Learning Optimal Q-Function Using Deep Boltzmann Machine for Reliable Trading of Cryptocurrency

Seok-Jun Bu and Sung-Bae Cho(✉)

Department of Computer Science, Yonsei University, Seoul, Republic of Korea
{sjbuhan, sbcho}@yonsei.ac.kr

Abstract. The explosive price volatility from the end of 2017 to January 2018 shows that bitcoin is a high risk asset. The deep reinforcement algorithm is straightforward idea for directly outputs the market management actions to achieve higher profit instead of higher price-prediction accuracy. However, existing deep reinforcement learning algorithms including Q-learning are also limited to problems caused by enormous searching space. We propose a combination of double Q-network and unsupervised pre-training using Deep Boltzmann Machine (DBM) to generate and enhance the optimal Q-function in cryptocurrency trading. We obtained the profit of 2,686% in simulation, whereas the best conventional model had that of 2,087% for the same period of test. In addition, our model records 24% of profit while market price significantly drops by −64%.

Keywords: Deep reinforcement learning · Q-network
Deep Boltzmann Machine · Portfolio management

1 Introduction

Bitcoin is a peer-to-peer, decentralized electronic cash protocol [1]. The explosive price volatility from the end of 2017 to January 2018 shows that bitcoin is a high risk asset that is insufficient to function as a currency. As shown in Fig. 1, the price has plummeted to the $6,000 from $13,000 and shows the extreme volatility of crypto-currency investment. Despite some potential threats, the cyptocurrency trading has become more active and attracts more attention both from the business and academia. In particular, Massive time-series data collected every single minute and single transaction from a market valued at $820 billion is attractive in terms of volume and volatility.

The term portfolio management is the decision making process of allocating an amount of asset into different financial investment products to maximize the profit and minimize the risk [2]. Many of the existing portfolio management methods that applying machine learning algorithm defines some rules based on domain knowledge, but some of the human-defined rules or even expert domain knowledge are not sufficient to deal with the market dynamics.

Besides, it is impossible to tune the parameters of the model with the supervised-manner in order to derive the optimal action from the posterior probability distribution

H. Yin et al. (Eds.): IDEAL 2018, LNCS 11314, pp. 468–480, 2018.
https://doi.org/10.1007/978-3-030-03493-1_49

Fig. 1. The volatility of the bitcoin price in the past years

of the state of the cryptocurrency market. In this paper, we propose the combination of two methods: Double Q-network and Deep Boltzmann Machine (DBM).

On the one hand, reinforcement learning using Q-network is a well-known method of learning that acts to maximize some measure of future payoff or reward [3]. It derives the optimal solution for trading based on Q-network that outputs adequate action from specific state. Double Q-network is a method designed to cope with non-stationary problems that arise when using only one neural network [4]. On the other hand, unsupervised learning using DBM is an effective method of estimating posterior probability distribution [5]. DBM is a method for modeling the prior distribution of the hidden layer on the input values of the neural network [6]. In order to find the optimal action for the cryptocurrency market state, the parameters of the neural network are tuned in an enormous search space. We propose the encoding network that is pre-trained with market states to reduce the search space.

With the combination of double Q-network and DBM, our trading method records 2,686% of profit in simulation while existing best model records 2,087% for same test period without domain knowledge or human-generated rules. Remarkably, our model for the same test periods as the existing models, including deep-learning based models. To analysis our model, we visualized the output decisions using t-Stochastic Neighbor Embedding (t-SNE) algorithm.

The remainder of this paper is organized as follows. In Sect. 2, we review existing trading algorithms or models based on machine learning methods and clarify the contributions of this paper by discussing the differences. Section 3 explains how the market history is encoded and modeled using double Q-network and DBM pre-training algorithm. The performance of our model is evaluated in Sect. 4 through various experiments, including visualizations of the decisions the model made, measurements of performance and comparisons with existing algorithms.

2 Related Works

In this section, we introduce various works based on machine learning methods for comparison with the proposed trading agent. Almost most of the cryptocurrency trading research was done in late 2010, several studies were included to introduce the basic framework of the study.

