

# Reinforce-lib: A Reinforcement Learning Library for Scientific Research

Wednesday, 23 March 2022 16:30 (30 minutes)

Since the breakthrough achieved by the *DQN agent* in 2013 and 2015 on the *Atari learning environment* - a benchmark that was thought to be feasible only for humans - *Reinforcement Learning* (RL) and, especially, its combination with *Deep Learning* (DL), called *Deep Reinforcement Learning* (DRL), have both gained a major interest in the field since then, likewise *AlexNet* (thanks to the astounding improvement achieved on the ILSVRC challenge, compared to the best classical computer vision algorithm at that time) had started the deep learning era.

After few years, we have now powerful actor-critic distributed agents that are able to solve complex control and robotic tasks, and surprising even able to handle exponentially large search spaces like the ones found in the board games of Chess and Go, as well as high-dimensional continuous state-spaces of multi-agent environments. All these successes have common roots: powerful neural-networks function approximators borrowed from DL, and distributed training.

Although DRL seems to be incredibly powerful, in practice training successful agents is notoriously difficult, time consuming, resource intensive, costly, and error-prone: mainly due to very sensitive hyperparameters, beyond problem complexity itself. Such difficulties may arise from a still very limited understanding of the underlying mechanisms that power both RL and DRL, effectively preventing us to derive simpler (i.e. with way less moving parts, and hyperparameters) and thus more effective, sample-efficient, RL algorithms.

As happened in DL, having widely-used tools like *Keras* that simplify the building and training of neural networks is essential for speeding-up and improving research in that particular and related field(s). With such in mind, our aim is to provide a tool to ease the workflow of *defining, building, training, evaluating and debugging* DRL agents.

Our Python library **reinforce-lib** will provide simple code interfaces to a variety of implement agents and environments. We adopt a *modular design* that allows users to replace components like the agent's *networks*, its *policy*, and even *memory buffer* with other components made available by the library or new modules designed by the users themselves; this should enable the practitioner or researcher to easily prototype either novel research ideas, improvements to existing algorithms and components, or to adapt the agent for new, previously unsolved, research problems.

The reinforce-lib library other than being designed to be easy to use and extend, it will be also complete in the long-term (and, indeed, open-source), encompassing the three main paradigms found in reinforcement learning, namely: *model-free RL*, *model-based RL*, and *inverse RL*. This will allow users to solve a broader variety of problems according to the prior problem setting. For example, if the problem we want to solve naturally allow us to define a reward function, we will choose a model-free agent to solve it. If instead is easy for the researcher to provide a model of the environment (task or problem), model-based agents will do the job. Lastly, if we have many data coming from optimal sources (like domain experts, precise but expensive numerical simulations, exact algorithms, etc) we can leverage inverse RL algorithms to learn a reward function, and then use the learned reward to power the learning of model-free or hybrid agents.

The design principles of reinforce-lib, namely *usability*, *extensibility*, and *completeness*, should make the library distinguish itself from currently available RL libraries which are often based on *old* or legacy code, resulting in *unfriendly* code interfaces that are *hard to use* and even *harder to extend* or adapt. Mayor RL libraries like OpenAI's baselines, stable-baselines, Google's dopamine, and TensorFlow (to name a few) are also *very narrow*, often providing only few of the agents developed even by their own researchers.

We believe reinforcement learning to have many applications in scientific fields, thus further improving over classical or deep learning baselines, being even able to provide an answer to previously infeasible problems like the amazing breakthrough *AlphaFold* accomplished about *protein folding*. Today RL is mostly used in *games*, and *control systems* (often in simulation). Having the right tools, like the library we aim to develop with success, could help reinforcement learning to find applications in many more scientific scenarios.

**Primary author:** Mr ANZALONE, Luca (University of Bologna)

**Co-author:** Prof. BONACORSI, Daniele (University of Bologna)

**Presenter:** Mr ANZALONE, Luca (University of Bologna)

**Session Classification:** Artificial Intelligence

**Track Classification:** Track 10: Artificial Intelligence (AI)