Genetic Deep Reinforcement Learning for Mapless Navigation

Extended Abstract

Enrico Marchesini
University of Verona
Verona, Italy
enrico.marchesini@univr.it

Alessandro Farinelli
University of Verona
Verona, Italy
alessandro.farinelli@univr.it

ABSTRACT

We consider Deep Reinforcement Learning (DRL) approaches to devise mapless navigation strategies for mobile platforms. We propose a Genetic Deep Reinforcement Learning (GDRL) method that combines Genetic Algorithms (GA) with discrete and continuous action space DRL approaches. The goal of GDRL is to reduce the sensitivity of DRL approaches to their hyper-parameter tuning and to provide robust exploration strategies. We evaluate GDRL in combination with Rainbow and Proximal Policy Optimization (PPO) in two navigation scenarios: i) a wheeled robot avoiding obstacles in an indoor environment and ii) a water drone that must reach a predefined location in presence of waves. Our empirical evaluation demonstrates that GDRL outperforms state-of-the-art DRL and GA methods as well as a previous hybrid approach.

KEYWORDS

Deep Reinforcement Learning, Genetic Algorithms, Mapless Navigation, Water Drones

ACM Reference Format:
Enrico Marchesini and Alessandro Farinelli. 2020. Genetic Deep Reinforcement Learning for Mapless Navigation. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 INTRODUCTION

The key to successfully apply DRL in robotics is the ability of adapting to the surrounding environment by generalizing from the training experiences. Robotic applications, however, have to cope with the uncertainties of both the physical hardware and the operational environment, hence they usually require a huge number of trials to achieve reasonable and stable performances. For this reason, devising stable learning approaches while reducing training time is a key issue for the application of DRL in real applications. Despite the promising results of DRL in robotics [8, 9, 17, 18, 22], these approaches still present open challenges such as avoiding fast convergence to a local optimal, mainly caused by the lack of diverse exploration when operating in high-dimensional spaces. An effective approach to address these challenges is the optimization of Evolutionary Strategies (ES) [2] and, in particular, GA [10].

In this paper, we propose a novel hybrid methodology, which we refer to as Genetic Deep Reinforcement Learning (Figure 1), to combine GA with DRL. Our goal is to exploit the high sampling-efficiency of DRL algorithms while incorporating a GA-based evaluation to search for a better-performing policy, hence improving the stability of the training and reducing training time. To validate our framework in the wide area of DRL, we propose the application of GDRL to both discrete and continuous action space domains, using two state-of-the-art learning algorithms such as Rainbow [5] and PPO [15]. This extends the use of our methodology beyond actor-critic settings of previous approaches, allowing the use of powerful discrete action space optimizations [3, 4, 14, 19, 21].

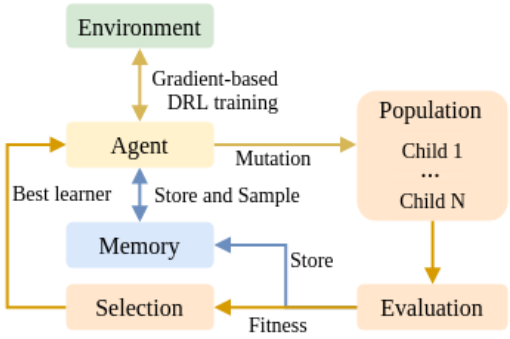

Figure 1: High level schematic of GDRL.

2 GENETIC DRL

Recently, there has been an increasing interest in the use of ES as an alternative for DRL [13]. In particular, [16] shows how a simple GA, based only on mutations, can be a competitive alternative to more complex gradient-based alternatives (e.g., DQN). Following these results, an emergent research direction focuses on combine gradient-free and gradient-based solutions into a hybrid framework [1, 6, 12]. These works share a common baseline in the first attempt to merge ES and DRL, called Evolutionary Reinforcement Learning (ERL) [6]; given this as a shared baseline architecture for [1, 12], we choose to compare GDRL with ERL.

