Large-Scale Multi-agent Reinforcement Learning Based on Weighted Mean Field

Baofu Fang[1(✉)], Bin Wu[1], Zaijun Wang[2], and Hao Wang[1]

[1] Hefei University of Technology, Hefei, Anhui, China
fangbf@hfut.edu.cn
[2] Civil Aviation Flight University of China, Guanghan, Sichuan, China

Abstract. Deep reinforcement learning is an emerging approach to solve the decision making of multi-agent systems in recent years, and currently has achieved good results in small-scale decision problems. However, when the number of agents increases, the dynamic of the other agent strategies and the proportional enlargement of information between the agent lead to "non-stationarity", "dimensional catastrophe" and many other problems. In order to solve Multi-Agent Deep Deterministic Policy Gradient (MADDPG) are difficult to converge when the size of multi-agent systems exceeds a certain number, a deep reinforcement learning collaboration algorithm for multi-agent systems based on weighted mean field is proposed. The mean field is used to reconstruct the dynamic decision action of other agent involved in the decision making, while assigning different weights to each agent action based on the set of relevant attributes, transforming the joint action of the agent into the mean action of the other agent formed through the weighted mean field, and serving as an update function of the actor network and state function in the multi-agent deep deterministic policy gradient algorithm parameters to simplify the scale of interaction. In this paper, the effectiveness of the algorithm is validated by Battle game scenarios from convergence, win rates at different scenario sizes, win rates of different algorithms, and other game performance.

Keywords: Multi-agent system · Reinforcement learning · Weighted mean field

1 Introduction

A multi-agent system is a collection of multiple autonomous, interacting agent, which is an important branch of distributed artificial intelligence. Such problems are significantly more difficult than those of single agent [1].

Reinforcement learning (RL) search for the optimal policy in a Markov process by obtaining as many rewards as possible. Articles [2] extends the Tit-for-tat policy with deep Q learning to maintain two agent cooperation even in the face of a prisoner's dilemma in order to achieve an optimal policy. Articles [3] investigates the performance of deep reinforcement learning. It is ultimately shown that deep reinforcement learning can achieve high performance. Articles [4] investigates the problem of cooperation in

F. Sun et al. (Eds.): ICCSIP 2020, CCIS 1397, pp. 309–316, 2021.
https://doi.org/10.1007/978-981-16-2336-3_28

simple scenarios in a discrete structure for each agent that learns independently using the Q learning algorithm. The problem sizes studied above are typically a few agents, however, in real life, a large number of agents are indeed required to interact strategically with each other, such as fleet dispatch management problems [5], online role-playing games [6] or online advertising bidding auctions [7]. Existing multi-agent reinforcement learning solutions are effective, but they can only solve the problems of small-sized agent. In an environment where "an infinite number of agents are approached and imprecise probabilistic models are assumed to operate", existing methods for deep reinforcement learning are not realistic [8]. Articles [9] designed an actor-critic algorithm to train a very large-scale taxi dispatching problem regionally. Articles [10] uses a centralized approach to learn distributed execution, learning to consider the Q-value function simultaneously. The team's Q-value functions are decomposed into sums of their respective Q-value functions through a value decomposition network. Articles [11] uses a mixing network to combine local values, but this network is very difficult to train.

Learning between agents promotes each other: the learning of a single agent is based on the state of cluster agents [12].This paper proposes a multi-agent reinforcement learning algorithm based on a weighted mean field, using the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [13] as the framework in order to highlight the importance of different agents, the algorithm Based on related attribute sets, different weights are assigned to each agent action, and the weighted mean field is used to simplify the interactive information, which improves the performance of multi-agent reinforcement learning algorithms under large-scale multi-agents.

2 Weighted Mean Field Reinforcement Learning

2.1 Mean Field Theory

The mean field theory is a method of collectively processing the effects of the environment on individuals, and replacing the accumulation of multiple effects with mean effects. The strategic decision-making relationship between individuals and the whole and the relationship between joint strategy $\pi(s, a)$ and single agent strategy $\pi(s, a_i)$ in a multi-agent system.

