International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 03 | Mar 2020 www.irjet.net p-ISSN: 2395-0072

Ensembling Reinforcement Learning for Portfolio Management

Atharva Abhay Karkhanis[1], Jayesh Bapu Ahire[2], Ishana Vikram Shinde[3], Satyam Kumar[4]

[1,2,3,4]*Dept. of Computer Engineering, Sinhgad College of Engineering (SPPU), Pune, India*

Abstract - *The Stereoscopic Portfolio optimization Framework introduces the concept of bottom-up optimization via the utilization of machine learning ensembles applied to some market micro-structure element. But it doesn't work always as expected. The popular deep Q learning algorithm is known to be instability because of the Q-value's shake and overestimation action values under certain conditions. These issues tend to adversely affect their performance. Inspired by the breakthroughs in DQN and DRQN, we suggest a modification to the last layers to handle pseudo-continuous action spaces, as required for the portfolio management task. The current implementation, termed the Deep Soft Recurrent Q-Network (DSRQN) relies on a fixed, implicit policy. In this paper, we have described and developed ensemble deep reinforcement learning architecture which uses temporal ensemble to stabilize the training process by reducing the variance of target approximation error and the ensemble of target values reduces the overestimate and makes better performance by estimating more accurate Q-value. Our aggregate architecture leads to more accurate and optimized statistical results for this classical portfolio management and optimization problem.*

Key Words: Reinforcement Learning, Deep Learning, Artificial Intelligence, finance, Algorithmic Trading

1. INTRODUCTION

Reinforcement learning (RL) algorithms are very suitable for learning to regulate associate agent by material possession it moves with associate atmosphere. In recent years, deep neural networks (DNN) have been introduced into reinforcement learning, and that they have achieved an excellent success on the value function approximation. The first deep Q-network (DQN) algorithm which successfully combines a powerful nonlinear function approximation technique known as DNN together with the Q-learning algorithm was proposed by Mnih et al. In this paper, we are proposing experience replay mechanism. Following the DQN work, a variety of solutions are proposed to stabilize the algorithms. The deep Q-networks classes have achieved unprecedented success in challenging domains like Atari 2600 and few different games.

Although DQN algorithms are made in resolution several issues due to their powerful perform approximation ability and powerful generalization between similar state inputs, they're still poor in resolution some problems. Two reasons for this are as follows: (a) the randomness of the sampling is likely to lead to serious shock and (b) these systematic errors might cause instability, poor performance, and sometimes divergence of learning. In order to address these issues, the averaged target DQN (ADQN) algorithm is implemented to construct target values by combining target Q-networks continuously with a single learning network, and the Bootstrapped DQN algorithm is proposed to get more efficient exploration and better performance with the use of several Q-networks learning in parallel. Though these algorithms do scale back the overestimate, they are doing not assess the importance of the past learned networks. Besides, high variance in target values combined with the max operator still exists.

There are some ensemble algorithms solving this issue in reinforcement learning, however these existing algorithms don't seem to be compatible with non-linearly parameterized value functions.

In this paper, we propose the ensemble algorithm as a solution to this current downside. so as to boost learning speed and final performance, we combine multiple reinforcement learning algorithms in a single agent with several ensemble algorithms to determine the actions or action probabilities. In supervised learning, ensemble algorithms such as bagging, boosting, and mixtures of experts are often used for learning and combining multiple classifiers. But in Reinforcement Learning, ensemble algorithms are used for representing and learning the value function.

Based on an agent integrated with multiple reinforcement learning algorithms, multiple value functions are learned at the identical time. The ensembles mix the policies derived from the value functions in a final policy for the agent. The majority voting (MV), the rank voting (RV), the Boltzmann multiplication (BM), and the Boltzmann addition (BA) are used to combine RL algorithms. Whereas these ways are costly in deep reinforcement learning (DRL) algorithms, we combine different DRL algorithms that learn separate value functions and policies. Therefore, in our ensemble approaches we combine the different policies derived from the update targets learned by deep Q-networks, deep Sarsa networks, double deep Q-networks, and different DRL algorithms. As a consequence, this results in to reduce over-estimations, a lot stable learning method, and improved performance.

2. REINFORCEMENT LEARNING APPLIED TO FINANCE

There are a multitude of papers which have already used Reinforcement Learning in trading stock, portfolio management and portfolio optimization.