Huang et al. used a basic machine learning algorithm to model the weekly volatility of the stock market [7]. Although the domains are different, the weekly price prediction

Table 1. Related works on financial trading using machine learning algorithm

Authors	Method	Domain
Huang [7]	SVM	Stock price prediction
Schumaker [8]	SVM	Stock price prediction
Patel [9]	NB, RF, SVM, NN	Stock price prediction
McNally [10]	ARIMA, RNN, LSTM	Cryptocurrency price prediction
Bell [13]	Wavelet, SVM	Cryptocurrency trading
Zbikowski [14]	EMA, SVM	Cryptocurrency trading
Jiang [12]	DQN(CNN)	Cryptocurrency trading

also significant in terms of feasibility. Schumaker et al. discuss about the necessity of external information to predict stock market [8]. They categorized the difficulty of price prediction into fundamental and technical aspect and modeled stock market with external text data from web source. The information from quarterly reports or breaking news stories made the price prediction more accurate. Patel et al. applied various machine learning methods to predict stock price and compared them [9]. Due to the uncertainty of the market fluctuation, they achieved under 80% of classification accuracy and showed the random forest algorithm is effective with short-term prediction.

In the late of 2010, a trading agent that includes existing prediction engine was studied in cryptocurrency domain, as well as stock market domain. McNally et al. applied wavelet transform to encode cryptocurrency history and modeled the encoded feature using deep learning algorithms including LSTM [10]. They achieved about 50\ % of classification accuracy, but contributed to the modeling the sequence of cryptocurrency price as encoded images using wavelet transformation. Amjad et al. proposed the bitcoin trading agent based on predicted price [11].

Among various studies suggesting a agent for modeling cryptocurrency markets and making optimal decision or action, our study using double Q-network and pre-training using DBM was inspired by the study of Jiang et al. [12]. Unlike previous approach, Jiang et al. do not include a prediction engine inside the agent. The Q-network directly map the input state into output action to deal with two problems. The first reason is to exclude the domain knowledge of the person, and the second is high accuracy in predicting price movement is usually difficult to achieve. Table 1 contains a summary of the methods discussed above.

3 Proposed Method

In this section, we present the architecture of the proposed trading agent that makes optimal decision to states by combining double Q-network and encoding network. The trading agent consists mainly of three components: the agent that in the form of two neural networks are connected in series, the unsupervised learning module and the environment.

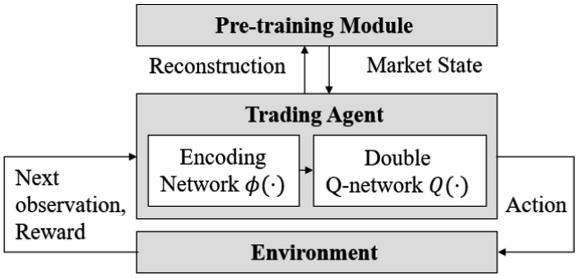

Fig. 2. Main components of the trading agent

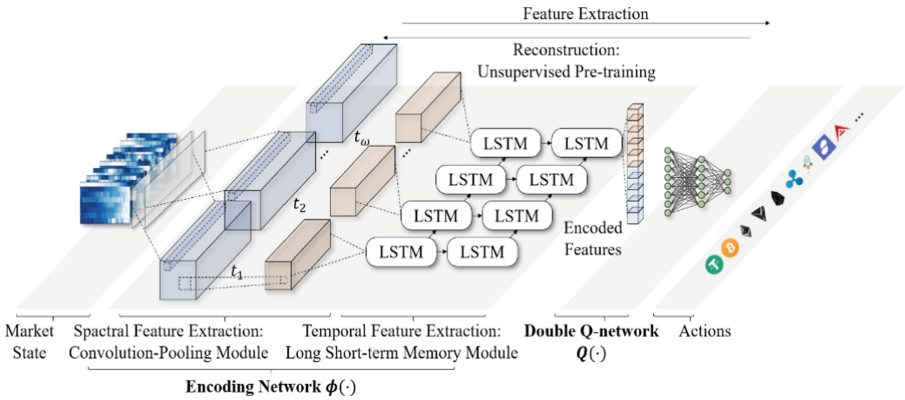

Fig. 3. Deep neural architecture of trading agent consisting of pre-trained encoding network and double Q-network

3.1 Overview

The proposed cryptocurrency trading agent consists of three main components, as shown in Fig. 2. As the first component, The purpose of the encoding network and the double Q-network is to map the input state to appropriate action directly. Figure 3 shows the architecture of the encoding network and the double Q-network in detail. Compared to conventional Q-learning algorithms use Q-tables to map states and actions [15], the generalization performance of double Q-network can reduce the computational complexity. The encoding network consists of a Convolutional Neural Network (CNN) which has a strengths in a wide range of vision fields [16], and a Long Short-Term Memory (LSTM) which has a well known recurrent neural network for time series modeling [17].

