Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments: A Critical Review

April 2018

Eduardo Guilherme Ferreira Morais de Araújo
Instituto Superior Técnico, Universidade de Lisboa
Lisboa, Portugal
eduardo.araujo@tecnico.ulisboa.pt

Abstract A Critical Review of Lowe, Ryan, et al. "Multi-agent actor-critic for mixed cooperative-competitive environments." Advances in Neural Information Processing Systems. 2017.

Keywords Critical Review, Multi-Agent, Actor-Critic, Cooperative Systems.

1. Introduction

Recent advancements in reinforcement learning (RL) confirm that RL techniques [8], combined with deep learning [9], can solve challenging problems ranging from game playing [12, 16] to robotics [10]. Particularly, RL proved to be a powerful tool for solving single agent Markov Decision Processes (MDP's), where modelling or predicting the behaviour of other actors in the environment is largely unnecessary [17].

Multi-agent systems, however, are finding applications in a variety of domains including robotic teams, distributed control, resource management, collaborative decision support systems, data mining, and more [2], where emergent behaviour and complexity arise from agents co-evolving together. Moreover there are a number of situations where multi-agent systems are the most natural way of looking at a system.

Unfortunately, when multiple agents share the environment and influence each other, the convergence guarantees of RL traditional methods such as Q-Learning no longer hold, since from the perspective of any individual agent the environment becomes non-stationary [20].

2. Related Work

The related work in this area suggests there is not a good and generic approach to learning in multi-agent settings. The simplest of them is to use independently learning agents. However, [18] tried this approach, while [11] concluded it does not perform well in practice, since it is impossible to rely on experience replay in a non-stationary environment. To address this problem, it is possible either to input other agent's policy parameters to the Q function [19] or explicitly add the iteration index to the replay buffer [5], using importance sampling.

Another way of facing multi-agent systems is to note that agent interactions can either be cooperative, competitive, or both, and design algorithms only for a particular nature of interaction. The most studied of which are cooperative, where it is assumed the actions of other agents are made to improve collective reward. Another approach is forcing cooperation via sharing of policy parameters, which requires homogeneous agent capabilities. The main problem with these algorithms is that they are generally not applicable in competitive or mixed settings.

Recent work focused on learning grounded cooperative communication protocols between agents to solve various tasks [3, 13, 14], which yields solutions only applicable when the communication between agents is carried out over a dedicated, differentiable communication channel.

Acknowledging the merits and faults of traditional RL approaches, Lowe, Ryan, et al. (2017) inspired by [3] adapted actor-critic methods to develop a general-purpose multi-agent learning algorithm that allows agents to operate under conditions where learned policies can only use local information and there are no assumptions regarding either the model of the environment dynamics, or the structure on the communication method between agents. Additionally, it was also devised a method to improve the stability of multi-agent policies by training agents with an ensemble of policies.

3. Methods

The authors methodology consisted on considering an extension of Markov decision processes called partially

observable Markov games and adopting the framework of centralized training with decentralized execution, i.e. it was allowed for the policies to use extra information *only* during training. Thus proposing a simple extension of actor-critic policy gradient methods where the critic is augmented with extra information about the policies of other agents.

To obtain multi-agent policies that are more robust to changes in the policy of competing agents, authors suggested yet to train agents with policy ensembles, in a way of coping with the environment non-stationarity due to the agents' changing policies. Lowe, Ryan, et al. (2017) did so by randomly selecting one particular sub-policy for each agent to execute, at each episode.

4. Experiments

To perform experiments, a two-dimensional world with continuous space and discrete time was adopted and, although agents could take physical and communication actions, it was not assumed all agents had identical action and observation spaces, nor that they acted according to the same policy π. The tasks considered consisted of a battery of both cooperative and competitive games requiring a series of different physical and communication actions in order to achieve the best reward. A list of the played games is presented in Table 1.

Table 1
Designed game tests

Games	Setting
Cooperative communication	Cooperative
Cooperative navigation	Cooperative
Keep-away	Mixed
Physical deception	Competitive / Mixed
Predator-prey	Competitive / Mixed
Covert communication	Competitive

5. Results

To evaluate the quality of policies learned in competitive settings, agents trained using the proposed algorithm (MADDPG) were pitched against agents trained under traditional RL methods and the resulting success of the agents and adversaries was compared.

In a nutshell, the proposed method significantly outperformed traditional RL algorithms in every environment, with the only downside being the input space of Q grows linearly with the number of agents N, as he authors point out.

It should also be mentioned during these tests a peculiar behaviour of traditional RL methods agents lead to a surprising hypothesis. Namely that many of the multi-agent methods previously proposed for scenarios with short time horizons may not generalize to more complex tasks. In the heart of this idea is the fact that in the cooperative communication scenario traditional RL methods failed to learn the correct behaviour, and in practice one of agents learned to ignore the other, jeopardizing the completion of the task. Observations suggest this is due to the lack of a consistent gradient signal, problem that is exacerbated as the number of time steps grows.

