Proceedings of the 33rd International Conference on Information Technologies (InfoTech-2019)
19-20 September 2019, Bulgaria

REINFORCEMENT LEARNING IN COLLECTIVE ROBOTS

Full paper

Vanya Dimitrova Markova and Ventseslav Kirilov Shopov

Institute of Robotics , Bulgarian Academy of Sciences
139 Rouski blvd., 4000 Plovdiv, Bulgaria
markovavanya@yahoo.com
Bulgaria

Abstract: In this article, we discuss deep learning methods for reinforcement learning agents. We study the application of multi-agent deep reinforcement learning techniques in agents. The main hypothesis is that deep reinforcement learning and collective behaviour approach demonstrate better performance than classic reinforcement learning. So autonomous agents are capable of discovering good solutions to the problem at hand by cooperate with other learners.
Key words: reinforcement learning; collective robots, knowledge transfer.

1. INTRODUCTION

Multi-Agent Systems (MAS) are studied extensively in reset years and have applications in a wide variety of domains. They make easier to create complex behaviours and improve the quality of solutions. Although the individual agents can exhibit some basic behaviours, in MAS they learn new behaviours online. In this way, the performance of the whole multi-agent system gradually improves.

Reinforcement learning (RL) is a paradigm that addresses the problem of how an agent can learn behaviour through trial-and-error interactions with a dynamic environment [1]. A RL agent learns by interacting with its environment, using a scalar reward signal as performance feedback [2, 3].

The primary contribution in this paper lies in the field of building rational collective behaviour in the face of uncertainty. We continue a series of works where

there is a partially informative response from the environment at connecting multi-agent systems. The main goal of this paper is to apply deep reinforcement learning in the area of building autonomous agents. In this way, we could build rational collective behaviour.

Also, we will describe the implementation of our approach. In the third part of the study, we describe the experiments and gathers evidence to support our hypothesis. A summary of some research in the discussed filed is presented in the next section. The third section discusses the single-agent case.

2. RELATED WORK

Pursuit-evasion is a problem area in computer science in which one group of agents attempts to catch members of another group in an environment [6]. The reinforcement learning techniques have been used in some of the recent studies in the field of pursuit-evasion games [7, 9]

In this study, we apply the techniques of Multi-Agent Reinforcement Learning (MARL) and Multi-Agent Deep Reinforcement Learning (MADRL). In addition we evaluate the results of the process of learning in collective robots (agents) work. This approach extends reinforcement learning using an artificial neural network without designing state-space or action space explicitly [3, 4, 5].

Sequential decision making under uncertainty is always a challenge for autonomous agents populating a multi-agent environment since their behaviour is inevitably influenced by the behaviour of others. Moreover, they need to profitably trade-off short-term rewards with anticipated long-term ones, while learning through interaction about the environment and others, employing techniques from RL, a fundamental area of study within Artificial Intelligence (AI) [9].

To gain new knowledge or skills, you also need to improve your knowledge or skills based on experience. Only through this, the autonomous agent can adapt to the new situation. There is a representative selection of algorithms for various research studies with multi-agent RL [11, 12].

3. REINFORCEMENT LEARNING CONCEPT

Single-agent RL concepts are given first, followed by their extension to the multi-agent case.

The single-agent case 1) Markov Decision Process (MDP): We formulate the transfer learning problem in sequential decision making domains using the following framework of Markov Decision Process is described in (1).

$$<S, A, p, R, y> \quad (1)$$

The set of states is denoted as S, A is the set of actions, p is transition function and R is a reward function. The transition function p maps the probability of moving to a new state given an action.

2) Reinforcement learning: [10] is a popular and effective method to solve an MDP. At each moment, the agent is in a given state s in S, and the agent's view is represented by a feature vector. Upon this information, the agent makes the decision which action a from a set of all possible actions A to take to reach its goal.

3) Deep Learning: Q-learning can be directly extended to deep reinforcement learning by using a deep neural network function approximator $Q(s; a_j)$ for the Q-values, where are the weights of the neural network that parametrize the Q-values. We update the neural network weights by minimizing the loss function.

Multi-agent case is an extension of multi-agent deep reinforcement learning (MADRL) approach presented in [13, 14]. We identify three major MADRL-related challenges and offer three solutions that make this approach possible. The first challenge is to present the problem in such a way that it is possible to develop an effective implementation. In other words, the problem is to present the problem in such a way, that it can be used by any number of agents without changing the deep Q-network architecture. To solve this problem, there have to be imposed several assumptions: time and space are discrete quantities, the agent's agent is 2D and the agents are divided into two groups of pursuers and evaders.

