	137.19	0.61	0.57	0.0.65
D = 2, P = 128, X = True	107.55	0.64	0.52	0.84
D = 3, P = 32, X = True	97.58	0.67	0.52	0.94

Synthesizing these analyses, the twice-decomposed pre-training model with an observation interval of 32 was finally selected as the best model to be used for the test set (Table 7).

Table 7. wLSTM test set indicator statistics

Evaluation indicators	RMSE	F1 score	Precision	Recall
wLSTM	176.94	0.67	0.55	0.85

3.5 Trading Strategies with Attention Mechanisms

Attentive LSTM. According to the previous experimental results, the LSTM model used in this section adopts a 3-layer unidirectional LSTM with an observation interval length of 128 and the remaining parameter settings remain unchanged. In addition, due to the higher demand of the attention mechanism for training data volume, three different training methods were used in the model (Table 8).

Table 8. Attention+LSTM validation set indicator statistics

Evaluation indicators	RMSE	F1 score	Precision	Recall
No pre-training	159.54	0.52	0.55	0.49
Pre-training (Al & Rb)	98.48	**0.68**	**0.52**	**0.99**
Pre-training (all)	**94.65**	**0.68**	**0.52**	0.98

Three sets of experiments were eventually conducted according to different training methods. As can be seen, the performance of the model continues to improve as the amount of pre-training data increases, but considering that there are many contracts with very little correlation to copper contracts in the previous period, the improvement is limited, so no more irrelevant contract data is considered for pre-training.

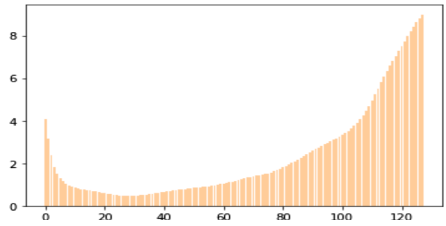

Fig. 3. Attention score

By obtaining the attention scores, we can observe the influence of the LSTM outputs on the results during the model prediction process. As can be seen in Fig. 3, the model mainly focuses on the later stages of the LSTM, but it also uses early stage information to make up for the omissions in the LSTM. Taken together, attentional mechanisms are helpful in capturing information.

Combining the above data, the model with an observation interval of 128 (based on all contractual pre-training from the previous period) was finally selected as the best model to be used in the test set (Table 9).

Table 9. Attention+LSTM test set indicator statistics

Evaluation indicators	RMSE	F1 score	Precision	Recall
Attention+LSTM	177.50	0.60	0.52	0.71

Transformer Model. The number of attention heads in the original paper was 8 and the number of layers was 6. According to Elena Voita et al. [23], an appropriate reduction in the number of attention heads in a machine translation task does not affect the effect and may even improve the effect. Therefore, attempts were made here to use 2, 4, 8 attentional heads, and 1, 2, 3 layers respectively. In addition, the observation interval was set at 128, all pre-trained using full data. Following the remaining settings, the experimental results are given by Table 10.

Table 10. Transformer validation set indicator statistics

Evaluation indicators	RMSE	F1 score	Precision	Recall
$n_{head} = 8, n_{layer} = 1$	95.72	**0.68**	0.52	**0.99**
$n_{head} = 4, n_{layer} = 1$	95.06	0.64	0.51	0.86
$n_{head} = 4, n_{layer} = 2$	**93.93**	**0.68**	**0.54**	0.90
$n_{head} = 4, n_{layer} = 3$	96.57	0.64	0.51	0.84
$n_{head} = 2, n_{layer} = 2$	94.38	0.65	0.53	0.82

From the first two rows, it can be seen that the reduction in the number of heads does not effect the results, while the number of parameters is much reduced. The number of layers derived from 2 to 4 rows is not as high as it should be, and needs to match the amount of data. And the last line can see that the effect of 2 attention heads is reduced compared to 4 attention heads. Further, the attention distribution of each head of the model at $n_{head} = 4$, $n_{layer} = 2$ was visualized by the Fig. 4 gives the four attention heads for each of the two layers. It can be seen that the pattern of capturing features is different between the two layers.

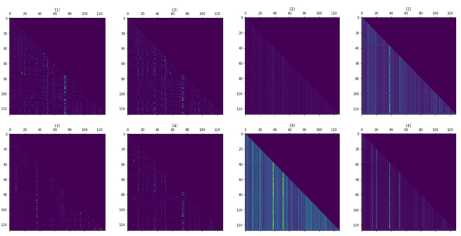

Fig. 4. Visualization of attention scores

Taken together, the model of $n_{head} = 4$, $n_{layer} = 2$ was chosen (Table 11).

Table 11. Transformer test set indicator statistics

Evaluation indicators	RMSE	F1 score	Precision	Recall
Transformer	173.88	0.65	0.55	0.79

3.6 Summary

This chapter uses the model in practice for forecasting and trading strategies, and the final results are summarized in Table 12. It can be seen that a large gap in the RMSE does make a significant difference in the effectiveness of the strategy, but when this gap is within a certain range, the effectiveness of the strategy is no longer linked to the indicator alone, but to the specific market conditions. And there is no clear link between F1 score and strategy performance. In the task of time series forecasting based on closing price information, the performance of the methods is relatively similar and has not yet been able to achieve a significant increase in yield.

