Predict the Future Price Movements of Gold and Bitcoin Based on The Long Short-Term Memory Neural Network Model

Jinchen Wu[1], Yihan Gao[2], Wenyu Wu[1, *]

[1] School of Economics, Lanzhou University, Lanzhou, Gansu, 730100 China

[2] School of Mathematics and Statistics, Lanzhou University, Lanzhou, Gansu, 730100 China

* Corresponding author. Email: wuwenyu220328@163.com

Abstract. Market traders often buy and sell volatile assets to maximize their returns, usually with Bitcoin or gold commissions. We started from November 9, 2016, to predict the future price movements of gold and bitcoin, based on the long short-term memory neural network model. First, we preprocessed the data for outliers and added the prices of gold and bitcoin to the training. At the same time, based on the price of the day, we also established a strategy model based on the Sharpe ratio and particle swarm algorithm. Then, through the sensitivity analysis, we found that as the transaction fee increases, the number of transactions of gold and Bitcoin decreases significantly, and the value decreases. On the contrary, there is the same theory, which proves that our model is perfect.

Keywords: LSTM Neural Network, Sharpe Rate, Particle Swarm Optimization.

1. Instruction

Traders have asked us to develop a model that uses only the daily price stream in the past to date to determine whether traders should buy, hold or sell assets in their portfolio daily. We will start at $1000 on November 9, 2016. We will use a five-year trading period from November 9, 2016, to October 9, 2021. On each trading day, the trader will have a portfolio consisting of cash, gold, and bitcoin [C, G, B]. Dollars, Troy Ounce and Bitcoin. The initial state is [1000, 0, 0]. The commission cost per transaction (buy and sell) is α% of the transaction amount.

We have predicted the future price trend of gold and bitcoin based on the neural network model of long-term and short-term memory. First, we preprocess the data with outliers and add the prices of gold and bitcoin to the training set.

Meanwhile, based on the price of the day, we also established a strategy model based on sharp interest rate and particle swarm optimization algorithm. In order to quantify the relationship between investment risk and return every day, we introduced a sharp ratio, set the objective function, set the adjustment position and the sharp rate on the next three days. When the objective function is the largest, it is the largest risk and returns.

Once again, we established a comprehensive planning model for cash, gold, and currency through the planning process.

We can use the neural network model based on the previous data, get the latest price through prediction, and get the best strategy through the optimization of the planning model. Secondly, we need to determine the sensitivity of the strategy to transaction costs. Through sensitivity analysis, we found that the model is in good agreement, especially if the transaction fee increases, the number of transactions of gold and Bitcoin will be significantly reduced, which is consistent with our understanding. Common sense is also the same.

2. Establishment and solution of model

Based on the above analysis of the problem, the following will establish the mathematical model, which explains the establishment process. Then, it is used the mathematical model to solve the problem.

2.1. Prediction model based on LSTM neural network

In order to predict the future price movements of gold and bitcoin, our team solves this problem based on long short-term memory (LSTM) neural network.

2.1.1. A brief introduction to LSTM

Among various neural network models, recurrent neural networks (RNN) are designed to handle time-series information better. It introduces state variables to store past information and determines the current output and the current input.

For a long time, latent variable models have problems of long-term information preservation and short-term input missing. One of the earliest solutions to this problem was long short-term memory (LSTM).

2.1.2. A brief introduction to LSTM

The logic gates of computers inspire the design of long short-term memory networks. Long short-term memory networks introduce memory cells or cells for short. Memories have the same shape as hidden states and are designed to record additional information.

To control memory cells, we need many gates. One of the gates is used to output items from the cell, and we call it the output gate. Another gate is used to decide when to read data into the cell, and we call it the input gate. We also need a mechanism to reset the contents of the cell, governed by a forget gate, a design that allows us to decide when to remember or ignore the input in the hidden state through a dedicated mechanism.

