Genetic algorithms for reinforcement learning problems

Autonomous and Adaptive systems project

Noemi Valentini

Introduction

- **Reinforcement learning (RL)**: an agent learns how to achieve a goal interacting in a certain environment. The agent improve itself through experience.
- **Evolutionary computation (EC)**: they search the best policy in a population, as in biology. Each individual interacts with the environment. The policies that obtain the most reward and their variations will be used for the next generation.
- **Environment**: Cart-pole environment
- **Semi-gradient Q-Learning (DQN)**: value approximation method. Estimation of Q action-value which directly approximates q_* without following its policy (decay epsilon-greedy policy).

$$w = w + \alpha\{R_{t+1} + \max_a \hat{q}(S_{t+1}, a, w) - \hat{q}(S_t, A_t, w)\} \nabla \hat{q}(S_t, A_t, w)$$

- **Advantage Actor Critic (A2C)**: policy approximation method that want to improve its policy (actor-network, θ) considering the state-value of another network (critic-network, w). The advantage is given from the difference between the estimation of the action value and the value function, it wants to consider how much an action is better than others on average.

  $\delta \leftarrow \hat{Q}(S, A, w) - \hat{v}(S, w)$

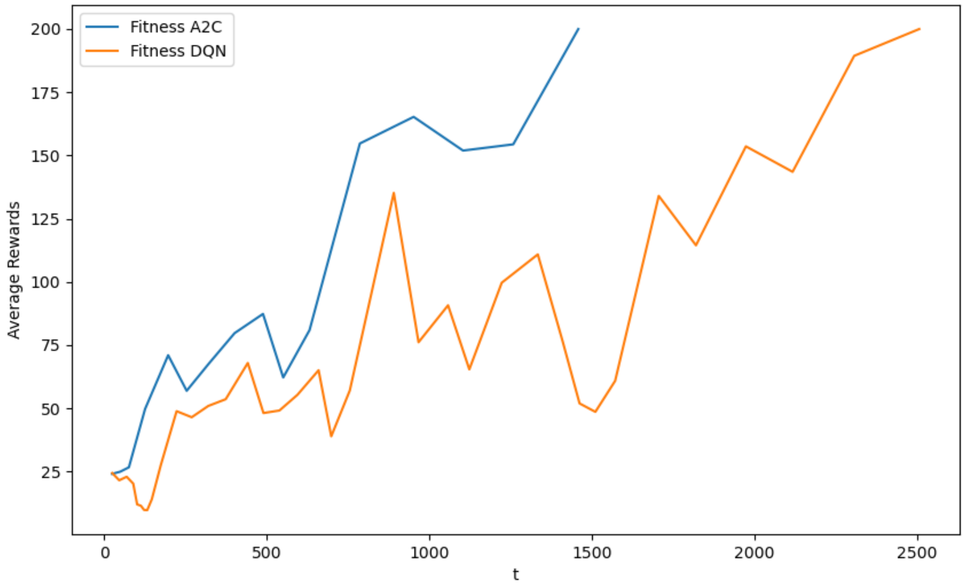
  $w \leftarrow w + \alpha_w w \delta \nabla_w \hat{v}(S, w)$

  $\theta \leftarrow \theta + \alpha_\theta \cdot \delta \nabla_\theta \ln \pi(A \mid S, \theta)$

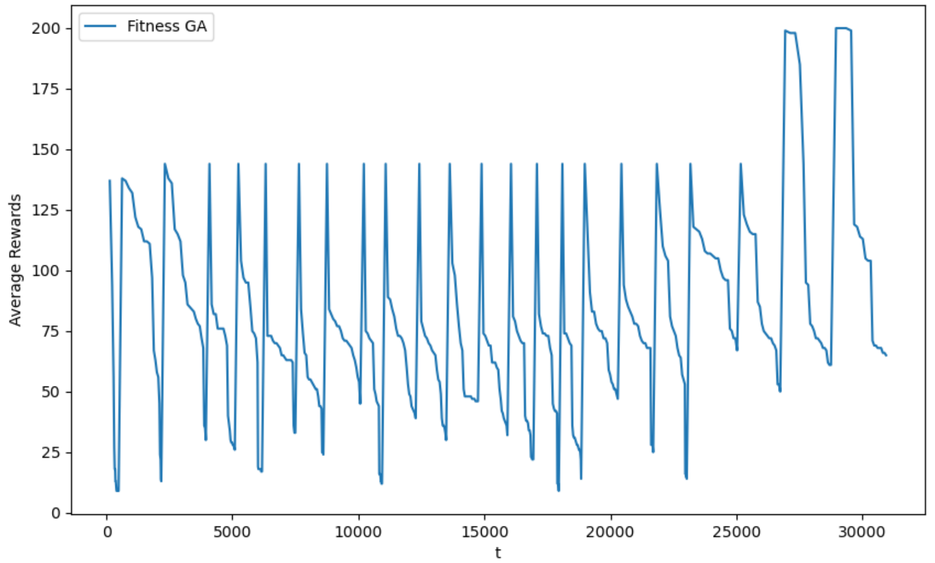
- **Base Genetic Algorithm (GA)**: a population P of N individuals (neural network parameter vectors θ). At each generation, **evaluation** of fitness (the mean of the rewards) $F(\theta_i)$, truncation **selection** (the top T individuals become the parents of the next generation), **mutation** (applying additive Gaussian noise, $\epsilon \sim \mathcal{N}(0,I)$), **elitism.**
  - Ga Encoding: $\theta_n = \psi(\theta_{n-1}, \tau_n) = \theta_{n-1} + \sigma\epsilon(\tau_n)$