The second main component is pre-training learning module. The encoding network is designed to extract the effective information from the state space and deliver it to the double Q-network. Unlike the typical classification problem, by learning to reduce the difference between the probability distribution of the input state and the

posterior probability distribution of the hidden neurons [18], encoding network can be effectively pre-trained.

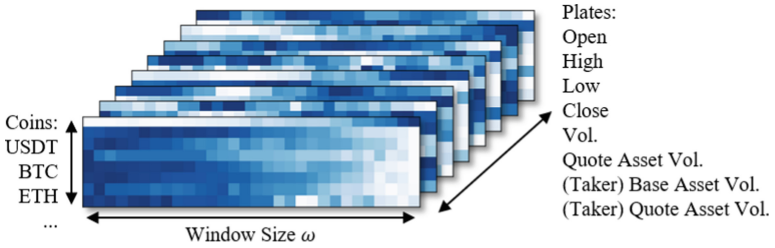

Fig. 4. Preprocessed 3D input state of market history

The last main component is the environment, which is the core of the Q-learning algorithm. For each of action from a double Q-network, the environment updates the current state and sends back to the Q-network and determines the reward for the action taken [19]. We chose eight coins based on trading volume and defined actions as 16-dimensional vectors which mean buying/selling weights for each coin. In order to model the time-series features of price, we preprocessed market state by sliding window as shown in Fig. 4. The market state is represented by 3D-blocks named as history block, with vertical axis (window size ω), horizontal axis (coin type) and depth axis (plates). The reward for the proposed agent is defined as +1 if the asset is increased, 0 if remain still and −1 if it is decreased.

3.2 Unsupervised Pretraining of Encoding Network

The encoding network consists on CNN which models the spatial feature using convolution-pooling operations $\phi_c(\cdot)$, $\phi_p(\cdot)$, and LSTM which models the temporal feature using a gate operation and a recurrent loop $\phi_s(\cdot)$ [20]. For every history block $X = \{x_1, \ldots, x_n\}$, the encoding function $\phi(\cdot)$ is defined as below with step t:

$$\phi(x_t) = \phi_p(\phi_c(x_t)) \tag{1}$$

The convolution operation, which preserves the spatial relationships between features by learning filters that extract correlations, is known to reduce the translational variance between features. The hidden correlations between features in cryptocurrencies and its financial attributes are modeled as a feature-map through emphasis or distortion during the convolution operation. Given t th input state, the encoding networks performs the convolution operation $\phi_c(\cdot)$ using $m \times m \times m$ sized filter w:

$$\phi_c^l(x_t) = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \sum_{c=0}^{m-1} w_{abc} x_{(i+a)(j+b)(k+c)}^{l-1} \tag{2}$$

The summary statistic of nearby outputs is derived from $\phi_p(\cdot)$ by max-pooling operation [21]. Because the dimension of the output vectors from the convolutional

layer is increased by the number of convolutional filters, it must be carefully controlled. Pooling refers to a dimension reduction process used in CNN in order to impose a capacity bottleneck and facilitate faster computation. The max-pooling operation has effects on feature selection and dimension reduction under $k \times k \times k$ sized area with pooling stride τ:

$$\phi_p^l(x_t) = max\, x_{t_{ijk\times\tau}}^{l-1} \tag{3}$$

The purpose of encoding network is to extract the effective information from the state space and deliver it to the double Q-network. Since the size of the available state space, it is difficult to find optimal parameters of encoding network being connected in series with the double Q-network. Focusing on the markov property of history block, we formalize the window size parameter ω and markov chain of sequence of price as below:

$$p(x_t|x_{t-1}, \ldots, x_1) = p(x_t|x_{t-1}, \ldots, x_{t-\omega}) \tag{4}$$

We define the energy $E(x_t, h)$, partition constant z and parameterize the joint probability x, h as $p(x_t, h)$ as shown below:

$$E(x_t, h) = -\phi_s\big(\phi_p(\phi_c(x_t))\big), z = \sum\nolimits_{i,j} e^{-E\left(xt_i, h_j\right)} \tag{5}$$

$$p(x_t) = \sum\nolimits_i p(x_{ti}, h) = \sum\nolimits_i \frac{e^{-E(x_t,h)}}{z} \tag{6}$$