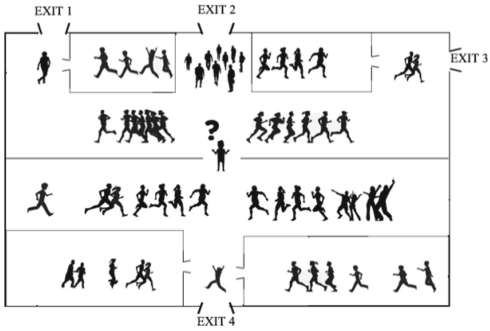


Fig. 1. Evacuation diagram

For example, many people do not know all the information about the surrounding environment, such as exit locations and routes, when people need to be evacuated urgently in Fig. 1. Each person does not need to observe the evacuation status of all people, but only needs to observe the status of the people around him. Based on the evacuation of the surrounding people, he can follow the people to different exits.

Wang [15] proposed the concept of mean field reinforcement learning (MFRL), but the use of mean field is simply to mean the action states of the surrounding agents of each agent and then import the critic network. But according to the multi-agent system consistency idea, the state information of different agents has different reference value to the central agent. Since exit 4 has the most people and may have the greatest impact on the evacuees, evacuees are directed to exit 4, in fact exit 2 is the best select. At this time, the surrounding people have different weights due to distance and other attributes. Those who are closer to the evacuees have a higher weight, and the evacuees are guided to choose the exit 2 which may be smaller but closer. At this time, the different distance or intimacy between each person forms a weight, and the relationship with the evacuees and the closest person. The weight distribution of the mean field is carried out through the influence of related factors, when applied to multi-agent reinforcement learning, it can better reflect the cluster effect of the mean field idea.

2.2 Weighted Mean Field

The size of the joint action increases exponentially with the size of the agent. All agents take their own strategy actions dynamically, which makes the reward function $Q(s, a)$ unable to converge effectively. In order to solve this problem, the reward function is reconstructed, and the local joint action using weighted mean field is used as the parameter added value function.

$$Q_{\mu}^{i}(s, a) = \frac{1}{\omega(x_i)} \sum\nolimits_{j \in N(i)} \omega\left(x_i^j\right) Q^i\left(s, a_i, a_j\right) \tag{1}$$

Where $N(i)$ is the collection of agents around the agent i, $N^i = |N(i)|$ represents the number of agents around the agent i, $\omega(x_i) = \sum\limits_{j \in N(i)} \omega\left(x_i^j\right)$, and $\omega\left(x_i^j\right)$ is the weight coefficient function of the corresponding central agent i to the agent j. The weight parameters $x_i^j = f(Parameter1, Parameter2, \cdots)$ are determined by the relevant agent attribute set, such as position, importance, intimacy, etc., which can be flexibly defined according to the scene and reward.

$$\begin{aligned} Q_{\mu}^{i}(s, a) &= \frac{1}{\omega(x_i)} \sum_j \omega\left(x_i^j\right) Q_{\mu}^{i}\left(s, a_i, a_j\right) \\ &= \frac{1}{\omega(x_i)} \sum\nolimits_j \omega\left(x_i^j\right) \left[Q_{\mu}^{i}(s, a_i, \overline{a}_i) + \nabla_{\overline{a}_i} Q_{\mu}^{i}(s, a_i, \overline{a}_i) * \delta\alpha_{i,j} \right. \\ &+ \left. \frac{1}{2} \delta\alpha_{i,j} * \nabla_{\overline{a}_{i,j}}^2 Q_{\mu}^{i} s, a_i, \overline{a}_i * \delta\alpha_{i,j} \right] \\ &= Q_{\mu}^{i}(s, a_i, \overline{a}_i) + \nabla_{\overline{a}\overline{a}_i} Q_{\mu}^{i}(s, a_i, \overline{a}_i) * \left[\frac{1}{\omega(x_i)} \sum\nolimits_j \delta\omega\left(x_i^j\right) \alpha_{i,j} \right] \end{aligned}$$

$$+ \frac{1}{2\omega(x_i)} \sum_j \left[\delta\omega\left(x_i^j\right)\alpha_{i,j} * \nabla^2_{\tilde{a}_{i,j}} Q^i_\mu s, a_i, \overline{a}_i * \delta\alpha_{i,j} \right]$$
$$\approx Q^i_\mu\left(s, a_i, a_j\right) \tag{2}$$

