Projects Proposal:

Reinforcement Learning Sustainability Benchmark

Software Engineering for Artificial Intelligence
l.strefezza1@studenti.unisa.it

lab sesa SOFTWARE ENGINEERING SALERNO

Projects Scope

The project scope addresses the energy consumption of deep reinforcement learning (DRL) solutions and their impact on the environment and business costs. Beginning with the resurgence of the field following the development of Deep Q-Networks (DQN) by DeepMind in the early 2010s, there have been a number of algorithm proposals over time that with minor modifications to DQN or using a completely different paradigm (such as policy gradient methods) sought to improve the performance achieved by the learning agent.
Although the performance of the various solutions has been extensively studied and tracked, little effort has been directed toward understanding how the tweaks to the DQN introduced to improve performance impacted energy consumption, or what the cost of the alternative approaches developed was, per se and in comparison with previous solutions.
The project will delve into this aspect by trying to identify the trade-off between performance and energy consumption of some of the most widely used DRL algorithms, so that an interested company or individual can evaluate which solution to use based on the needs of the specific use case.

Starting Assets

(Works cited in the APA format)

- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction.* MIT press.
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602.*
- Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., ... & Silver, D. (2018, April). Rainbow: Combining improvements in deep reinforcement learning. *In Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1).
- Kostrikov, I., Yarats, D., & Fergus, R. (2020). Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649.*
- Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A., & Bachman, P. (2020). Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929.*
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347.*
- Fujimoto, S., Hoof, H., & Meger, D. (2018, July). Addressing function approximation error in actor-critic methods. In *International conference on machine learning* (pp. 1587-1596). PMLR.

Project Idea

The main idea of the project is to train various reinforcement learning algorithms on the same task to assess their energy consumption and performance. DRL algorithms can be devided in two main cathegories: value based (i.e. algorithms based on the approximation of a value function, be it the state-value function or the action-value function) and policy gradient. The latter are methods that approximate directly the policy, and includes as a special case the actor-critic methods, which approximate simultaneously a policy (said actor) and a value function (said critic).
In our benchmark we will consider, regardind the first cathegory, the DQN, which constitutes the first example of success of deep reinforcement learning (so that we have a sort of baseline), and RAINBOW, a method that involves a lot of the tweaks and improvement made to the original DQN. In addition to these two, we will test various of the single tweaks to assess their individual contribution to energy consumption and performance, and more advanced methods like SPR (Self-Predictive Representations, introduced in the fifth work cited).
Regarding policy gradient and actor critic methods, we will start with a basic one like REINFORCE and/or REINFORCE with baseline (chapter 13 of the first cited work) or the very similar Vanilla Policy Gradient (VPG). We will then move on to Proximal Policy Optimization (PPO), Deep Deterministic Policy Gradient (DDPG) and its evolutions Twin Delayed DDPG (TD3, seventh work cited) and DRQ (Data-regularized Q, fourth cited work).
Regarding the task on which to compare the algorithms, there are several suitable candidates: Atari 100k, one of the continuous control task of the DeepMind Control Suite, or one of the many other task (besides Atari) included in OpenAI Gymnasium (formerly Gym), and so on. We will probably opt for the Atari 100k, a discrete task that consists of playing one of the Atari games for 100000 interactions. The reason for this choice is that this is a widely used benchmark, well suited for running almost all popular DRL algorithms; however, we reserve the right to change the number of interactions (obviously keeping it consistent for all algorithms) or the task altogether, should hardware and time constraints dictate it.
The basic idea of this benchmark is to execute all the algorithms for the same number of environment interactions, so that we can compare the score they achieve and the energy consumption of each one of them. Another interesting comparison might be to take the score obtained by the lowest performer on the trial just described and re-train all the algorithm only until they reach that score, this way we can compare how much time and energy each one of them takes to achieve it.

Minimum Requirements

We can assume as a minimum requirement to meet in order to consider the project completed the carrying out the benchmark described in the previous sections, using Atari 100k, on at least five value-based algorithms (including the "baseline" DQN) and three policy gradient algorithms: one of the basics (REINFORCE, REINFORCE with baseline or VPG), DDPG and one of its evolutions (TD3 or DRQ).

Award Criteria

- Add Rainbow to the benchmark and all its constituent algorithms taken individually (seven in total, eight with rainbow, so three more value based methods than in the minimum requirements);
- add SPR to the benchmark, a more advanced value-based method;
- add the remaining gradient policy methods to the benchmark (so in addition to the basic, DDPG and one between TD3 and DRQ, the remaining between the two and PPO);
- add the benchmark described in the final part of the projcet idea, in which we compare all the algorithms the score being equal, instead of the number of interactions with the environment.