After we define the probability distribution of the hidden layer and the input state x_t above, we define the loss of the encoding network L_ϕ using the Kullback-Leibler divergence Δ_{KL} between the observed distribution:

$$L_\phi = \sum\nolimits_{t=1}^{N} L(\theta|x_t) = \Delta_{KL}(x_t, p(x_t|\theta)) = \left(\left(-\sum x_t\, ln\, p(x_t|\theta)\right) - \left(-\sum x_t\, ln\, x_t\right)\right) \tag{7}$$

3.3 Double Q-Network and Market Environment

The objective of Q-network training based on reinforcement learning is to expect to output appropriate action in response to the generalization performance of the neural network. The minimizing process of the loss function L_Q using stochastic gradient descent algorithm considers the reward of next step. To cope with the non-stationary problem, Hasselt et al. proposed the double Q-network that copies the Q-network into two networks and fix the target of Q-network [4]. We formalized out objective as a loss of double Q-network L_Q, where reward r and decaying hyperparameter γ :

$$L_Q = min_\theta \sum_{t=0}^{n} \left[\widehat{Q}(\phi(x_t|\theta)) - \left(r_t + \gamma \widehat{Q}\left(\phi(x_{t+1})|\bar{\theta}\right) \right) \right]^2 \tag{8}$$

The details of the training algorithm used are presented in Fig. 5. In each episode, Q-network takes a different action by the random probability ϵ even if it is the same step. Since the hyperparameter ϵ, also called exploration rate, can be the method to find a new buying/selling strategy that can raise profit but also can be the pithole of the entire method. Once the training process starts, the encoding network $\phi(\cdot)$ pre-trained by unsupervised manner. For the steps that make up each episode, Q-network repeatedly receives the state and outputs an action vector. Each element of the action vector represents the weight of the asset movement. In this paper, we have selected eight coins based on volume, so we output a 16-dimensional action vector consisting of buy/sell actions for each coin.

Pretrain $\phi(\cdot)$ by unsupervised learning
Initialize replay memory D to capacity d
Initialize double Q-network Q and target-network $\hat{Q}$
Initialize exploration rate ϵ
for episode $e = 1, E$ **do**
  Initialize score
  for step $t = 1, T$ **do**
    Preprocess state $\phi(s_t)$
    Set $a_t = \begin{cases} \text{Random Sequences}, & \text{if } \epsilon_t \leq \epsilon. \\ \text{Q}(\phi(s_t)|\theta), & \text{otherwise.} \end{cases}$
    Execute a_t and observe reward r_t, next state s_{t+1}
    Store (s_t, a_t, r_t, s_{t+1}) in D
    Sample random minibatch if step>memory
    Set $y_t = \begin{cases} \text{r}_j, & \text{if episode terminates.} \\ \text{r}_j + \hat{Q}(\phi(s_j|\bar{\theta})), & \text{otherwise.} \end{cases}$
    Perform a gradient descent on L_Q
    Set $\hat{Q} = Q$ for every c steps
  end for
end for

Fig. 5. The training algorithm of proposed double Q-network

Putting it all together, the main contribution of our study is to propose a combination of the two existing methods that can efficiently model the history of financial or derivatives markets. First, we introduce a framework of existing research to map the vast state space and action space corresponding the cryptocurrency market. Second, we modified and adapted the existing DBM pre-training algorithm to reduce the parameter space of encoding network. In addition, we collected a large amount of cryptocurrency market historical data and preprocessed.

4 Experimental Results

In this section, we evaluated our agent through various experiments, including measurements of performance, comparisons with existing algorithms, parameter optimization and visualizations of the decisions the agent made. While short trades that are difficult for humans to understand, the final profit is the highest among the existing algorithms. To cope with the high computational complexity that is proportional to the amount of historical data of the cryptocurrency transactions, we used four NVIDIA GTX1080-Ti to learn a large number of agents.

4.1 Comparisons with Existing Methods

For quantitative comparison with existing studies, we conducted the back-test during test period 2016/05/14-2016/07/03. We selected bitcoin and seven altcoins which had the highest trading volumes during the period, as assets. In order to compare performance intuitively, the score was defined as the ratio between total value after investment and initial value.