For the $Q^i_\mu\left(s, a_i, a_j\right)$ according to Eq. (1), we approximate it by mean field theory, and the agent i action adopts a_i. For example, Eq. (2) is used to calculate the mean weighted actions $\overline{a}_i$ of the agents around the agent i, and the actions a_j of the neighboring agent j are converted into $\overline{a}_i$ and a margin $\delta\alpha_{i,j}$.

$$\begin{cases} \overline{a}_i = \frac{1}{\omega(x_i)} \sum_j \omega\left(x_i^j\right) a_j \\ a_j = \overline{a}_i + \delta\alpha_{i,j} \end{cases}, \quad j \in N(i) \tag{3}$$

2.3 Weighted Mean Field Multi-agent Deep Deterministic Policy Gradient (WMPG)

The decision-making behavior between multiple agents is transformed into the weighted behavior of central agent and adjacent agent by mean field theory, so as to modify the reward function and state value function of Multi-Agent Reinforcement Learning (MARL). By means of mean field theory, for example, the original critic network of multi-agent depth deterministic strategy gradient algorithm is simplified to the weighted mean of adjacent agent actions, which simplifies the calculation of other agent information, so that it can be applied to large-scale agent scenarios, and introduces weight information. The algorithm framework is shown in Fig. 2.

When calculating the loss function L and the strategy gradient, where $Q^i_\varphi(s, a_1, \cdots, a_n)$ is approximately transformed into $Q^i_\varphi(s, a_i, \overline{a}_i)$. The calculation of $\overline{a}_i$ is as shown in Eq. (3), which simplifies the iteration of value function and state value function in critic network.

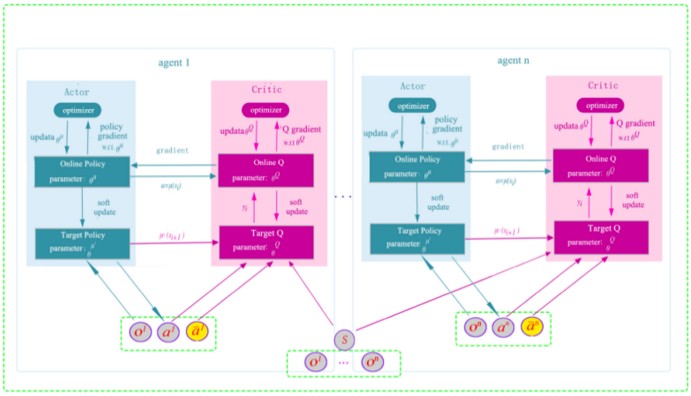


Fig. 2. Weighted Mean Field Multi-Agent Deep Deterministic Policy Gradient, WMPG

The update mode of the critic network is changed to, φ is the weight parameter of the critic network

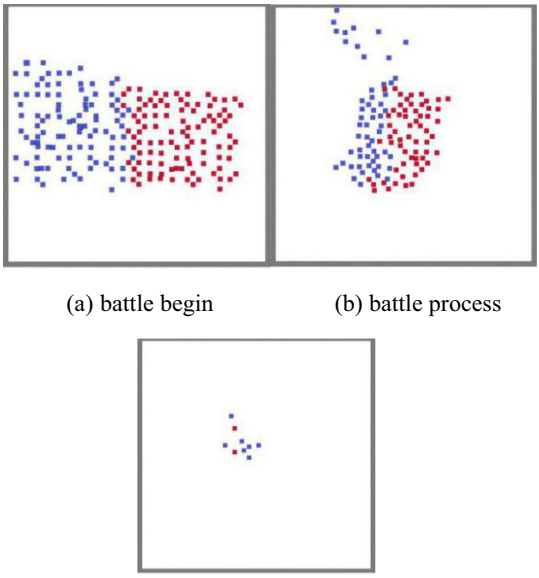
$$L(\varphi_i) = \frac{1}{S}\sum_j \left(y_j - Q^j_{\varphi_j}(s, a_j, \overline{a}_j)\right)^2 \tag{4}$$

