Meta-learning in evolutionary reinforcement learning: some paths forward

Bruno Gašperov, PhD

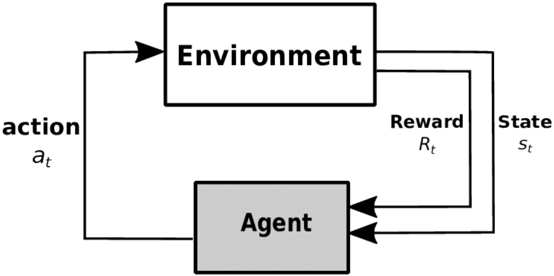
Prof. Branko Šter, PhD

October 8, 2024

SMASH
machine learning for science and humanities postdoctoral program

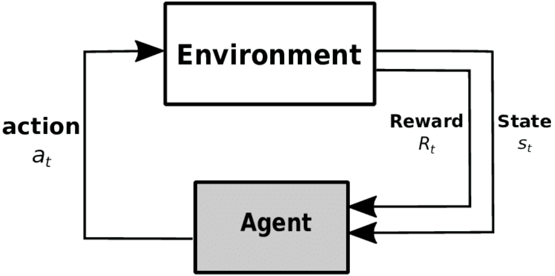
UNIVERSITY OF LJUBLJANA
FRI
Faculty of Computer and Information Science

Introduction - reinforcement learning (RL)



- A decision-maker (**the agent**) interacts with some **environment** that changes states
- The agent observes state S_t, selects action A_t which **leads to reward** R_{t+1} and **influences the next state** S_{t+1}, etc.
- Its behavior is given by a policy π mapping states to actions (deterministic) or probability distributions over the action space (stochastic)
- Trial-and-error interaction yields trajectories $(S_0, A_0, R_1, ..., S_t, A_t, R_{t+1}, ...)$ based on which learning is done

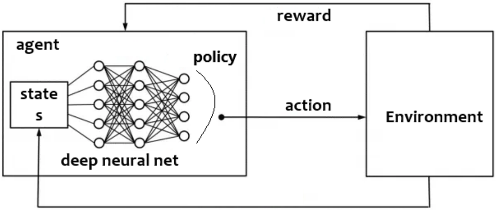
Introduction - reinforcement learning (RL) (cont.)



- The agent tries to maximize the expected return: $G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$ by selecting actions given states
- $0 \leq \gamma \leq 1$ is a discount factor: $\gamma = 0 \rightarrow$ myopic agent, $\gamma = 1 \rightarrow$ long-termist
- Finding **the optimal policy** π^* that maximizes the expected return
- Formally, modeled as a Markov Decision Process (MDP), given by the tuple: $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $\mathcal{P}$ the probability transition matrix, $\mathcal{R}$ the reward function
- Markov property: S_{t+1} depends only on S_t and A_t and not the history of states/actions

Deep reinforcement learning (DRL)

- The use of deep neural networks (DNNs) as **function approximators** in RL
  - used to approximate entities of interest, commonly the policy π, parametrized as π_θ, where θ denotes the DNN parameters



- Astonishing accomplishments in multiple domains: games [1], robotics [2], etc.

Meta-reinforcement learning (metaRL)

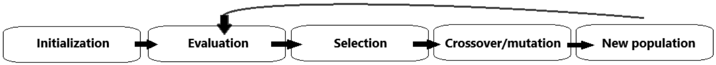
- In meta-RL [5, 6], instead of solving a single task (environment), the goal is quick adaptation to different, unseen tasks (environments)
- Using knowledge from previous tasks to tackle new ones
- Represent [6] some meta-knowledge (meta-parameters) as ω. Now we search for:

$$\omega^* = \arg\max_{\omega} \mathbb{E}_{\mathcal{M}\sim p(\mathcal{M})} \mathbb{E}_{\tau\sim\mathcal{M},\pi_{\theta^*}} [G_T] \tag{1}$$

  where $\mathcal{M}$ denotes an MDP, $p(\mathcal{M})$ a distribution over MDP-s, τ a trajectory /an episode, and T the total number of time-steps in an episode
- Essentially bi-level optimization:
  - the inner level (loop) optimizes **the objective** (i.e., RL policy **parameters** θ)
  - the outer level optimizes **the meta-objective** (e.g., reward formulation, initialization, any type of **meta-parameter** ω)

Evolutionary reinforcement learning (evoRL)

- Evolutionary reinforcement learning (evoRL) includes any method integrating evolutionary computation (EC) into RL, including metaRL
  - Directly finding (near-)optimal policies π^* (*policy search*)
  - Finding a wide array of policies exhibiting mutually diverse behaviors (*diversity encouragement*)
  - Finding the optimal initialization of policy parameters (*meta-learning*)
  - Reward shaping (also *meta-learning*)
  - etc.
- Why evoRL?
  - Papers showing that evolutionary strategies (ES) [3] and genetic algorithms (GA) [4] offer a competitive alternative to gradient-based approaches
  - Simple, can also work with deterministic policies, reducing the noise