Table 2. Score and risk measure comparison with other algorithms

Algorithm	Score	Sharpe ratio	Maximum drawdown
Uniform buy and hold	0.8760	−1.5413	0.3820
Best single asset	1.3776	1.1257	0.2883
Universal portfolio [22]	1.0484	−1.0110	0.3309
Online neuton step [23]	2.6482	1.0458	0.2787
PAMR [24]	21.8728	0.0062	0.3530
DQN (CNN) [12]	16.3053	0.0368	0.2960
DQN (Ours)	27.8684	0.0027	0.4627

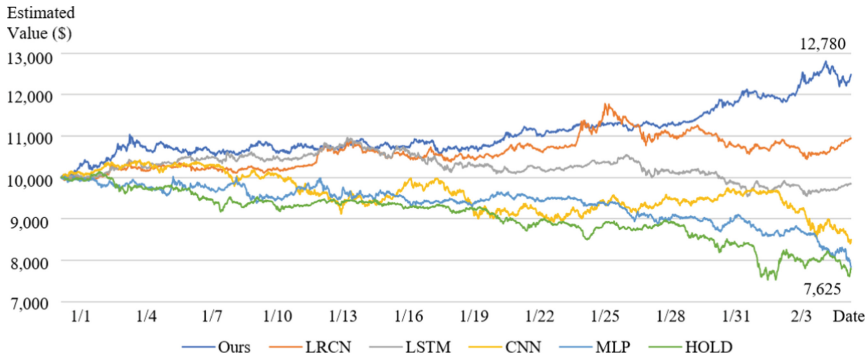

Fig. 6. The short-periodic profit comparisons with other deep models

Two financial measures, sharpe ratio and maximum drawdown, are used to evaluate the risk of strategies. Sharpe ratio are used to average return earned in excess of the risk-free rate per unit of volatility or total risk, defined as $Sharp_r = \frac{\widehat{r_p - r_f}}{\sigma_p}$ where $\widehat{r}_p$ is expected portfolio return, r_f is risk free rate, σ_p is the standard deviation for portfolio. Generally, higher sharpe ratios are known to guarantee higher returns for the same risk level. Maximum drawdown is defined as the maximum distance from a peak to portfolio, and it can be used as an measure of the rate of change of price.

Table 2 summarizes the change in assets for the initial $10,000 asset. Our agent outperformed among existing methods by achieving 27.86 of score. But in terms of risk, our agent has the volatility about price fluctuation. The Online Newton Step algorithm has the largest sharpe ratio and smallest maximum drawdown, indicating the algorithm is most stable.

To verify the performance of our agent against the latest cryptocurrency market and evaluate the effect of unsupervised learning, we compared our agent to other double Q-network based on deep learning methods as shown in Fig. 6. We conducted a back-test during 2018/01/01-2018/01/31. Long-term Recurrent Convolutional Network (LRCN) is the same architecture as the proposed agent but except pre-training algorithm. Remarkably, our proposed model achieved 1.27 score despite of the crash of bitcoin price while the investors who only bought bitcoin had a −23.75% of loss.

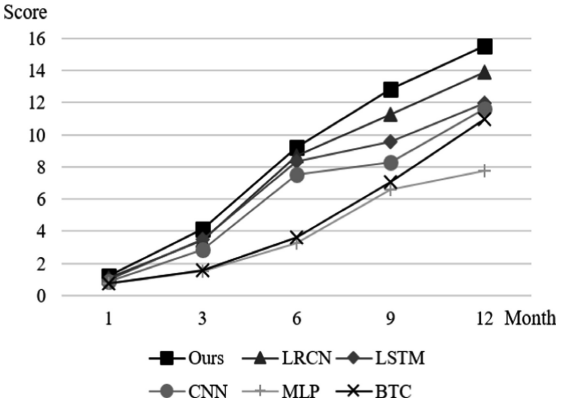

Fig. 7. The long-periodic profit comparisons with other deep models

Figure 7 represents the long-periodic profit of our agent, which is trained and tested up to 1 year. Since cryptocurrency market and price grows ten times between early in 2017 to 2018, investors who invested only in bitcoin also gained a 10 times of profit. Double Q-network with simple neural network recorded lower profit than bitcoin growth rate, which is similar behavior of individual investors suffering losses for no apparent reason.