Additionally, it was also shown to be possible to improve the performance of the MADDPG algorithm by training agents with an ensemble of policies. In order to assess the effectiveness of policy ensembles, the authors focused on competitive environments, enforcing that cooperative agents should have the same policies at each episode, and similarly for the adversaries. Observations ultimately showed agents with policy ensembles were stronger than those with a single policy. The results obtained reinforced the authors' belief of such an approach being generally applicable to any multi-agent algorithm

6. Conclusions

If a scientific paper is as relevant as the questions it arises, this one proves to be of significant importance in the field of Reinforcement Learning, since it questions the generalization of previous multi-agent methods proposed for scenarios with short time horizons, while presenting both a general-purpose multi-agent learning algorithm that outperforms traditional RL methods, and a generally applicable way of improving convergence speed in RL training.

The designed tests proved to be suitable, the results clear and its analysis thorough. Notwithstanding, concurrently to this paper, [4] proposed a similar idea of using policy gradient methods with a centralized critic. Although the latter focused only on cooperative environments, it would be interesting to compare the performance of both methods using the same battery of tests used to evaluate the MADDPG algorithm.

All in all, this paper proposes a generic solution to a specific, practical problem. And although the intuition underlying the core idea yields some parallels with real agents, in the sense that even humans usually train together sharing all kinds of information before being able to act in a decentralized, autonomous and coordinated way (e.g. collective sports, military exercises), from an Artificial Intelligence point of view it fell short on expectations, since there was no allusion on how were agents coordinating themselves, or what was their decision making process, which renders the agents' individual learning and collective coordination processes uninterpretable.

Moreover, from a practical (and probably biased Control Systems) point of view, analysing the videos of the experimental results, one cannot but conclude there is yet a long way until this type of general purpose multi-agent RL methods to be able to perform well in real world situations, since agents exhibited an oscillatory behaviour around their final target positions rather than remaining there still.

The aforementioned problems may seem of minor importance if one focus only on inconsequent tasks such as trivial games, but to deal with both lack of interpretability and oscillatory behaviour are of paramount importance in real situations, specially if human safety can be compromised. Examples of these situations range from simple logistic tasks such as robots (mobile or fixed) operating on environments where they are allowed to coexist with human operators to more complex tasks such as rescue and security patrolling missions.

To cope with the lack of interpretability the most obvious (and possibly the simplest) is to switch from a black-box to an expert knowledge paradigm, where instead of expecting agents to come up with opaque solutions, the agent designer makes use of knowledge from experts and inputs it into the agent's decision making function.

From a classical AI point of view, one possible solution is to implement a belief–desire–intention (BDI) model of agency such as the Procedural Reasoning System (PRS) system [6, 7, 21], where the planning module simply chooses one of the plans from a library of expert knowledge designed plans [22].

On the other hand, a typical Control Systems approach would be to make use of Fuzzy Logic as a framework for intelligent decision support. By definition fuzzy logic provides a way to develop and code rule-based behaviours, based on expert knowledge. Besides interpretability, fuzzy logic has yet the advantage of being able to be refined as new information becomes available [15]. Moreover, since the fuzzy logic approach can deal with various situations without analytical model of environments, it is easy to integrate it with Reinforcement Learning techniques [1, 23].

References

[1] Hee Rak Beom and Hyung Suck Cho. A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning. *IEEE transactions on Systems, Man, and Cybernetics*, 25(3):464–477, 1995.

[2] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. Multi-agent reinforcement learning: An overview. In *Innovations in multi-agent systems and applications-1*, pages 183–221. Springer, 2010.

[3] Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2137–2145, 2016.

[4] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. *arXiv preprint arXiv:1705.08926*, 2017.

[5] Jakob Foerster, Nantas Nardelli, Gregory Farquhar, Philip Torr, Pushmeet Kohli, Shimon Whiteson, et al. Stabilising experience replay for deep multi-agent reinforcement learning. *arXiv preprint arXiv:1702.08887*, 2017.

[6] Michael P Georgeff and Francois Felix Ingrand. *Decision-making in an embedded reasoning system*. Citeseer, 1989.

[7] Michael P Georgeff, Amy L Lansky, and Marcel J Schoppers. Reasoning and planning in dynamic domains: An experiment with a mobile robot. Technical report, SRI INTERNATIONAL MENLO PARK CA, 1987.

[8] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[11] Laetitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *The Knowledge Engineering Review*, 27(1):1–31, 2012.

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[13] Peng Peng, Quan Yuan, Ying Wen, Yaodong Yang, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017.

[14] Peng Peng, Quan Yuan, Ying Wen, Yaodong Yang, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017.

[15] Gloria Phillips-Wren. Ai tools in decision making support systems: a review. *International Journal on Artificial Intelligence Tools*, 21(02):1240005, 2012.

[16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[17] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[18] Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pages 330–337, 1993.

[19] Gerald Tesauro. Extending q-learning to general adaptive multi-agent systems. In *Advances in neural information processing systems*, pages 871–878, 2004.

[20] John N Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine learning*, 16(3):185–202, 1994.

[21] Jeffrey Tweedale, Nikhil Ichalkaranje, Christos Sioutis, Bevan Jarvis, Angela Consoli, and G Phillips-Wren. Innovations in multi-agent systems. *Journal of Network and Computer Applications*, 30(3):1089–1115, 2007.

[22] Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[23] Changjiu Zhou and Qingchun Meng. Dynamic balance of a biped robot using fuzzy reinforcement learning agents. *Fuzzy sets and Systems*, 134(1):169–187, 2003.