Meta evolutionary reinforcement learning (meta-evoRL)

- Optimizing the outer loop (meta-objective) in a gradient-free manner [5, 6]
  1. no need for explicit bi-level optimization
  2. works with non-differentiable meta-objectives
  3. avoid the high computational overhead of high-order gradients
  4. scalable: easy parallelization (population-based)
- Example: population-based evolution via a genetic algorithm, where each solution (individual) is given by:

$$x = (\theta_1, \theta_2, \ldots, \theta_n, \omega_1, \omega_2, \ldots, \omega_m) \tag{2}$$

  where n (resp. m) is the number of parameters (resp. meta-parameters).
- The parameters and meta-parameters then coevolve
- Search in the union of the space of parameters and meta-parameters ($\Theta \cup \Omega$)

FERLUDE SMASH and meta-learning

- As part of my recently started SMASH project FERLUDE (**Few-shot evolutionary reinforcement learning under uncertain and dynamic environments**), we are particularly interested in exploring the intersection of *evolutionary computation, reinforcement learning, and meta-learning*
- We are particularly interested in investigating underemployed evolutionary/biological mechanisms and principles in the context of evoRL and meta-evoRL - these are **not** novel meta-heuristics
- We hypothesize that the use of evolutionary concepts/principles such as evolvability and higher-order mutation rates can lead to more robust evoRL agents, especially when facing dynamic (non-stationary) environments

Principle 1: evolvability

- While many definitions of **evolvability** exist, it is commonly defined as *the ability of an individual or population to produce offspring with mutually diverse behaviors/phenotypes*
- From an EC/ERL perspective, two separate functions are needed:
  - the fitness function $f : \Theta \mapsto \mathbb{R}$ mapping solutions to fitness values
  - the behavior function $b : \Theta \mapsto \mathcal{B}$ mapping solutions to their corresponding behaviors/phenotypes
- Used in quality-diversity (QD) and novelty search (NS) families of approaches
- Example: given robot parameters $\boldsymbol{\theta}$, $f(\boldsymbol{\theta})$ is the robot's speed, and $b(\boldsymbol{\theta})$ the type of its gait (e.g. one-legged, symmetric, etc.)

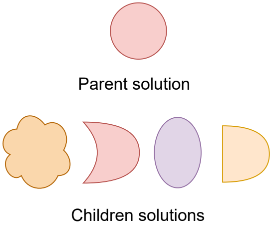


Figure: Phenotypically evolvable solution where phenotype = [color, shape]

Principle 1: evolvability (cont.)

- A solution $\boldsymbol{\theta}$ is phenotypically evolvable if small perturbations of $\boldsymbol{\theta}$ (representing its children) lead to significant changes in the corresponding phenotypes/behaviors ($\boldsymbol{\theta}' \approx \boldsymbol{\theta}$, $b(\boldsymbol{\theta}') \not\approx b(\boldsymbol{\theta})$)
- Highly evolvable solutions might serve as good starting points (initializations) when facing dynamic environments, as only a few mutations are needed to obtain different behaviors, each of which might perform well under different circumstances

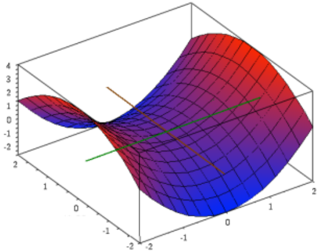
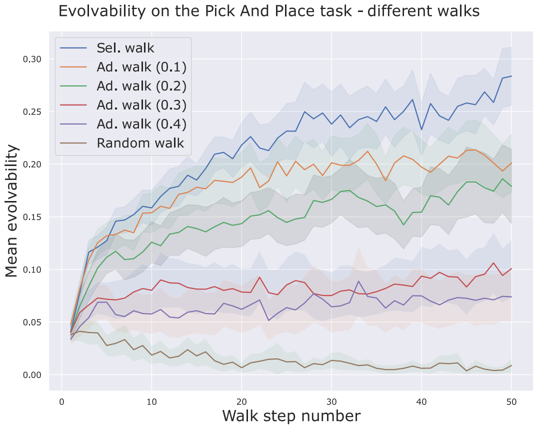


Figure: Imagine evolvable solutions as saddle points in the $\Theta \mapsto \mathcal{B}$ mapping

Principle 1: evolvability (cont.)

- Some prior research: Gasperov *et al.* [7] study evolvability in the context of neuroevolutionary divergent search (a form of novelty search) on an evoRL robotics task, finding that more pressure for novelty means higher evolvability
- Similar prior findings by Doncieux *et al.* [8] with novelty search promoting evolvability



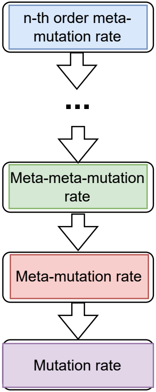
Principle 1: evolvability - future work

- Current research assumes that the variation (mutation) operators are static, and themselves exempt from the evolutionary process, which is not the case with biological evolution
- General idea: ideally, no operators are fixed, **everything evolves**!
- Rethinking evolvability...