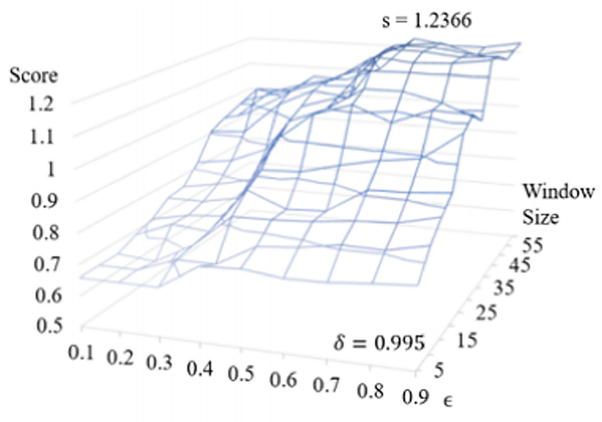

Fig. 8. The grid search to find optimal parameter for trading agent (Color figure online)

4.2 Parameter Optimization

Since the proposed trading agent combines double Q-network framework with DBM, various types of parameters can be adjusted. Typically, these parameters include the probabilistic factors that are used for exploration of Q-networks, as well as the hyperparameters for deep learning models such as the number of training iterations and the size of the convolution-pooling layer or LSTM layers. We used a traditional grid search to perform parameter optimization for trading agent (Fig. 9).

A grid search is simply an exhaustive search of a manually specified subset of the parameter space. The parameters to be optimized were set as the exploration rate, which is known to be responsible to escape the local minima in each episodes, and the window size of the market history in blocks, which is one of the major factors

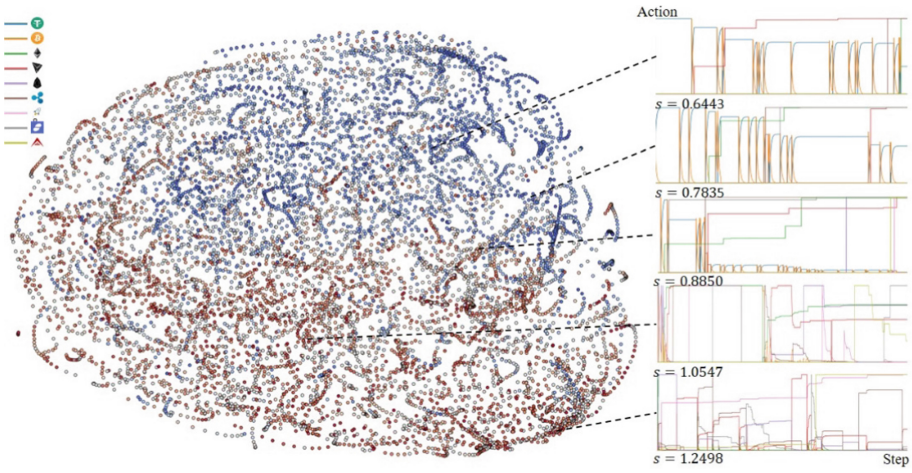

Fig. 9. The visualization of the states and actions

influencing the performance of sequence modeling. Figure 8 presents the performance of the double Q-network combined with DBM based on hyperparameters. After we fix the remaining parameters, we conducted a back-test for 2017/07/01-2017/07/31. As expected, the window contains the temporal information. The trading agent failed to map the appropriate action when the window is less than 15 min long.

4.3 Analysis of Action Vector

Figure 8 presents the entire dataset after the t-SNE algorithm is applied to the pre-processed states. t-SNE algorithm is a dimension reduction technique that is capable of retaining the local structures in data with while revealing important global structures [25]. Each of point is a single history block and the blue color represents the price drop in the block. The cluster of blue points and red points indicates the unsupervised training of encoding network was effective to reduce the search space.

On the right side of the figure, we visualized the action vector that occurred in the episode containing the state. From top to bottom, as learning progresses, the action decision of the trading agent is increasingly complex. The decision to buy and sell for the very top simply a few coins evolves into a complex decision that can not be interpreted at the very bottom.

5 Conclusion

We proposed a combination of double Q-network and DBM to generate and enhance the optimal Q-function in cryptocurrency trading. As the first component, The purpose of the encoding network and the double Q-network, in which two networks are connected in series, is to map the input state to appropriate action directly. The second main component is unsupervised learning module that pretrains the encoding network using modified DBM pre-training algorithm. We evaluated our agent through various experiments, including measurements of performance and risk, comparisons with existing algorithms, parameter optimization and visualization of the decisions. We achieved the highest profit among the existing models and deep learning models. Surprisingly, even when bitcoin price plummeted to −40%, the proposed agent achieved 20% of profit.

