# Towards Continual Reinforcement Learning through Evolutionary Meta-Learning

Djordje Grbic
IT University Copenhagen
djgr@itu.dk

Sebastian Risi
IT University Copenhagen
sebr@itu.dk

## ABSTRACT

In continual learning, an agent is exposed to a changing environment, requiring it to adapt during execution time. While traditional reinforcement learning (RL) methods have shown impressive results in various domains, there has been less progress in addressing the challenge of continual learning. Current RL approaches do not allow the agent to adapt *during execution* but only during a dedicated training phase. Here we study the problem of continual learning in a 2D bipedal walker domain, in which the legs of the walker grow over its lifetime, requiring the agent to adapt. The introduced approach combines neuroevolution, to determine the starting weights of a deep neural network, and a version of deep reinforcement learning that is continually running during execution time. The proof-of-concept results show that the combined approach gives a better generalisation performance when compared to evolution or reinforcement learning alone. The hybridization of reinforcement learning and evolution opens up exciting new research directions for continually learning agents that can benefit from suitable priors determined by an evolutionary process.

## KEYWORDS

Reinforcement learning, Continual learning, Meta-learning

**ACM Reference Format:**
Djordje Grbic and Sebastian Risi. 2019. Towards Continual Reinforcement Learning through Evolutionary Meta-Learning. In *Proceedings of the Genetic and Evolutionary Computation Conference 2019 (GECCO '19).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION & RELATED WORK

The current way that AI methods learn is fundamentally different than the ways humans learn; they only perform well in situations they have been trained for in advance and are not able to learn new knowledge during execution time. If situations or circumstances change only slightly, current AI systems will likely fail to perform well. These issues clearly limit the usage of these machines, which have to be taken offline to be re-trained.

An approach that tries to address this challenge is meta-learning. The idea of meta-learning or *learning to learn* has been around since the late 1980s and early 1990s but recently gathered wider interest [4]. One of the earliest proponents of meta-learning was Schmidhuber, who invited methods that allowed neural networks to learn how to modify their own weights [6] but while very general, these approaches proved difficult to train. A recent trend in meta-learning is to find good initial weights (e.g. through gradient descent [3] or evolution [2]), from which adaptation can be performed in a few iterations. One such approach is Model-Agnostic Meta-Learning (MAML) [3], which allows simulated robots to quickly adapt to different goal directions.

The approach presented in this paper aims to combine the benefits of artificial evolution with the sample efficiency of state-of-the-art RL algorithms to create a continually learning agent. Through a meta-learning setup, evolution is tasked with finding good initial weights that allow a reinforcement learning algorithm to continually adapt the networks' weights during the *lifetime* of the agent. The approach shares similarities with the work by Fernando et al. [2], in which the authors also evolve parameters for a network that then learns further through evolution but the way it is applied is different. Here we are interested in discovering parameters that allow reinforcement learning to *continually learn*, while the goal in Fernando et al.'s approach was to learn as fast as possible.

We test our approach on a new 2D bipedal walker domain (`Growing-Bipedal-Walker`) that was modified to increase the size of the walker's legs while it is exploring the environment. This domain requires the agent to continually adapt while avoiding falling over or getting stuck.

## 2 CONTINUAL REINFORCEMENT LEARNING THROUGH NEUROEVOLUTION

The approach in this paper is a combination of an evolutionary algorithm, which searches for the initial network weights, and an RL algorithm that adapts the model to a changing environment during the lifetime of the agent. A lifetime of the agent is defined as a single episode in the environment, during which the RL algorithm has to transform the reward signal into weight corrections. The corrections made by the RL algorithm disappear after the episode ends. Details of the approach are shown in Algorithm 1. Note that, in contrast to typical RL algorithm, RL is still updating the weights of the agent's neural network even during execution.