Where S is the number of training samples θ is actor network parameter, the gradient equation of actor network strategy can be written.

$$\nabla_{\theta_j} J(\theta_i) \approx \frac{1}{S}\sum_j \nabla\theta_j \log \pi^j_{\theta_j}(s) Q^j_{\varphi_j}(s, a_j, \overline{a}_j)\Big|_{a_j=\pi_{\theta_j}(s)} \tag{5}$$

3 Experiment Results and Analysis

In order to verify the effectiveness of the algorithm, the performance of weighted mean field in large-scale Multi-Agent Reinforcement Learning is explored by using the Battle environment. Battle is a mixed combat multi-agent combat scenario of the open-source MAgent [16] framework. In Fig. 3, the blue side is the algorithm of this paper, and the red side is MFAC. It can be seen from the course of the battle that when an agent chooses to encircle, the nearby agents will be affected by the decision of the agent, thus forming a trend of encircling the red side by the blue side.



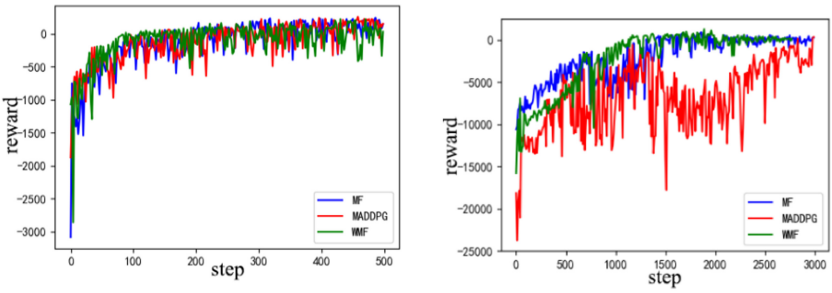
(a) battle begin (b) battle process

(c) battle end

Fig. 3. Confrontation graph between WMPG and MFAC under Battle (Color figure online)

3.1 Convergence Experiment

In order to show the training performance of the algorithm under different scale agents, this paper conducts experiments on the algorithm convergence of MADDPG which only uses mean field and weighted mean field. It can be seen that in a large scale, the convergence speed of this algorithm is obviously better than the other two. The multi-agent depth deterministic strategy gradient algorithm has been difficult to converge at the scale of 400. The weighted mean field algorithm proposed in this paper can reduce the difficulty of convergence to a certain extent (Fig. 4).



(a)64vs64 (b)400vs400

Fig. 4. Analysis of algorithm convergence under different agent scale

3.2 Cross-Contrast

In order to prevent contingency, in the Battle environment, set the size of both agents to 200vs200.WMPG performed 200 rounds of cross-comparison with traditional reinforcement learning algorithm actor-critic AC, multi-agent deep reinforcement learning algorithm MADDPG, and mean field algorithm MFQ and MFAC [15] algorithm to verify the performance of this algorithm. Table 1 and Table 2 show the experimental results of the win rate and total return in the comparative experiment. It can be seen from the results in Table 1 that the WMPG algorithm is significantly better than other algorithms when experimenting with different types of algorithms, and from Table 2, it can be seen that the total return can also be higher, which also reflects the effectiveness of our algorithm. The poor performance of the multi-agent deep deterministic strategy gradient algorithm and the actor-critic algorithm also reflects the current traditional deep reinforcement learning algorithms mentioned in this article facing the problems of large-scale multi-agent systems. The research value of this article.