Moody et al. were the pioneers in applying the RL paradigm to the problem of stock trading and portfolio optimization. In our references they proposed the idea of Recurrent Reinforcement Learning (RRL) for Direct Reinforcement. RRL is an adaptive policy search algorithm that can learn an investment strategy on-line. Direct Reinforcement was a term coined to show algorithms that don't need to learn a value function in order to derive a policy. In other words, policy gradient algorithms in a Markov Decision Process framework are generally referred to as Direct Reinforcement. Moody et al. showed that a differential form of the Sharpe Ratio and Downside Deviation Ratio can be formulated to enable efficient on-line learning with Direct Reinforcement.

David W. Lu used the idea of Direct Reinforcement with an LSTM learning agent to learn how to trade in a Forex and commodity futures market. Du et al. used value function-based algorithm Q Learning for algorithmic trading. They use different forms of value functions like interval profit, Sharp Ratio and derivative Sharp Ratio to evaluate the performance of the approach.

Tang et al. used an actor-critic based portfolio investment method taking into consideration the risks involved in asset investment. The paper uses approximate dynamic programming to setup a Markov Decision model for the multi-time segment portfolio with transaction cost.

Jiang et al. in his one of the first papers, which provides a detailed Deep Reinforcement Learning framework which can be used in the task of Portfolio Management in a cryptocurrency market exchange. They used the concept of a Portfolio Vector Memory to help train the network, which they call the Ensemble of Identical Independent Evaluators (EIIE). They take into consideration market risks and the transaction costs associated with buying and selling assets in a stock exchange.

3. RECURRENT REINFORCEMENT LEARNING

In this approach the decision-making of investment developed by J. Moody and M. Saffell is considered as a stochastic problem and strategies are directly identified. They have an adaptive algorithm for discovering investment policies, called Recurrent Reinforcement Learning (RRL). Dynamic programming and enhancement algorithms like TD-learning and Q-learning are different from direct enforcement approaches, which try to estimate a value function for the control problem. This facilitates the representation of the problem through the RRL Direct Reinforcement Framework and prevents Bellman's dimensionality and offers convincing efficiency benefits. They demonstrate how direct reinforcement can be used to optimize risk-adjusted returns on investment, taking account costs. They use real financial information intra-daily and find that their RRL-based approach produces better trade strategies than Q-learning systems.

Steve Y. Yang and Saud Almahdi are also taking another approach to solving optimal asset allocation problems and a number of trading decision schemes based on methods of enhanced learning. They establish an optimum allocation of variable weights in line with a consistent downside risk measure E(MDD). The Calmar Ratio, specifies their method using the RRL method for both buying and selling signals and asset allocation weights, with a consistent risk-adjusted performance goal. The expected maximum risk downward-focused objective function is shown through the most frequently traded exchange fund's portfolio as a higher return than previously proposed RRL functions (i.e. Sharpe or Sterling Ratio), and variable weight portfolios in various scenarios of transactions cost equal portfolios. NOTE: The Calmar ratio represents a comparison between the average annual compound rate of return and the maximum risk attraction for commodity trading consultants and hedge funds. The smallest the Calmar ratio, the worse the investment was carried over the specified period on a risk-based basis, the higher the Calmar ratio, the better it was.

Deep learning (DL) combined with reinforcement learning in the work of Deng et al., introduced a recurrent deep neural network (NN) for real-time financial signal representation and trading. DL automatically detects the dynamic market conditions for informative learning, and the RL module then interacts with deep representations and decides to accumulate ultimate income in an unknown environment. The system of learning is performed in a complex NN with a highly recurring structure. They thus propose a time-based task-back-cutting to tackle the problem of deep training slowdown. The strength of the neural system is confirmed on both the stock and commodity markets under wide-ranging test conditions.

The RRL approach is clearly different from dynamic programs and strengthening algorithms, such as TD-learning and Q-learning, which try to approximate calculate a value function for the control problem. With the RRL framework, simple, elegant problem representation is created, the dimensionality of Bellman is avoided and efficiency offers compelling advantages: Compared to Q-learning when exposed to rowdy data sets, RRL has a more stable performance. Q-learning algorithm is more sensitive to selecting the value (maybe) because of the recursive dynamic optimization property whereas RRL algorithms can choose the objective function and save time.