Table 12. Summary of strategy results

Evaluation indicators	Rate of return (%)	RMSE	F1 score
Market	−1.62	–	–
LSTM	**10.88**	**173.19**	0.57
wLSTM	8.13	176.94	**0.67**
Attention+LSTM	8.19	177.50	0.60
Transformer	6.46	173.88	0.65

It is worth noticing that an anomalous volatility point was last seen in the test range, which was a huge swing on February 3, the first working day after the Chinese New Year holiday, when it was hit by a special event, the epidemic, that could not be predicted from the index. Therefore, Table 13 shows the strategy indicators with the cut-off time at that point in time, which leads to a completely different conclusion from the previous one, where the yield has a certain correlation with the F1 score in the case of close RMSE.

Overall, the combined RMSE and F1 score provides a better measure of the effectiveness of a quantitative trading strategy based on time series predictions in the absence of a large external market shock, while after a shock, the results of the strategy become uncontrollable due to the misalignment of predictions.

Table 13. Summary of pre-shock strategy results

Evaluation indicators	Rate of return (%)	RMSE	F1 score
Market	2.43	–	–
LSTM	7.97	96.71	0.56
wLSTM	6.50	98.32	**0.68**
Attention+LSTM	9.64	97.28	0.62
Transformer	**12.39**	**96.23**	**0.68**

4 Conclusions and Prospects

This paper establishes a futures trading framework based on time-series forecasting that attempts statistical arbitrage by leveraging historical information about prices. For time series prediction, three different types of approaches are used, namely, LSTM with gate structures, wLSTM combining wavelet transformations with LSTM, and network structures that introduce attentional mechanisms (attention+LSTM, Transformer), all of which aim to obtain more information from long sequences.

Experiments show that the performance gap between the root mean square error of each method is small in the final prediction result, but due to the different characteristics of the network, there is a certain gap between the actual strategy effect, which can be reflected in the F1 score to some extent. In general, the best-performing Transformer model can achieve stable excess gains independent of market ups and downs through both long and short mechanisms. However, neural network models based on internal market information can appear uncontrollable when relatively rare external shocks that cannot be reflected in prices occur, leading to a certain loss of yield.

It can be found through experiments that the common root mean square error is not comprehensive for the measurement of prediction results, and the ability to distinguish between fitting and prediction is poor. More evaluation methods need to be combined, and the loss function and model evaluation function applicable to quantitative transactions can be further explored. In addition, unpredictable external information requires timely stop-loss and adjustment of the strategy, as well as the introduction of online information such as news to help the model make decisions.

References

1. Sezer, O.B., Gudelek, M.U., Ozbayoglu, A.M.: Financial time series forecasting with deep learning: a systematic literature review: 2005–2019. arXiv (2019)
2. Rumelhart, David E., Hinton, Geoffrey E., Williams, Ronald J.: Learning representations by back propagating errors. Nature **323**(6088), 533–536 (1986)

3. Bengio, Y.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw. **5**, 157–166 (1994)
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
5. Graves, A., Mohamed, A.R., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (2013)
6. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. CoRR abs/1409.3215 (2014). http://arxiv.org/abs/1409.3215
7. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
8. Graves, A., Schmidhuber, J.: Offline arabic handwriting recognition with multidimensional recurrent neural networks. In: Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, 8–11 December 2008 (2008)
9. Koutník, J., Greff, K., Gomez, F., et al.: A Clockwork RNN. Computer Science, pp. 1863–1871 (2014)
10. Chang, S., Zhang, Y., Han, W., et al.: Dilated recurrent neural networks (2017)
11. Cui, Z., Chen, W., Chen, Y.: Multi-scale convolutional neural networks for time series classification (2016)
12. Chung, J., Ahn, S., Bengio, Y.: Hierarchical multiscale recurrent neural networks (2016)
13. Zhang, L., Aggarwal, C., Qi, G.-J.: Stock price prediction via discovering multi-frequency trading patterns. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining-KDD 2017, ACM Press the 23rd ACM SIGKDD International Conference - Halifax, NS, Canada, 13–17 August 2017, pp. 2141–2149 (2017)
14. Wei, B., Yue, J., Rao, Y., et al.: A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLoS ONE **12**(7), e0180944 (2017)
15. Hui, L., Tian, H.Q., Pan, D.F., et al.: Forecasting models for wind speed using wavelet, wavelet packet, time series and Artificial Neural Networks. Appl. Energy **107**, 191–208 (2013)
16. Wang, J., Wang, Z., Li, J., et al.: Multilevel wavelet decomposition network for interpretable time series analysis (2018)
17. Mallat, S.G.: A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans. Pattern Anal. Mach. Intell. **11**(7), 674–693 (1989)
18. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate (2014)
19. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 1412–1421. https://www.aclweb.org/anthology/D15-1166
20. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need (2017)
21. Li, S., Jin, X., Xuan, Y., et al.: Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting (2019)

22. Heaton, J., Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. Genet. Program. Evolvable Mach. https://doi.org/10.1007/s10710-017-9314-z
23. Voita, E., Talbot, D., Moiseev, F., et al.: Analyzing multi-head self-attention: specialized heads do the heavy lifting, the rest can be pruned. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (2019). http://dx.doi.org/10.18653/v1/p19-1580