The main contribution of our research is to propose a combination of the two existing methods that can effectively model the history of financial or cryptocurrency markets. We introduce a framework of existing research to map the state into action corresponding the cryptocurrency market. We modified the existing DBM training algorithm and adapt to reduce the parameter space of encoding network. In addition, we collected and preprocessed 210.24 million of cryptocurrency trading records.

Since empirical results indicates that our agent is more risky than existing algorithms, Future work will include a stabilization process of volatile decision that our agent made by redesigning the Q-network. Secondly, we will enhance the performance of our agent by attaching generative model. Based on the generative network that generates and classifies virtual trade record, we will improve our Q-network more precisely.

Acknowledgements. This research was supported by Korea Electric Power Corporation. (Grant number:R18XA05).

References

1. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)
2. Agarwal, A., Hazan, E., Kale, S., Schapire, R.E.: Algorithms for portfolio management based on the Newton method. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 9–16. ACM (2006)
3. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**, 529 (2015)
4. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: AAAI, vol. 16, pp. 2094–2100 (2016)
5. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 609–616 (2009)
6. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**, 1527–1554 (2006)
7. Huang, W., Nakamori, Y., Wang, S.Y.: Forecasting stock market movement direction with support vector machine. Comput. Oper. Res. **32**, 2513–2522 (2005)
8. Schumaker, R.P., Chen, H.: Textual analysis of stock market prediction using breaking financial news: the AZFin text system. ACM Trans. Inf. Syst. **27**, 12 (2009)
9. Patel, J., Shah, S., Thakkar, P., Kotecha, K.: Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Syst. Appl. **42**, 259–268 (2015)
10. McNally, S.: Predicting the Price of Bitcoin using Machine Learning. National College of Ireland (2016)
11. Amjad, M., Shah, D.: Trading bitcoin and online time series prediction. In: NIPS 2016 Time Series Workshop, pp. 1–15 (2017)
12. Jiang, Z., Liang, J.: Cryptocurrency portfolio management with deep reinforcement learning. In: Intelligent Systems Conference, pp. 905–913 (2017)
13. Bell, T.: Bitcoin Trading Agents. University of Southampton (2016)
14. Żbikowski, K.: Application of machine learning algorithms for bitcoin automated trading. In: Ryżko, D., Gawrysiak, P., Kryszkiewicz, M., Rybiński, H. (eds.) Machine Intelligence and Big Data in Industry. SBD, vol. 19, pp. 161–168. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30315-4_14
15. Tesauro, G.: Extending Q-learning to general adaptive multi-agent systems. In: Advances in Neural Information Processing Systems, pp. 871–878 (2004)
16. Pinheiro, P.H., Collobert, R.: Recurrent convolutional neural networks for scene labeling. In: International Conference on Machine Learning, pp. 82–90 (2014)
17. Sainath, T.N., Vinyals, O., Senior, A., Sak, H.: Convolutional, long short-term memory, fully connected deep neural networks. In: Acoustics, Speech and Signal Processing, pp. 4580–4584 (2015)
18. Ren, Y., Wu, Y.: Convolutional deep belief networks for feature extraction of EEG signal. In: International Joint Conference on Neural Networks, pp. 2850–2853 (2014)
19. Lample, G., Chaplot, D.S.: Playing FPS games with deep reinforcement learning. In: AAAI, pp. 2140–2146 (2017)

20. Donahue, J., et al.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2625–2634 (2015)
21. Bu, S.-J., Cho, S.-B.: A hybrid system of deep learning and learning classifier system for database intrusion detection. In: Martínez de Pisón, F.J., Urraca, R., Quintián, H., Corchado, E. (eds.) HAIS 2017. LNCS (LNAI), vol. 10334, pp. 615–625. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59650-1_52
22. Cover, T.M.: Universal portfolios. In: The Kelly Capital Growth Investment Criterion: Theory and Practice, pp. 181–209 (2011)
23. Das, P., Banerjee, A.: Meta optimization and its applications to portfolio selection. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1163–1171 (2011)
24. Li, B., Zhao, P., Hoi, S.C., Gopalkrishnan, V.: PAMR: passive aggressive mean reversion strategy for portfolio selection. Mach. Learn. **87**, 221–258 (2012)
25. Maaten, L.V.D., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**, 2579–2606 (2008)