Multi-Agent Pattern Formation with Deep Reinforcement Learning* (Student Abstract)

Elhadji Amadou Oury Diallo,[1] **Toshiharu Sugawara**[1]
[1]Department of Computer Science and Communications Engineering, Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan
diallo.oury@fuji.waseda.jp, sugawara@waseda.jp

Abstract

We propose a decentralized multi-agent deep reinforcement learning architecture to investigate pattern formation under the local information provided by the agents' sensors. It consists of tasking a large number of homogeneous agents to move to a set of specified goal locations, addressing both the assignment and trajectory planning sub-problems concurrently. We then show that agents trained on random patterns can organize themselves into very complex shapes.

1 Introduction

Let us assume that an instructor needs her n students in the play area to form a 2D shape/pattern such as a circle so that, for example, they can play a game. The instructor may draw a hover on the ground as a rule or even give every student a particular position to move to. Now, imagine a scenario where the instructor does not give such help. Indeed, even without such help, the kids may, in any case, have the option to frame an adequately decent estimation of a circle if every one of them moves depending on the movement of others by directly observing their neighborhood region. If successful, this method can be called a distributed solution to the circle formation problem for children (Suzuki and Yamashita 1999).

We utilized a methodology based on previous example to control a group of various agents. The principal idea is to give every agent a chance to execute a straightforward estimation of its states and plan its actions depending on the actions and states of the remaining agents so that the agents as a team will cooperatively accomplish the given objective. Up to now, the vast majority of the current methods have been focused on either centralized methods in which a leader agent estimates the actions of all agents or a decentralized method (Lowe et al. 2017) in which the agents have a full view and knowledge of their environment and its dynamics.

In this paper, we investigate how a large-scale system of independent learning agents can achieve an acceptable collective pattern formation. For this purpose, we propose (1) an end-to-end decentralized learning architecture in which agents do not explicitly communicate; (2) use a centralized replay memory to share knowledge; (3) and use a centralized target network to take into account the dynamics of others. The goal is to control the overall shape of a robot team by using only the local information provided by agents' sensors. Interestingly, the positions or goals of the individual agents in the group are not explicitly controlled. Agents should concurrently and independently learn to locate their goal position and consequently plan a smooth trajectory towards it.

*Partly supported by JSPS KAKENHI Grant No. 17KT0044.

2 Methods

A decentralized partially observable Markov decision process (dec-POMDP) (Bernstein et al. 2002) is defined as a tuple $\langle \mathcal{D}, \mathcal{S}, \boldsymbol{\mathcal{A}}, \mathcal{T}, \mathcal{R}, \boldsymbol{\Omega}, \mathcal{O}, h, \mathcal{I} \rangle$, where $\mathcal{D} = \{1, ..., n\}$ is a set of n agents. $\mathcal{S}$ is a finite set of states s in which the environment can be. $\boldsymbol{\mathcal{A}}$ is the finite set of joint actions of agents, $\boldsymbol{\mathcal{A}} = \mathcal{A}^1 \times ... \times \mathcal{A}^n$. $\mathcal{T}$ is a probabilistic transition function and $\mathcal{R}$ is the immediate reward function. $\boldsymbol{\Omega}$ is the finite set of joint observations with $\boldsymbol{\Omega} = \mathcal{O}^1 \times ... \times \mathcal{O}^n$. $\mathcal{O}$ is the observation probability function, h is the horizon of the problem, and $\mathcal{I}$ is the initial state distribution at time $t = 0$. The environment transits from s_t to s_{t+1} with a probability $p(s_{t+1}|s_t, \mathbf{a}_t) \in \mathcal{T}$ when all actions $\mathbf{a}_t = \langle a_t^1, \dots a_t^n \rangle$ are executed. Then agent i receives a reward $r_t^i = \mathcal{R}(s_{t+1}|s_t^i, a_t^i)$. The observation o can be approximated by a *kth order history* approach which uses the last k observations and actions. In our case, $o = \langle o_t, o_{t-1}, o_{t-2}, \mathbf{a_{t-1}} \rangle$ (Diallo and Sugawara 2018). This approach can manage any latent state information compared to using directly the latest observation as the observation.

We propose a decentralized system with a centralized replay memory (Fig. 1) to tackle the problem of multi-agent pattern formation. Each agent has a limited visual field of depth k (shape $= [2k + 1, 2k + 1]$) – that is, an agent can observe teammates, obstacles, and walls within its neighborhood region. The agents are homogeneous and anonymous, that is they cannot be distinguished by their appearance. Besides, our framework works in a completely distributed mode and agents have no preference for goal destinations.