> **We will focus on finding solutions that are not only evolvable in producing diverse offspring, but are also tied to mutation operators that promote long-term evolvability.**
> **$\rightarrow$ We aim to find evolvable solutions within the $\Theta \cup \Omega$ space, enhancing the evolutionary potential of the system.**

Principle 2: higher-order mutation rates

- We also investigate the use of higher-order mutation rates; while meta-mutation rate corresponds to meta-learning, higher-order mutation rates represent higher-order meta-learning
- Idea: mutation rate is not fixed, but its variance is controlled by a meta-mutation rate, which is in turn controlled by a meta-meta-mutation rate, etc. A tower.



Principle 2: higher-order mutation rates (cont.)

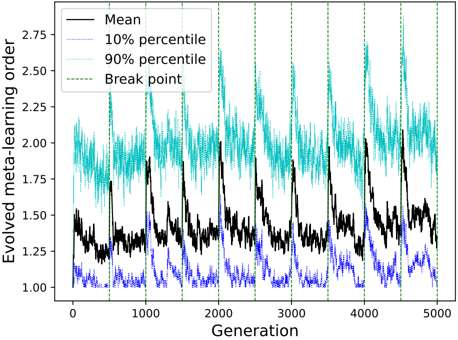
- In a Gaussian case:

$$
\begin{aligned}
\theta' &\sim \mathcal{N}\left(\theta, \sigma_1^2\right), \\
\sigma_i' &\sim \mathcal{N}\left(\sigma_i, \sigma_{i+1}^2\right), 1 \leq i < n, \\
\sigma_n' &\sim \mathcal{N}\left(\sigma_n, \sigma_{\text{meta}}^2\right),
\end{aligned}
\tag{3}
$$

where θ denotes the solution, σ_i the mutation rate of order i, σ_{meta} the fixed top meta-mutation rate, and $\mathcal{N}(\cdot, \cdot)$ the Gaussian mutation operator parametrized by the mean and variance. The order is given by n - the tower height.

Principle 2: higher-order mutation rates (cont.)

- We also study what happens if we let the meta-learning order itself evolve.
- Some preliminary results indicate that the mean meta-learning order in the order increases precisely when dynamic changes in the environment take place.
- The system adjusts the mean meta-learning order accordingly!



Conclusion (with further principles and ideas)

The exploration of different evolutionary principles for the development of more robust, high-performing, sample-efficient RL agents, especially in uncertain and dynamic environments - the essence of the FERLUDE project.

- Much remains to be investigated
  - Self-adaptivity in general: dynamic (evolving) evolutionary operators - co-evolution of agents, environments, and operators themselves
  - For example, evolving the amount of selective pressure, instead of setting it exogenously ("selecting for selection") [9]
  - New types of regularization (e.g. sparsity, binary mask overlaid over DNN weights)
  - Links between risk-aversion and exploration strategies

Bibliography

[1] Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, Lanctot M, Sifre L, Kumaran D, Graepel T, Lillicrap T. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science. 2018 Dec 7;362(6419):1140-4.

[2] Gu S, Holly E, Lillicrap T, Levine S. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In2017 IEEE international conference on robotics and automation (ICRA) 2017 May 29 (pp. 3389-3396). IEEE.

[3] Salimans T, Ho J, Chen X, Sidor S, Sutskever I. Evolution strategies as a scalable alternative to reinforcement learning. arXiv preprint arXiv:1703.03864. 2017 Mar 10.

[4] Such FP, Madhavan V, Conti E, Lehman J, Stanley KO, Clune J. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. arXiv preprint arXiv:1712.06567. 2017 Dec 18.

[5] Bai H, Cheng R, Jin Y. Evolutionary reinforcement learning: A survey. Intelligent Computing. 2023 May 10;2:0025.

[6] Hospedales T, Antoniou A, Micaelli P, Storkey A. Meta-learning in neural networks: A survey. IEEE transactions on pattern analysis and machine intelligence. 2021 May 11;44(9):5149-69.

[7] Gašperov B, Djurasević M. On evolvability and behavior landscapes in neuroevolutionary divergent search. In Proceedings of the Genetic and Evolutionary Computation Conference 2023 Jul 15 (pp. 1203-1211).

Bibliography (cont.)

[8] Doncieux S, Paolo G, Laflaquière A, Coninx A. Novelty search makes evolvability inevitable. InProceedings of the 2020 Genetic and Evolutionary Computation Conference 2020 Jun 25 (pp. 85-93).

[9] Frans K, Soros LB, Witkowski O. Selecting for Selection: Learning To Balance Adaptive and Diversifying Pressures in Evolutionary Search. arXiv preprint arXiv:2106.09153. 2021 Jun 16.

[10] Luong NC, Hoang DT, Gong S, Niyato D, Wang P, Liang YC, Kim DI. Applications of deep reinforcement learning in communications and networking: A survey. IEEE communications surveys & tutorials. 2019 May 14;21(4):3133-74.