In contrast to previous works, we evaluate GDRL on two mapless navigation scenarios, a well-known benchmark in recent DRL literature [18, 20, 22]. In particular, we consider a wheeled robot that must reach a target point while avoiding obstacles in an indoor environment, and an aquatic drone that must reach a target point in presence of waves. Our two environments present different characteristics (e.g., static and dynamic environments, sparse and dense rewards), and we show that using only the evolutionary component of the framework it is not possible to deal with the high

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

generalization required by the indoor scenario[1]. We show how we combine both discrete (i.e., Rainbow) and a continuous (i.e., PPO) action space DRL algorithms with a pure GA method [16]. These combinations result in two algorithms named GRainbow and GPPO. We compare the performance of GDRL to the stand-alone GA, Rainbow [5], PPO [15] and the to ERL framework introduced in [6] and applied to the PPO approach (we refer the interested reader to the original papers for details on the considered algorithms).

2.1 Genetic Evaluation

In this section, we show the details of the periodical evaluation.

Overall approach: as in a standard training setup, a main DRL agent (drl_a, with weights θ_a) starts to collect experiences interacting with its environment. Such experiences are stored in a replay buffer to train the network. Periodically, GDRL generates a population of children, each one characterized by a different genome. Each child is generated by applying a mutation function to θ_a. Furthermore, a separate thread is instantiated for each child, along with a copy of the environment. The weights in θ_a are used to create the N individuals applying Gaussian noise to the parameter vector: $\theta_a + mut_p n$, where $n \sim N(0, mut_v)$ and mut_p is the mutation probability. These children, together with a copy of θ_a are then evaluated over a set S of episodes in their copy of the drl_a environment and the best performing individual is replaced to the current drl_a. We use the same genetic hyper-parameters for both GRainbow and GPPO. In particular, the size N of the population P is set to 10 and the number of trials conducted in the evaluation phase, to compute the fitness score, ranging from 10 to 20 across tasks. The mutation probability of a network weight is $mut_p = \{0.75, 0.4, 0.1\}$ based on the current success rate (i.e., the number of obstacle-free trajectories that reach the target over the last 100 episodes) and a constant mutation value $mut_v = 0.1$. Our evaluation shows that in contrast to [6] our mutation pattern finds more often a better policy. In particular, in [6] the network switch occurs more rarely ≈ 25 times over 250 trials, while in our case we have ≈ 30 times in 45 evaluations.

GRainbow for the indoor navigation: the genetic evaluation of the Rainbow agent presents some challenges given the instability of the training algorithm and all its optimizations which have to cooperate in order to successfully train the agent [2]. In particular, when the genome of the drl_a is switched with one of the mutated children a soft update of the target network is performed with $\tau = 0.1$ to approach the new weights θ_a; we tried different setup for this, but copying the 10% of the weights showed us the better performance. Moreover, we exploit the redundancy offered by the population, to introduce some of the experience of the genetic evaluation into the same prioritized buffer of the drl_a.

GPPO for the water drone scenario: in contrast to GRainbow, the genetic evaluation phase of the continuous PPO agent does not present particular issues, given the simple setup of the algorithm and the reduced number of hyper-parameters. Given the limited size of the buffer, we did not include the experience of the children agent in this case.

[1] In contrast, GA have been successfully used in locomotion [7, 11]

[2] Such instability are a known drawback of DQN-based algorithms. Nonetheless, this remains the stat-of-the-art for discrete action space.

Table 1: Average Performance in the Evaluation Phase

Algorithm	Reward	Step	Sim. Time (s)	Speed (m/s)
Rainbow	36.2	370	42	constant
GRainbow	38.2	272	32	constant
PPO	4.5	98	27	0.22
GPPO	4.8	89	22	0.23

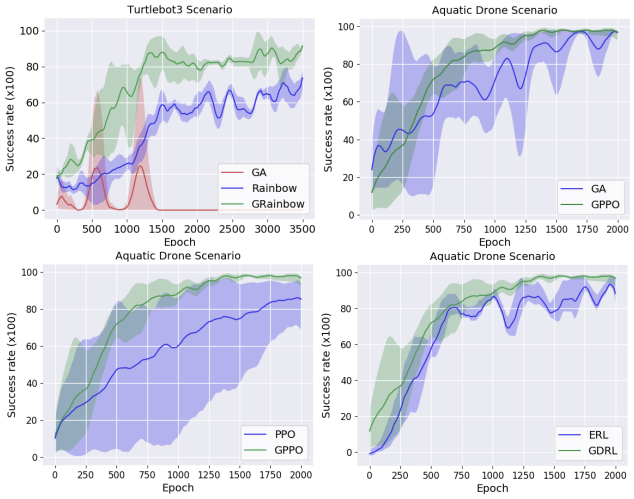

Figure 2: Top: Success rate for the Turtlebot3: GA-Rainbow-GRainbow (left) and for the water scenario: GA-GPPO (right). Bottom: PPO-GPPO (left) and ERL-GPPO (right).