2.1.3. Basic Architecture

The STM neural network structure is a variant structure of the species RNN proposed by Sepp H ochreiter and Jurgen Schmidhuber in 1997. As a variant of the RNN neural network, the biggest difference between the LSTM NN network is that it draws on the selective input and selective forgetting mechanism of the human brain, introducing three "gate" structures of forgetting gate, input gate and output gate, and a memory unit to receive information from the afferent neural network selectively. In them, the "gate" structure belonging to the logical unit is only responsible for completing the set of weights at the edge of the neural network connected to the memory unit without impact on other neuron nodes. The LSTM neural network structure is shown in Figure 1.

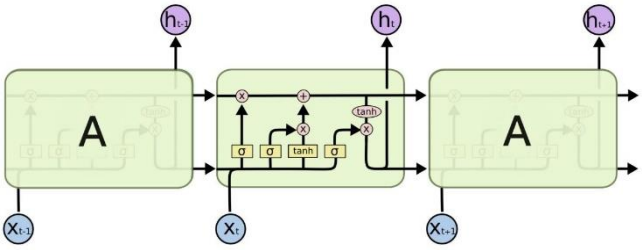


Figure 1 Structural diagram of the LSTM neural network

In the network structure of LSTM, the forgetting door is responsible for receiving the previous moment information to the current point, which determines the previous memory unit state can be retained to the current moment, the purpose of the logical unit is to forget the useless information, forgetting door according to the output of the previous moment and the input of the information retention degree: *ct*−1 *ct ht*−1 *xt ft*

f_t = sigmoid (W_f h_{t-1}, *xt* $+b_f$)

After the action of the sigmoid activation function, the f_t. The more information remains, the values range between 0 and 1, ftCloser to 1. After retaining the previous information, the neural network also needs input x from the current moment$_t$. New information is generated in to update the memory unit status c in$_t$。

The effect of the input gate is performed according to h_{t-1}And, x_t. To determine which information can be added to the previous moment of memory unit state c_t 1In and updated to the new memory unit

state c_t. Firstly, the input door is based on the output h of the previous moment$_{t-1}$And the input x for the current moment$_t$

Figure out i_t:

$$i_t = \text{sigmoid}\,(W_i[h_{t-1}, x_t] + b_i)$$

In formula, W_i note the hidden layer output h at the previous moment -1 Enter x, and at the current moment $_t$ The parameter matrix of, b_f represents a biased item. Then the z is calculated $_t$:

$$z_t = \tanh\left(W_g[h_{t-1}, x_t] + b_g\right)$$

Updated current moment memory unit status c_t For:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot z_t$$

The function of the output gate is to determine how much information at the current moment can be output, according to the output h of the previous moment 1 And the input x for the current moment. Computing the information output degree O_t :

$$o_t = \text{sigmoid}\,(W_o[h_{t-1}, x_t] + b_o)$$

The final output is the output h at the current moment t :

$$h_t = o_t \cdot \tanh\,(c_t)$$

At this point, the forward propagation of the circulation body in the LSTM neural network ends, and then the reverse propagation of error corrects the weight coefficients in the process of error back transmission. The core idea is to find the partial derivative of the parameters in each gate unit and more weights according to the direction of gradient convergence, which will not be repeated here.

It can be seen from the above derivation that the structural design of the LSTM neural network is very dexterous and very suitable for modeling and analysis of financial time series. Meanwhile, the emergence of the LSTM neural network also solves the problem that RNN neural networks cannot deal with and improves the problem of gradient disappearance and gradient explosion during model training. By adding multiple "gate" structure logic units in each layer, LSTM neural network in the error propagation directly through the "gate" improves the gradient disappearance and divergence so that no matter how far the data will not gradient disappear entirely. In addition, LSTM neural network gradient is relatively stable in propagation, will not like the RNN neural network gradient same weight matrix backpropagation phenomenon.

2.1.4. Strategies for Optimization Algorithms Normalization

In processing data, due to the large difference between the size range and dimension of the data itself, the non-standard data will lead to longer training time and worse training effect of the model. Therefore, we need the data to be normalized.

In this paper, we choose the more common maximum and minimum standardization. In the data preprocessing stage, we map the value of the data to [0, 1], and after the prediction result is obtained, the obtained prediction value is reversed. Normalized.