Each agent of the team has its own main network and a shared target network. It stores and updates its representation of the environment by randomly sampling from the re-

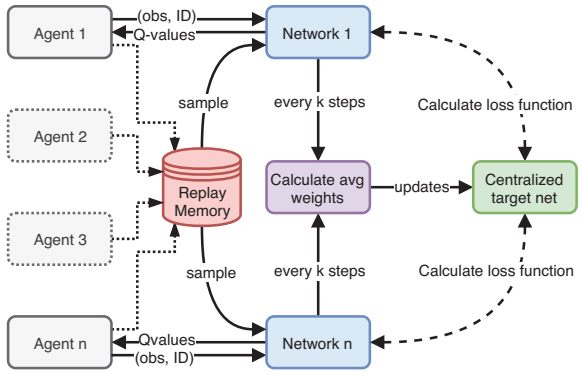

Figure 1: Distributed model with centralized replay memory.

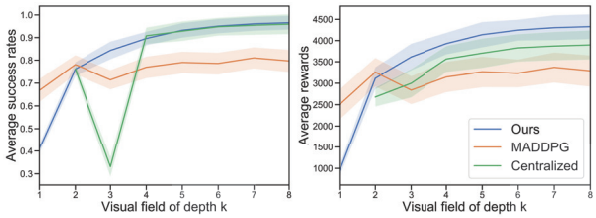

Figure 2: Left: average success rate. Right: average reward

play memory. As a consequence, each agent independently computes its patrolling plan by taking advantage of other agents knowledge without exchanging coordination messages. This approach can accelerate the convergence rate if combined with a good exploration strategy. In addition, the centralized target network is updated by the average weights of individual agents' networks. By doing so, we also ensure that an agent can react to the previous actions and rewards of others by taking into account their dynamics. As a result, this somehow provides some sort of communicate to cope with the local view.

3 Results

We use a centralized system as a baseline, in which a team's strategy is computed by a central agent and subsequently communicated to all teammates. We also compare our method against a discrete action space version of MADDPG (Lowe et al. 2017). To improve the effectiveness of our method and its ability to generalize on completely random and unseen environment, the agents were trained on a large 2-d grid graph (150×150) with utmost 500 randomly generated landmarks at the beginning of each episode.

Figure 2 shows the average reward and completion rate of different visual field depths. With a smaller view range, MADDPG achieved the highest reward, followed by the centralized network while our method struggles to learn anything useful. As the view range increases ($k \geq 2$), the proposed method steadily improves while the performance of MADDPG was less stable. The instability of MADDPG is due to the exchange of observation during training. In sum, with a large enough view range, agents can solve their tasks by using either our proposed method or a centralized system

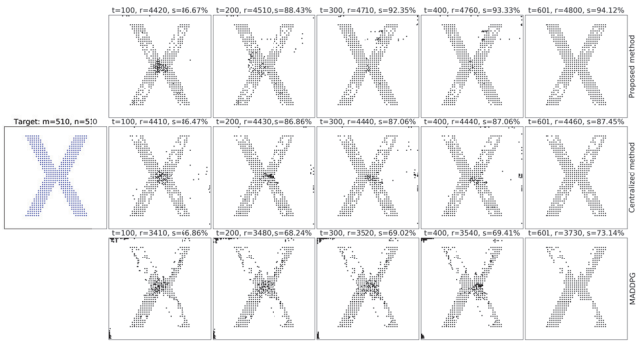

Figure 3: 550 agents try to organize themselves into an X shape in an 150×150 grid-graph.

with a lead agent. However, the gains are very marginal with a very large k compared to the computational cost.

While our method has a similar average success rate as the centralized method for $k \geq 4$, it also achieved the highest rewards among all methods. This means that agents using this method take less time to learn acceptable strategies. Surprisingly, we observe in all methods, agents do prefer to achieve finish most of their tasks during the first hundred steps before slowly trying to complete and improve the global shape (Fig. 3). In addition, MADDPG agents trained on randomly generated patterns cannot generalize well on unseen and structured patterns when the number of agents is large as in Fig 3.

4 Conclusion

We showed that agents using our method can organize themselves into complex 2-dimensional pattern even though they were trained on random patterns. While a centralized technique can achieve acceptable results, our method outperformed all the baselines. Furthermore, our method generalized better to unseen environments without retraining the agents. Finally, it would be interesting to investigate our method for multi-pattern formations in which agents are expected to achieve smooth transitions between given patterns.

References

Bernstein, D. S.; Givan, R.; Immerman, N.; and Zilberstein, S. 2002. The complexity of decentralized control of markov decision processes. *Mathematics of operations research* 27(4):819–840.

Diallo, E. A. O., and Sugawara, T. 2018. Learning coordination in adversarial multi-agent dqn with dec-pomdps. Workshop on Reinforcement Learning under Partial Observability, 32nd Neurips, 2018.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, O. P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, 6379–6390.

Suzuki, I., and Yamashita, M. 1999. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal on Computing* 28(4):1347–1363.