Table 1. The win rate of different algorithms

Algorithm	AC	MADDPG	MFQ	MFAC	WMPG
AC VS	–	0.34	0.15	0.29	0.0
MADDPG VS	0.66	–	0.44	0.38	0.17
MFQ VS	0.85	0.56	–	0.0	0.0
MFAC VS	0.71	0.62	1.0	–	0.285
WMPG VS	**1.0**	**0.830**	**1.0**	**0.715**	–

Table 2. The total reward of different algorithms

Algorithm	AC	MADDPG	MFQ	MFAC	WMPG
AC VS	–	13773	4645	3716	5712
MADDPG VS	18920	–	17053	17560	12845
MFQ VS	13040	14729	–	7959	12617
MFAC VS	22510	17471	16894	–	19759
WMPG VS	**21918**	**22265**	**22575**	**19758**	–

4 Conclusion

Based on the idea of mean field, this paper proposes a deep reinforcement learning algorithm of mean field multi-agent with weight information. Aiming at the problem that when the number of larger multi-agent systems exceeds a certain scale, the complex interactive information between agents will make the original reinforcement learning algorithm difficult to converge, and the weighted mean field is used to simplify the interactive information and transform the multi-agent information It is two-body information, used to simulate the interaction in the multi-agent system. The algorithm in this paper reduces the instability and difficulty of convergence of the combat environment in the open-source MAgent framework. In addition to giving certain proofs in theory, the experimental results also verify the rationality and effectiveness of this algorithm. In the follow-up, we need to consider the combination of large-scale multi-agent and mean field in heterogeneous scenarios.

References

1. Hernandez-Leal, P., Kartal, B., Taylor, M.E.: A survey and critique of multiagent deep reinforcement learning. Autonom. Agents Multi Agent Syst. **33**, 750–797 (2019)

2. Peysakhovich, A., Lerer, A.: Maintaining cooperation in complex social dilemmas using deep reinforcement learning. arXiv: Artificial Intelligence (2018)
3. Raghu, M., Irpan, A., Andreas, J., et al.: Can deep reinforcement learning solve Erdos-Selfridge-Spencer games? arXiv: Artificial Intelligence (2017)
4. Foerster, J.N., Nardelli, N., Farquhar, G., Afouras, T., Torr, P.H.S., Kohli, P., Whiteson, S.: Stabilising experience replay for deep multi-agent reinforcement learning. In: International Conference on Machine Learning (2017).
5. Lin, K., Zhao, R., Xu, Z., et al.: Efficient large-scale fleet management via multi-agent deep reinforcement learning. In: Knowledge Discovery And Data Mining, pp. 1774–1783 (2018)
6. Peng, P., Wen, Y., Yang, Y., et al.: Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games (2017)
7. Jin, J., Song, C., Li, H., et al.: Real-time bidding with multi-agent reinforcement learning in display advertising (2018).
8. Hernandez-Leal, P., Kartal, B., Taylor, M.E.: Is multiagent deep reinforcement learning the answer or the question? A brief survey. Learning **21**, 22 (2018)
9. Thien, N., Kumar, A., Lau, H.: Policy gradient with value function approximation for collective multiagent planning (2017)
10. Sunehag, P., Lever, G., Gruslys, A., et al.: Value-decomposition networks for cooperative multi-agent learning (2017)
11. Rashid, T., Samvelyan, M., De Witt, C.S., et al.: Monotonic value function factorisation for deep multi-agent reinforcement learning (2020)
12. Sharma, M.K., Zappone, A., Assaad, M., et al.: Distributed power control for large energy harvesting networks: a multi-agent deep reinforcement learning approach. IEEE Trans. Cogn. Commun. Network. **5**(4), 1140–1154 (2019)
13. Lowe, R., Wu, Y., Tamar, A., et al.: Multi-agent actor-critic for mixed cooperative-competitive environments. In: Neural Information Processing Systems, pp. 6379–6390 (2017)
14. Zhang, K., Yang, Z., Basar, T., et al.: Multi-agent reinforcement learning: a selective overview of theories and algorithms. arXiv: Learning (2019)
15. Yang, Y., Luo, R., Li, M., et al.: Mean field multi-agent reinforcement learning. In: International Conference on Machine Learning, pp. 5567–5576 (2018)
16. Zheng, L., Yang, J., Cai, H., et al.: MAgent: a many-agent reinforcement learning platform for artificial collective intelligence (2017)