Experiments

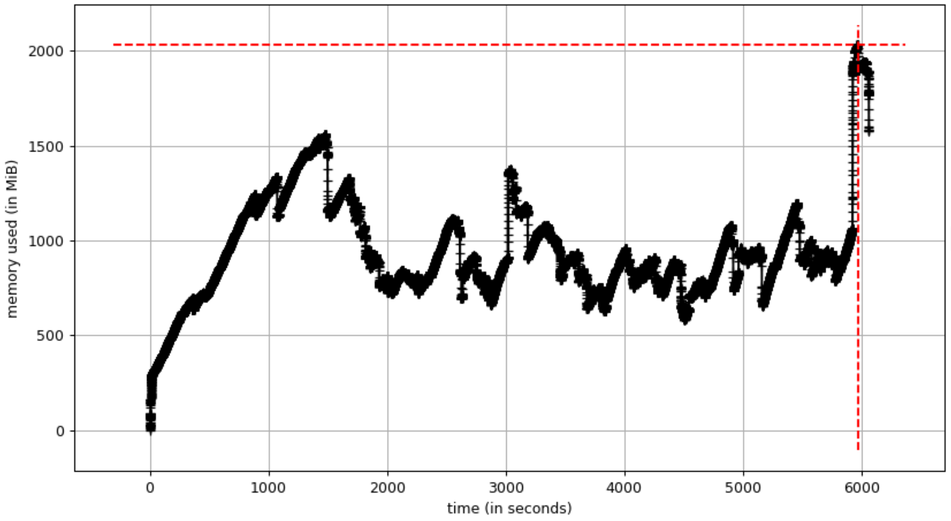
Reinforcement learning



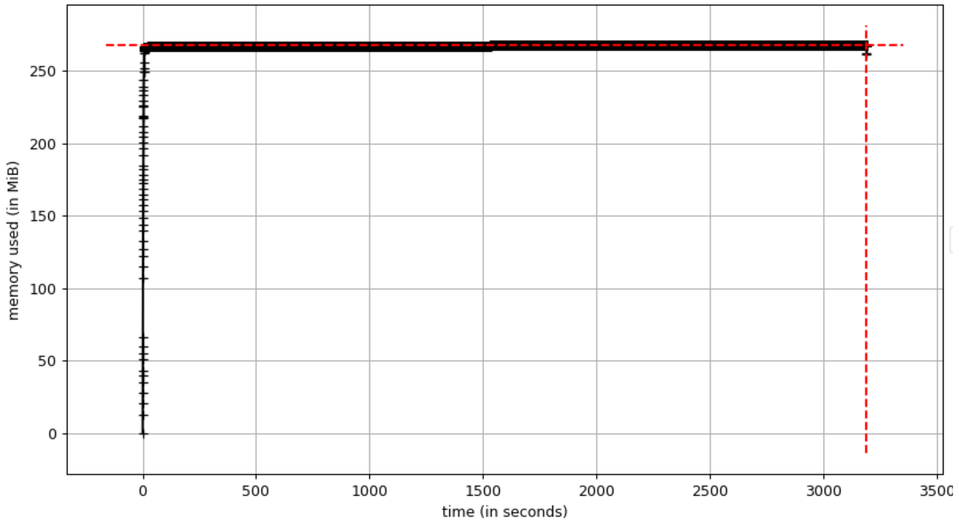
Evolutionary computation



DQN memory consumption



GA memory consumption



Discussion and conclusion

Differences between RL and GA

- Gradient based vs free gradient approach
- Spaces: on exploring the state-action space to discover optimal policies or value functions vs explore the solution space through candidate solution, to discover optimal or near-optimal candidate
- Exploration vs exploitation: leveraging known information for immediate rewards balanced by trying new actions for potential improvements; elitism and truncation selection balanced by mutation (and crossover) operators.

Future works

- Higher-dimensional environment: in the experiments, we considered a really simple problem, but it could be possible to analyse the behaviour of our agents in higher-dimensional problems.
- New improvements, using more sophisticated algorithmic techniques and exploiting parallelisation.
- Consider the possibility of using both approaches as a representation of human history — a collection of agents that learn throughout their lives and evolve as a species.

References

A. E. Eiben, J. E. Smith, et al. *Introduction to Evolutionary Computing*, volume 53. Springer, 2003.

John H. Holland. Genetic algorithms. Scientific American*, 1992.*

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. *Playing atari with deep reinforcement learning.* DeepMind Technologies, 2013.

Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. *Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning*

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, 1998.

Christopher J. C. H. Watkins. *Learning from delayed rewards.* 1989.