These assumptions allow us to present the state of the global system as an image-like tensor. So that each image channel contains an agent and environmental information. This presentation allows us to take advantage of the convolutional neural networks that are proven to work well for image processing tasks.

3.2. Implementation

We generate a discrete map with predefined dimensions. Then randomly place obstacles on the map. The next stage we generate two lists: one with the pursuers and one with the prey. We study the impact of the number of pursuers on the summary reward. We also investigate the impact of the number of obstacles on the performance of predator's group. So, we present our Pursue-evasion problem as an MDP task.

We define the stochastic behaviour of both of the groups imposing some additional rules. With a small probability, evader will miss the opportunity to move out and will give some handicap to the pursuer. From the other hand, the pursuer with small probability will lose the evader out from sight. So, in this way the prey get a chance to evade.

In general, predators have a small negative reward for every empty step and the prey have a small positive reward for every evasion. If a pursuer catches a prey, then

predator's reward increases considerably (at almost two orders of magnitude). From the other hand, the prey's reward will be reduced by the same amount.

The groups are implemented by two lists: one for the predators and another one for the preys. A new prey is generated in a random place on the map but out of sight of the pursuers.

For our implementation, we claim that the results of MADRL will surpass MARL approach in mater of maximum reward. So, we will reach an optimal policy for a final number of epochs (steps) faster.

The classic reinforcement learning consists of finding an optimal policy for the whole area with high details. In order to speed up the training, it is desirable that the coefficients of the policy's matrix are somewhat closer to the desired policy. This can be achieved through a TL in a simpler environment (or just in a part of the environment).

Our approach is based on following:

- ✓ In our case, group of predators(pursuers) pursue a group of preys (intruders)
- ✓ Loading the whole map and scraping all details but geometric obstacles
- ✓ Find a reinforcement learning solution for this plain map
- ✓ Use the MADRL and MARL to train both groups
- ✓ Load full map and use learned knowledge to study the impact of chosen factors in learning speed

Notation and transfer learning: Let G be the set of all possible tasks. Let G_{source} be a set of source tasks for which the pursuers and evaders has already learned a policy and let G_{target} G be another set of target tasks that have to be learned by the agents. For each task g_i in G; *let* D_i *in* R_n is a descriptor of features for the given agent(either pursuer or prey). We assume that g_i and D_i that are known to the all agents.

So, we define a target task g_j in G_{target}, as the goal of the agents. In both groups this should lead to higher summary rewards. We assume that for each pair of tasks $(g_i; g_j)$ such that g_i; and g_j are in G_{source}, the agents could reliable estimate $fu(g_i; g_j)$. E.g. the pursuer "catch" a prey and respectively the evader "evades". So both groups of agents can use these similar policies estimates to predict the expected transfer benefit between tasks in G_{source} and tasks in G_{target}.

4. KNOWLEDGE TRANSFER IN REINFORCEMENT LEARNING

We gather evidence to support the hypothesis that using Deep Learning will significantly speed up the training process of MARL. Hence we claim that building of MADR for Autonomous Agent behaviour building is more efficient than applying the MARL direct approach. We perform the following experiment: for a given map we should find an optimal autonomous agent group behaviour. The map is described by its size $(n \times n)$ and complexity rate Rc. We have two method: We compare

following approaches: case I - Multi Agent Reinforcement Learning (MARL); case II - Multi Agent Deep Reinforcement Learning (MADRL). And we have two cases: case study I - Study the impact of the number of pursuers and booty on the speed of reinforcement learning; case study II - The impact of the number of obstacles on the speed of learning.

We do the following: for a given map first, we need to find optimal behaviour of pursuers. The agent's task is to travel on a chased the maximum preys for a given amount of time. The environment is represented as a two-dimensional obstacle map. The map is described by its size (*n* x *n)* and the rate of complexity *Rc*.

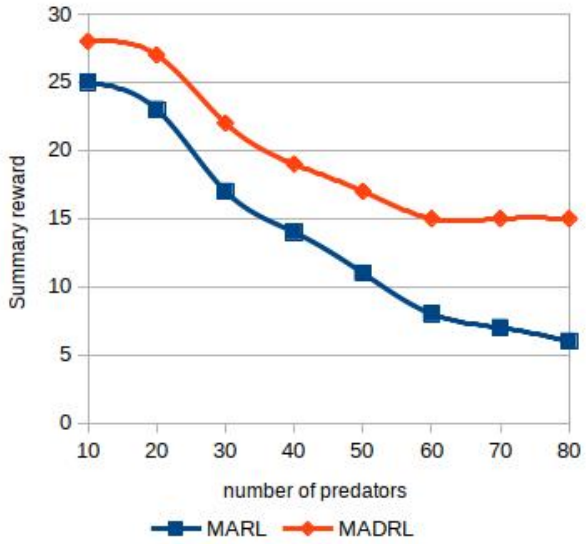

Figure 1. We study the impact of the number of predators on summary reward.