4. ENSEMBLE METHODS FOR DEEP REINFORCEMENT LEARNING

As DQN classes use DNNs to approximate the value function, it has strong generalization ability between similar state inputs. The generalization can cause divergence in the case of repeated bootstrapped temporal difference updates. So we can solve this issue by integrating different versions of the target network.

In contrast to a single classifier, ensemble algorithms in a system have been shown to be more effective. They can lead to a higher accuracy. Bagging, boosting, and Ada Boosting are methods to train multiple classifiers. But in RL, ensemble algorithms are used for representing and learning the value function. They are combined by major voting, Rank Voting, Boltzmann Multiplication, mixture model, and other ensemble methods. If the errors of the single classifiers are not strongly correlated, this can significantly improve the classification accuracy.

5. THE ENSEMBLE NETWORK ARCHITECTURE

The temporal and target values ensemble algorithm (TEDQN) is an integrated architecture of the value-based DRL algorithms. As shown in previous sections, the ensemble network architecture has two parts to avoid divergence and improve performance.

The architecture of our ensemble algorithm is shown in Figure 1; these two parts are combined together by evaluated network.

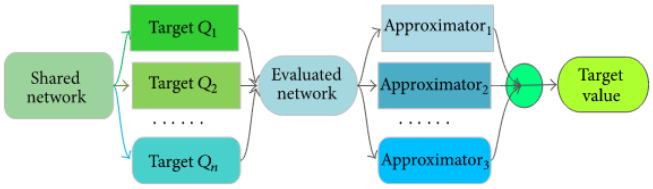

Fig -1: The architecture of the ensemble algorithm

The temporal ensemble stabilizes the training process by reducing the variance of target approximation error [10]. Besides, the ensemble of target values reduces the overestimate and makes better performance by estimating more accurate Q-value. The temporal and target values ensemble algorithm are given by Algorithm 1.

(1) Initialize action-value network Q with random weights θ
(2) Initialize the target neural network buffer $(Q_i)_{i=1}^{L}$
(3) For episode 1, M do
(4) For $t = 1, T$ do
(5) With probability ε select a random action a_t, otherwise $a_t = \text{argmax}_a Q(s_t, a; \theta)$
(6) Execute action a_t in environment and observe reward r_t and next state s_{t+1}, and store transition (s_t, a_t, r_t, s_{t+1}) in D
(7) Sample random *minibatch* of transition (s_t, a_t, r_t, s_{t+1}) from D
(8) set $w_i = \lambda^{i-1} / \sum_{i=1}^{N} \lambda^{i-1}$
(9) Ensemble Q-learner $\widetilde{Q}(s, a; \theta) = \sum_{i=1}^{N} w_i Q_i(s, a; \theta_i)$
(10) set $y_i^{\text{DQN}} = r_t + \gamma \max_a \widetilde{Q}(s_{t+1}, a; \theta_t^-)$
(11) set $y_i^{\text{Sarsa}} = r_t + \gamma \widetilde{Q}(s_{t+1}, a_{t+1}; \theta_t^-)$
(12) set $y_i^{\text{DDQN}} = r_t + \gamma \widetilde{Q}(s_{t+1}, \text{argmax}_{a'} \widetilde{Q}(s_{t+1}, a_{t+1}; \theta_t); \theta_t^-)$
(13) Set $y_i = \{r_j$, if episode terminates at step $j+1$; $\sum_{i=1}^{k} \beta_i y_t^i$, otherwise$\}$
(14) $\theta_i = \underset{\theta}{\text{argmin}}\, E\left[\left(y_{(s,a)}^i - Q(s, a; \theta)\right)^2\right]$
(15) Every C steps reset $\widetilde{Q} = Q$
(16) End for
(17) End for

Fig -2: The temporal and target values ensemble algorithm.

As the ensemble network architecture shares the same input-output interface with standard Q-networks and target networks, we can recycle all learning algorithms with Q-networks to train the ensemble architecture.

6. CONCLUSIONS

We introduced a new learning architecture, making temporal extension and the ensemble of target values for deep learning algorithms, while sharing a common learning module. The new ensemble architecture, in combination with some algorithmic improvements, leads to dramatic improvements over existing approaches for deep RL in the challenging classical control issues. In practice, this ensemble architecture can be very convenient to integrate the RL methods based on the approximate value function.