Since the prices of gold and bitcoin are updated every day, we can add the latest prices of gold and bitcoin into our training set every day, and under this, our training set has been expanded, thus Ability to better predict future gold and bitcoin prices.

2.1.5. Forecast result

We bring in the data, add the latest gold and bitcoin prices to the training set every day for continuous training, record the predicted values, and the fitting results are as follows:

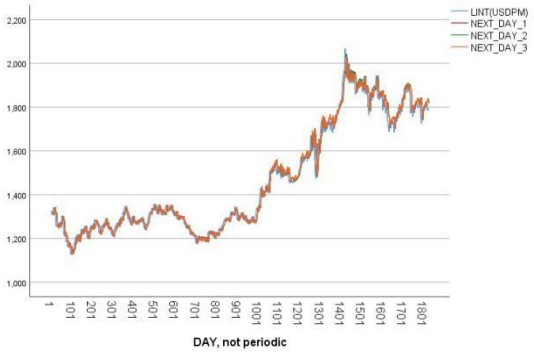


Figure 2 Gold price prediction renderings

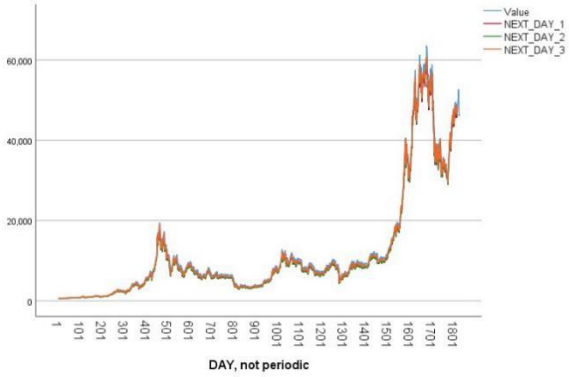


Figure 3 Bitcoin price prediction renderings

2.2. Policy Model Based on Sharpe Rate and Particle Swarm Optimization

2.2.1 Sharpe ratio

To facilitate the calculation of the planning model in the future, it is assumed that the proportion of cash, gold, and Bitcoin held in the total assets on the kth day is recorded as $[c_k,g_k,b_k]$, So we have the relation

$$c_k + g_k + b_k = 1$$

We make the decision variable a two-tuple $[\Delta G_k, \Delta B_k]$ are the changes in the proportion of gold and bitcoin, respectively.

It is supposed that the increase of cash, gold, and bitcoin on day k are respectively $[0.00547\%,\overline{G_k}, \overline{B_k}]$, According to the prediction result, on the kth day, we will have the data of the kth day and the prediction result of the kth day. In order to better quantify the relationship between daily investment risk and return, we introduce the Sharpe ratio. The Sharpe ratio (Sharpe, 1966) proposed the Sharpe ratio based on modern portfolio theory. The Sharpe ratio pays attention to the return of the asset and pays attention to the risk of the asset. It measures the return of the asset after adjusting the risk and is the price display of the unit risk. Since the Sharpe ratio comprehensively reflects the risk-return characteristics of the capital market, it has been widely used in evaluating asset portfolios' performance. The capital market's operating efficiency is judged. Its mathematical expression is:$SR_p = \frac{E(r_p)-r_f}{\sigma_p}$

$$E(r_p) = c_k \cdot E(r_c) + g_k \cdot E(r_g) + b_k \cdot E(r_b)$$

$$\delta_p = \sqrt{g_k\delta_g^2 + b_k\delta_b^2 + 2g_k b_k \mathrm{cov}(G,B)}$$

In the formula, $E(r_p), \sigma_p, r_f$ are the expected return, the standard deviation of the return, and the return of the risk-free asset during the observation period, respectively.

According to the planning model, on the day, we set the objective function to $f(k)$. It is set to the value of the Sharpe ratio of the third day when the position is adjusted today and the next three days remain unchanged. When this objective function reaches the maximum, it is the state where the risk and return are balanced.