### 2.1 The Growing Bipedal Walker Domain

The approach in this paper is tested on a modified version of the `Bipedal-Walker-v2`, a commonly used environment present in the OpenAI Gym library [1]. The environment consists of a single bipedal walker that has to traverse a randomly constructed, mostly flat terrain. The walker needs to get to the rightmost end of the world in a limited amount of time. In the introduced modified version of the task, the legs of the walker are increased three times during the lifetime of the agent (Figure 1). The modified environment presented in this paper splits the terrain into three equally sized parts (beginning, middle, and end). At the start of each part the agent is reconstructed with a new legs length, where the length is sampled from three uniform distributions: $U(0.5, 0.7)$ for the beginning, $U(0.9, 1.1)$ for the middle and $U(1.3, 1.5)$ for the end part of

*GECCO '19, July 13–17, 2019, Prague, Czech Republic*
2019. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

**Figure 1: Growing bipedal walker.** In the 2D bipedal walker tasks in this paper the legs of the agent grow over its lifetime. Shown are samples of leg lengths in the three growth phases.

the world. This schedule is supposed to: (a) simulate the growth of the legs over time, (b) challenge the agents to generalize by varying the body composition, and (c) to minimize the influence of "luck" by controlling the minimum amount of change at predefined intervals (Figure 1). A video of an agent successfully solving the task can be found here: https://youtu.be/cO3h_YinngE.

## 2.2 Training setup

The agent model has two networks (actor and critic), each with two hidden layers of 64 fully-connected neurons. Each model runs 100 episodes and the cumulative episode rewards are taken as the fitness of the agent. Three different setups are compared to elucidate the advantages of the evolutionary meta-learning approach:

**RL**: A2C algorithm with a separate network for the actor and the critic (which is a synchronous version of A3C [5].), which is run with 16 parallel environments and the algorithm performs the gradient update after 200 steps in each environment. The training parameters are chosen manually to get a good trade-off between performance and training times. Training runs for 5,000 gradient descent steps.

**EA**: Darwinian evolution determines the weights of the controller neural network. The weights of the network stay unchanged during the lifetime (evaluation episode) of the individual controller and their fitness is the cumulative episode reward. The top 20 individuals are chosen as parents and are re-evaluated 10 times. The individual with the best median fitness is copied unchanged to the next generation.

**EA + lifetime RL**: The combined approach in which the RL algorithm runs gradient descent every 20 steps. The fitness function and selection scheme are the same as in the EA experiments.

**Algorithm 1** Continual Learning Through Evolution

```
function LIFETIME(evolved_weights, env)
    model = init(evolved_weights); state = env.reset()
    for not done do
        for step = 1 to 20 do
            state, done, reward = env.step(model.act(state))
            if env.phase_finished then
                env.grow_legs()
            end if
            obs_buffer.add(state, reward)
        end for
        A2C.gradient_descent(model, obs_buffer)
        obs_buffer.reset()
    end for
    return env.episode_reward
end function
```

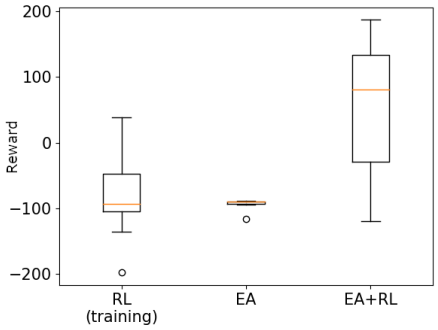


**Figure 2:** Testing reward over 100 trials.

## 3 RESULTS & DISCUSSION

All models are evaluated on 100 episodes during testing (Figure 2). The results show that the model produced by the EA+RL method produces higher scoring episodes more often. Furthermore, this setup is the only one that produces episodes with a reward higher than 100. The high variance is the obvious downside of the approach and requires further investigation. The choice of the RL algorithm, for instance, could have a significant impact on the robustness of the approach. While an EA by itself is unable to learn a high-performing policy, in combination with on-line RL it presents a promising direction for further research into continual learning.

## ACKNOWLEDGMENTS

This work was supported by the Lifelong Learning Machines program from DARPA/MTO under Contract No. FA8750-18-C-0103. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA.

## REFERENCES

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. (2016). arXiv:cs.LG/1606.01540

[2] Chrisantha Thomas Fernando, Jakub Sygnowski, Simon Osindero, Jane Wang, Tom Schaul, Denis Teplyashin, Pablo Sprechmann, Alexander Pritzel, and Andrei A Rusu. 2018. Meta Learning by the Baldwin Effect. *arXiv preprint arXiv:1806.07917* (2018).

[3] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400* (2017).

[4] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2016. Building machines that learn and think like people. *Behavioral and Brain Sciences* (2016), 1–101.

[5] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*. 1928–1937.

[6] Jürgen Schmidhuber. 1992. Learning to control fast-weight memories: an alternative to dynamic recurrent networks. *Neural Computation* 4, 1 (1992), 131–139.