Although the ensemble algorithms are superior to a single reinforcement learning algorithm, it is noted that the computational complexity is higher. The experiments also show that the temporal ensemble makes the training process more stable, and the ensemble of a variety of algorithms makes the estimation of the -value more accurate. The combination of the two ways enables the training to achieve a stable convergence. This is due to the fact that ensembles improve independent algorithms most if the algorithms predictions are less correlated. So that the output of the -network based on the choice of action can achieve balance between exploration and exploitation.

In fact, the independence of the ensemble algorithms and their elements is very important on the performance for ensemble algorithms. In further works, we want to analyze the role of each algorithm and each -network in different stages, so as to further enhance the performance of the ensemble algorithm.

REFERENCES

[1] S. Mozer and M. Hasselmo, "Reinforcement learning: an introduction," IEEE Transactions on Neural Networks and Learning Systems, vol. 16, no. 1, pp. 285-286, 2005. View at Publisher · View at Google Scholar

[2] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: a survey," Journal of Artificial Intelligence Research, vol. 4, pp. 237–285, 1996. View at Google Scholar · View at ScopusR. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[3] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing Atari with deep reinforcement learning [EB/OL]," https://arxiv.org/abs/1312.5602.

[4] M. A. Wiering and H. van Hasselt, "Ensemble algorithms in reinforcement learning," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 38, no. 4, pp. 930–936, 2008. View at Publisher · View at Google Scholar · View at Scopus

[5] S. Whiteson and P. Stone, "Evolutionary function approximation for reinforcement learning," Journal of Machine Learning Research (JMLR), vol. 7, pp. 877–917, 2006. View at Google Scholar · View at MathSciNet

[6] P. Preux, S. Girgin, and M. Loth, "Feature discovery in approximate dynamic programming," in Proceedings of the 2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, ADPRL 2009, pp. 109–116, April 2009. View at Publisher · View at Google Scholar · View at Scopus

[7] T. Degris, P. M. Pilarski, and R. S. Sutton, "Model-Free reinforcement learning with continuous action in practice," in Proceedings of the 2012 American Control Conference, ACC 2012, pp. 2177–2182, June 2012. View at Scopus

[8] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015. View at Publisher · View at Google Scholar · View at Scopus

[9] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-Learning," in Proceedings of the 30th AAAI Conference on Artificial Intelligence, AAAI 2016, pp. 2094–2100, February 2016. View at Scopus

[10] O. Anschel, N. Baram, N. Shimkin et al., "Averaged-DQN: Variance Reduction and Stabilization for Deep Reinforcement Learning [EB/OL]," https://arxiv.org/abs/1611.01929.

[11] I. Osband, C. Blundell, A. Pritzel et al., "Deep Exploration via Bootstrapped DQN [EB/OL]," https://arxiv.org/abs/1602.04621.

[12] S. Faußer and F. Schwenker, "Ensemble Methods for Reinforcement Learning with Function Approximation," in Multiple Classifier Systems, pp. 56–65, Springer, Berlin, Germany, 2011. View at Google Scholar

[13] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: a review," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 4–37, 2000. View at Publisher · View at Google Scholar · View at Scopus

[14] T. Schaul, J. Quan, I. Antonoglou et al., "Prioritized Experience Replay [EB/OL]," https://arxiv.org/abs/1511.05952.

[15] I. Zamora, N. G. Lopez, V. M. Vilches et al., "Extending the OpenAI Gym for robotics: a toolkit for reinforcement learning using ROS and Gazebo [EB/OL]," https://arxiv.org/abs/1608.05742.

[16] D. Ernst, P. Geurts, and L. Wehenkel, "Tree-based batch mode reinforcement learning," Journal of Machine Learning Research (JMLR), vol. 6, no. 2, pp. 503–556, 2005. View at Google Scholar · View at MathSciNet

[17] Coleman, L. (2017). "What is the Stereoscopic Portfolio Optimization Framework: Applying Machine Learning Ensembles to Market Microstructure to Achieve Portfolio Optimization" QuantInsti Quantitative Ltd.,

https://ijarcce.com/papers/reinforcing-portfolio-management-through-ensemble-learning/