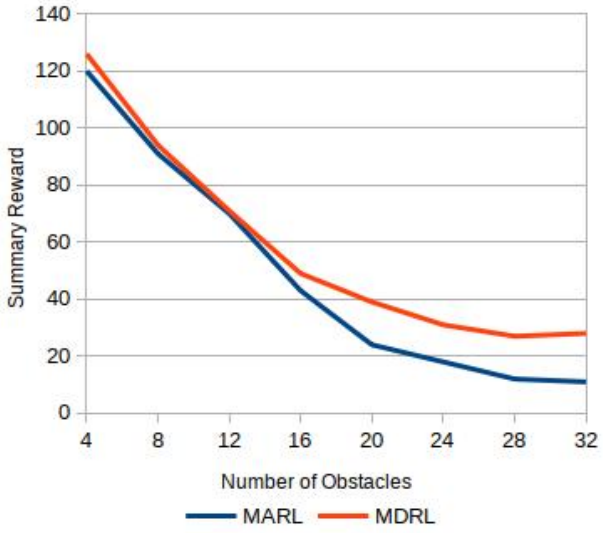

Figure 2. We study the impact of the number of obstacles on summary reward.

When the number of pursuers and prey is roughly the same then the MADRL is significantly better than the direct MARL. We study the impact of the number of pursuers and booty on the speed of reinforcement learning. The results are shown in Figure 1. Also, we study the impact of the number of obstacles on the speed of learning. One can see in Figure 2 that MADRL performs better in more complex maps.

5. CONCLUSION

The impact of different factors for the building of Multi Agent behaviour is discussed in this paper. Two different approaches are presented: Multi Agent Reinforcement Learning and Multi Agent Deep Reinforcement Learning. The impact of four factors on Reinforcement Learning performance has studied.

With the rise in the number of ages, the quality of the gauze is significantly reduced while the deep approach is weakly affected by this degradation. On a map similar to the first case, but with more obstacles, the deep approach demonstrates better performance, while MARL has a considerably lower reward. The summary reward is used as a measure of performance. In both cases, MADRL demonstrate better performance than MARL.

REFERENCES

[1] Wilson, A., Fern, A., & Tadepalli, P. (2012, June). Transfer learning in sequential decision problems: A hierarchical Bayesian approach. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning* (pp. 217-227).

[2] Wilson, A., Fern, A., Ray, S., & Tadepalli, P. (2007, June). Multi-task reinforcement learning: a hierarchical Bayesian approach. In *Proceedings of the 24th international conference on Machine learning* (pp. 1015-1022). ACM.

[3] Li, Y. **(2017)**. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274.*

[4] Lillicrap, T. P. et all. **(2015)**. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

[5] Kulkarni, T. D., Narasimhan, K., Saeedi, A., & Tenenbaum, J. **(2016)**. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in neural information processing systems* (pp. 3675-3683).

[6] Borie, R., Tovey, C., & Koenig, S. **(2009, June).** Algorithms and complexity results for pursuit-evasion problems. In *Twenty-First International Joint Conference on Artificial Intelligence*.

[7] Barrett, S., Stone, P., & Kraus, S. **(2011, May)**. Empirical evaluation of ad hoc teamwork in the pursuit domain. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2* (pp. 567-574). International Foundation for Autonomous Agents and Multiagent Systems.

[8] Grondman, I., Busoniu, L., Lopes, G. A., & Babuska, R. **(2012)**. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42*(6), 1291-1307.

[9] Gmytrasiewicz, P. J., & Doshi, P. **(2005)**. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research*, *24*, 49-79.

[10] Sutton, R. S., & Barto, A. G. **(1998)**. *Reinforcement learning: An introduction*. Cambridge, Massachusetts: MIT Press. .

[11] Nowé, A., Vrancx, P., & De Hauwere, Y. M. **(2012)**. Game theory and multi-agent reinforcement learning. In *Reinforcement Learning* (pp. 441-470). Springer, Berlin, Heidelberg.

[12] Markova, V. **(2012)**. Adaptive behaviour approach for autonomous mobile sensor agent. In *Proceedings of the International Conference on Information Technologies (InfoTech-2012)* (pp. 1314-1023).

[13] Lanctot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., Pérolat, J., & Graepel, T. **(2017)**. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems* (pp. 4190-4203).

[14] Hausknecht, M. J. **(2016)**. *Cooperation and communication in multiagent deep reinforcement learning* (Doctoral dissertation).