Taking the end of day k as an example, the proportion of cash, gold, and bitcoin in total assets is $[c_k,g_k,b_k]$. At the end of each day, the end of the k-th day, the ups and downs need to be settled first.

If the increase of cash, gold, and bitcoin on that day is $[0.00547\%,\overline{G_k},\overline{B_k}]$, the proportion will become $[(1+0.00547\%)c_k,(1+\overline{G_k})g_k,(1+\overline{B_k})b_k]$, It can be seen that the sum of the proportions here may no longer add up to 100%, which is due to changes in the total holding amount. After we have completed all changes, we will re-normalize to perform the kth +1 day for repeated actions. Then, trade to adjust the position and settle the transaction fee.

When only the purchase of gold is required, record the change in the proportion of gold as deltaG_k>0. At this time, the change in the proportion of cash is 1.0101*deltaG_k (because the buying rate is 1%, 101 blocks can not buy 100 pieces of gold). When only the transaction of selling gold is required, record the change of the gold amount as deltaG_k<0, at this time, the change of the cash proportion is 0.99*deltaG_k. When only the transaction of buying Bitcoin is required, record the change in the proportion of Bitcoin amount as deltaB_k>0. At this time, the change in the proportion of cash is 1.0204*deltaB_k. When only the transaction of selling Bitcoin is required, Note that the change in the proportion of Bitcoin amount is deltaG_k<0, at this time, the change in the proportion of cash is 0.98*deltaG_k.

After the transaction fee settlement is completed, the final day's cash, gold, and bitcoin will account for the proportion of total assets as $[(1+0.00547\%)c_k+(1\pm0.01)\Delta G_k+(1\pm0.02)\Delta B_k,(1+\overline{G_k})g_k-\Delta G_k,(1+\overline{B_k})b_k-\Delta B_k]$

Then, we normalize it and multiply the sum of the three proportions by the total amount of yesterday to get today's total amount and rate of return and today's proportion.

According to our assumption, for the function f_k, The amounts will remain the same for the next three days, therefore, in the following process ΔG_{k+1} andΔB_{k+1} Directly equal to 0 until the third day, and then calculate the Sharpe rate of the third day. Constraints: After all operations are performed on the day, the proportion of cash, gold, and Bitcoin in total assets should be greater than or equal to 0, that is,

$$\begin{cases}(1+0.00547\%)c_k+(1\pm0.01)\Delta G_k+(1\pm0.02)\Delta B_k\geq 0\\ (1+\overline{G_k})g_k-\Delta G_k\geq 0\\ (1+\overline{B_k})b_k-\Delta B_k\geq 0\end{cases}$$

2.2.2 Particle swarm algorithm.

In seeking the optimal trading strategy, to obtain the optimal solution of the objective function, we use the particle swarm algorithm to simulate the optimization process and seek the optimal solution.

Particle swarm optimization (PSO) is a biological heuristic method in computational intelligence, which belongs to a kind of swarm intelligence optimization algorithm.

Particle swarm optimization (PSO) originated from the study of bird predation behavior. When birds prey, the simplest and most limited strategy for finding food is to search around the bird that is currently closest to the food.

The advantage of PSO is that it is simple and easy to implement, and there is no adjustment of many parameters. It has been widely used in function optimization, neural network training, fuzzy system control, and other application fields of genetic algorithms.

The particle swarm algorithm simulates birds in a flock by designing a massless particle. The particle has only two properties: speed and position. The speed represents the movement speed, and the position represents the direction of movement. Each particle independently searches for the optimal solution in the search space and records it as the current individual extremum. The individual extremum is shared with other particles in the entire particle swarm. It finds the optimal individual

extremum as the entire particle the current global optimal solution of the swarm, all particles in the particle swarm adjust their speed and position according to the current individual extreme value found by themselves and the current global optimal solution shared by the entire particle swarm.

PSO is initialized as a group of random particles (random solution). Then iteratively find the optimal solution. In each iteration, the particle updates itself by tracking two "extremes" (pbest, gbest). After finding these two optimal values, the particle updates its velocity and position by the following formula.