3 EMPIRICAL EVALUATION

The collected data in Figure 2 are related to training phases performed on an Intel i7-8550U. After the learning phase, the trained models are able to navigate, generalizing: (i) robot starting position, (ii) target position, (iii) velocity. We evaluated the trained models of Rainbow, GRainbow, PPO and GPPO to compare the performance of each method; Table 1 resumes the collected data. Crucially, GDRL based algorithms outperform the standard methods in every considered metrics.

4 DISCUSSION

We presented GDRL, a novel hybrid framework that exploits the robustness of population-based GA to improve DRL agents. Crucially, our GDRL method is the first framework that combines GA and DRL for both continuous and discrete action space methods in navigation scenarios. This work paves the way for several interesting research directions which includes the possibility of extending GDRL in a multi-agent scenario, exploiting the crossover of similar agents.

5 ACKNOWLEDGEMENT

The research has been partially supported by the projects "Dipartimenti di Eccellenza 2018-2022", funded by the Italian Ministry of Education, Universities and Research (MIUR).

REFERENCES

[1] Cristian Bodnar, Ben Day, and Pietro Lio'. 2019. Proximal Distilled Evolutionary Reinforcement Learning. In *CoRR*.

[2] David Fogel. 2006. Evolutionary computation - toward a new philosophy of machine intelligence (3. ed.).

[3] Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Rémi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. 2017. Noisy Networks for Exploration. In *CoRR*.

[4] Roy Fox, Ari Pakman, and Naftali Tishby. 2015. Taming the Noise in Reinforcement Learning via Soft Updates. In *AUAI*.

[5] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. 2018. Rainbow: Combining Improvements in Deep Reinforcement Learning. In *AAAI*.

[6] Shauharda Khadka and Kagan Tumer. 2018. Evolutionary Reinforcement Learning. In *NIPS*.

[7] Jeong Hoon Lee and Jong Hyeon Park. 2019. Time-dependent genetic algorithm and its application to quadruped's locomotion. In *Robotics and Autonomous Systems*.

[8] E. Marchesini, D. Corsi, A. Benfatti, A. Farinelli, and P. Fiorini. 2019. Double Deep Q-Network for Trajectory Generation of a Commercial 7DOF Redundant Manipulator. In *IRC*.

[9] Enrico Marchesini and Alessandro Farinelli. 2020. Discrete Deep Reinforcement Learning for Mapless Navigation. In *ICRA*.

[10] David J. Montana and Lawrence Davis. 1989. Training Feedforward Neural Networks Using Genetic Algorithms. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*.

[11] Miguel Oliveira, Lino Costa, Ana Rocha, Cristina Santos, and Manuel Ferreira. 2011. Multiobjective Optimization of a Quadruped Robot Locomotion Using a Genetic Algorithm. In *Soft Computing in Industrial Applications*.

[12] Pourchot and Sigaud. 2019. CEM-RL: Combining evolutionary and gradient-based methods for policy search. In *ICLR*.

[13] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. In *arXiv*.

[14] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2016. Prioritized Experience Replay. In *ICLR*.

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. In *CoRR*.

[16] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. 2017. Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning. In *CoRR*.

[17] Lei Tai and Ming Liu. 2016. Towards Cognitive Exploration through Deep Reinforcement Learning for Mobile Robots. In *CoRR*.

[18] L. Tai, G. Paolo, and M. Liu. 2017. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In *IROS*.

[19] Hado van Hasselt, Arthur Guez, and David Silver. 2016. Deep Reinforcement Learning with Double Q-learning. In *AAAI*.

[20] Ayzaan Wahid, Alexander Toshev, Marek Fiser, and Tsang-Wei Edward Lee. 2019. Long Range Neural Navigation Policies for the Real World. In *CoRR*.

[21] Ziyu Wang, Nando de Freitas, and Marc Lanctot. 2016. Dueling Network Architectures for Deep Reinforcement Learning. In *ICML*.

[22] J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard. 2017. Deep reinforcement learning with successor features for navigation across similar environments. In *IROS*.