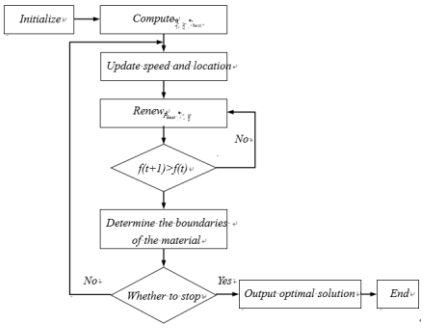


Figure 4 Flow chart of the P S O algorithm

To obtain the optimal value of each variable further, the particle group algorithm (PSO) is introduced to solve it. The idea of particle group algorithm comes from the foraging process of birds and belongs to the category of intelligent algorithms. At first, each bird is randomly in different positions, and they cannot determine the optimal foraging point when each bird's flight speed and direction are random. Over time, the birds transmit foraging information to each other so that each bird can judge the foraging value of the current location based on its own experience and peers and then start in the direction of greater foraging value. The particle swarm algorithm is designed based on the characteristics of birds foraging, abstracts the birds into particles, quantifies the flight direction and speed of the birds as the various dimensions and instantaneous velocity of the particles, and simulates the best decision-making process.

Based on the above discussion, the particle velocity and position:

$$vit{+}1 = \alpha vit + \beta r1(xit - xibest) + \gamma r2(xit - xbest)$$

$$xit{+}1 = xit + vit{+}1$$

Where α represents the weight of inertia, β the self-learning factor, and γ the group information transfer factor r_1 and, r_2Represents the random number within the [0,1] range, used to increase the randomness of the search. It should be noted that the position and velocity of the particle are the vectors, and the formula design follows the principle of the vector operation.

According to the above model process, when the Sharpe rate on the third day is used as the objective function to take the maximum value, we can obtain that the final 1,000 yuan can get 128,675 yuan.

3. Results

The neural network is used for time series prediction, and the prediction effect is good, and the hypothesis test of the goodness of fit is carried out. The particle swarm optimization algorithm is an iterative optimization algorithm similar to the genetic algorithm. The system is initialized to a set of random solutions and iteratively finds the optimal value. Compared with the genetic algorithm, the advantage of PSO is that it is simple and easy to implement, and there are not many parameters that need to be adjusted.

The first nine days are not profitable for traders. Neural networks are similar to using a black box to make predictions. Compared to traditional machine learning algorithms, neural networks usually require more data, at least thousands of millions of labeled samples. While many machine learning problems can be solved well with less data if other algorithms are used. The algorithm is complex

and requires a large amount of data. The coding of network weights and the selection of genetic operators are sometimes troublesome, and the errors are relatively difficult to control

References

[1] Ilin Igor V., Levina Anastasia I., Dubgorn Alissa S., Abran Alain. Investment Models for Enterprise Architecture (EA) and IT Architecture Projects within the Open Innovation Concept [J]. Journal of Open Innovation: Technology, Market, and Complexity,2021,7(1):

[2] Wang D. Quantitative Investment Model Based on Data Mining[J]. Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería,2020:

[3] Shanwen Yan. Research and Investment Model of Stock Index Fluctuation based on Entropy Method [J]. World Scientific Research Journal,2020,6(8):

[4] Matthias Ondra, Christoph Hilscher. A probabilistically constrained extension of the integrated portfolio investment model [J]. Energy Reports,2020,6(Supl.1):

[5] T. Luo, Noura Metawa. Intelligent algorithm of optimal investment model under stochastic interest rate and stochastic volatility. [J]. Journal of Intelligent and Fuzzy Systems,2019,37(1):

[6] Nie Jianquan. A Stock Investment Strategy Based on Quantitative Analysis [J]. Economic Management Journal,2021,10(2):

[7] Minsoo Kim, Pyunghoi Koo. A Study on Big Data Based Investment Strategy Using Internet Search Trends [J]. Journal of the Korean Operations Research and Management Science Society,2013,38(4):

[8] Liqi Kong. Application of Quantitative Investment Principles in Market of Financial Derivatives [J]. Scientific Journal of Economics and Management Research,2021,3(1)