The Optical RL-Gym: an open-source toolkit for applying reinforcement learning in optical networks

The Optical RL-Gym: an open-source toolkit for applying reinforcement learning in optical networks

Carlos Natalino and Paolo Monti
Electrical Engineering Department, Chalmers University of Technology, Gothenburg, Sweden
E-mail: carlos.natalino@chalmers.se, mpaolo@chalmers.se

ABSTRACT Reinforcement Learning (RL) is leading to important breakthroughs in several areas (e.g., self-driving vehicles, robotics, and network automation). Part of its success is due to the existence of toolkits (e.g., OpenAI Gym) to implement standard RL tasks. On the one hand, they allow for the quick implementation and testing of new ideas. On the other, these toolkits ensure easy reproducibility via quick and fair benchmarking. RL is also gaining traction in the optical networks research community, showing promising results while solving several use cases. However, there are many scenarios where the benefits of RL-based solutions remain still unclear. A possible reason for this is the steep learning curve required to tailor RL-based frameworks to each specific use case. This, in turn, might delay or even prevent the development of new ideas.

This paper introduces the Optical Network Reinforcement-Learning-Gym (Optical RL-Gym)[1], an open-source toolkit that can be used to apply RL to problems related to optical networks. The Optical RL-Gym follows the principles established by the OpenAI Gym, the *de-facto* standard for RL environments. Optical RL-Gym allows for the quick integration with existing RL agents, as well as the possibility to build upon several already available environments to implement and solve more elaborated use cases related to the optical networks research area. The capabilities and the benefits of the proposed toolkit are illustrated by using the Optical RL-Gym to solve two different service provisioning problems.

Keywords: Machine learning, autonomous network management, resource assignment, reinforcement learning environments

1. INTRODUCTION

In recent years, the increasing need from the industry for autonomous and cognitive network management stimulated the application of Machine Learning (ML) in various optical networking use cases. Supervised, semi-supervised, and unsupervised learning techniques have been successfully applied to solve problems related to Quality of Transmission (QoT) prediction, modulation format identification, and anomaly detection, just to mention a few examples [1], [2]. On the other hand, another category of ML algorithms, i.e., Reinforcement Learning (RL) techniques, has not been used extensively to solve used cases related to the performance optimization of optical networks. This is despite several interesting results that show the potentials of RL-based tools [3]–[6]. The latter is confirmed by looking at recent surveys of ML-based approaches applied to optical networks [1], [2] and to more general networking problems [7], which show that RL is used less frequently in comparison with other ML techniques.

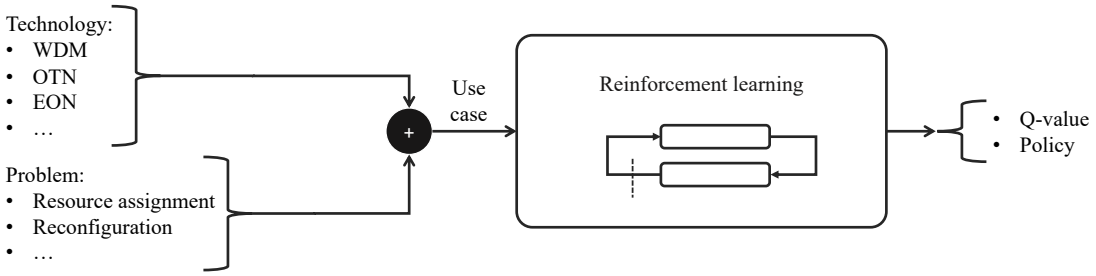


Figure 1: How a RL framework can be used to solve some of the most common use cases, e.g., Wavelength Division Multiplexing (WDM), Optical Transport Networks (OTN), or Elastic Optical Networks (EON).

Fig. 1 illustrates, in very general terms, how RL can be applied to solve some of the problems that can be encountered more frequently when working in the optical networks research field. The RL framework is applied to a use case (also referred to as *task* in RL specific terminology). The use case is the combination of a particular problem which should be solved for a specific technology choice. For instance, we might need to solve a resource assignment problem in Wavelength Division Multiplexing (WDM) optical networks, a use case commonly referred to as Routing and Wavelength Assignment (RWA). During the learning phase, the RL framework uses a loop where the learner (also known as *agent*) interact the *environment* (i.e., derived from the specific use case) to learn and optimize some long-term property or properties of the use case (also known as *Q-value*). Instead of a Q-value estimation method, a RL framework can also return, as a result, a stochastic

[1]The toolkit presented in this paper is available at: https://github.com/carlosnatalino/optical-rl-gym.

policy method. The RL framework described in general terms so far can be potentially deployed and used to make decisions in real-world networks.

The are several challenges in developing RL-based solutions to problems in optical networks. One of them is the difficulty in reproducing results due to the enormous number of parameters to be defined and set in the RL models. Even minor changes in the parameters or the implementation of the RL model might lead to major differences in results [8]. In the ML community, this has been mitigated in part by sharing the implementations, allowing for quicker adoption and validation by peers. Another barrier is the steep learning curve involved in applying RL models to solve optical network use cases. Differently from use cases in other research areas (e.g., games, robotics) which have many resources (i.e., open-source RL environments) readily available to be used, optical network practitioners need to implement their entire RL framework (i.e., environment, agent, action space and reward function) to get started. Implementing an entire RL framework from scratch requires not only knowledge on optical networks, but proficiency in the specifics of RL algorithms. For this reason, an optical network researcher interested in applying RL to his/her specific use case might greatly benefit from the presence of a RL framework already equipped with some basic environment implementations, a sort of starter tool kit that can be either used as-is or adapted to the researcher's specific needs.

This paper introduces the Optical RL-Gym, an open-source toolkit already equipped with a basic set of optical network use cases packaged as RL environments. The main aim of the Optical RL-Gym is to reduce the time and complexity to start applying RL models to the solution of optical network problems. The toolkit allows for the quick development of RL-ready optical network use cases by using the already provided environments or by extending the already provided functionalities. Once a specific use case is defined, the environment can be easily integrated into a standardized RL framework loop thanks to the use of the OpenAI Gym [9] interface. In the following sections, we first introduce the architecture of the Optical RL-Gym. Then, the toolkit functionalities are demonstrated considering two basic use cases, *(i)* routing and wavelength assignment with quality of service constraints in WDM networks, and *(ii)* routing, modulation, and spectrum assignment in Elastic Optical Networks (EONs). The results derived from the use cases' analysis show that thanks to the Optical RL-Gym, different RL models can be easily benchmarked highlighting which one is the most appropriate for a given use case.

2. The Optical RL-Gym Architecture

RL techniques are the subset of ML methods that study models that can (learn how to) take decisions over a given environment to maximize some notion of cumulative reward. A RL framework is composed of two components, i.e., the *Environment* and the *Agent*, whose interaction is illustrated in Fig. 2.

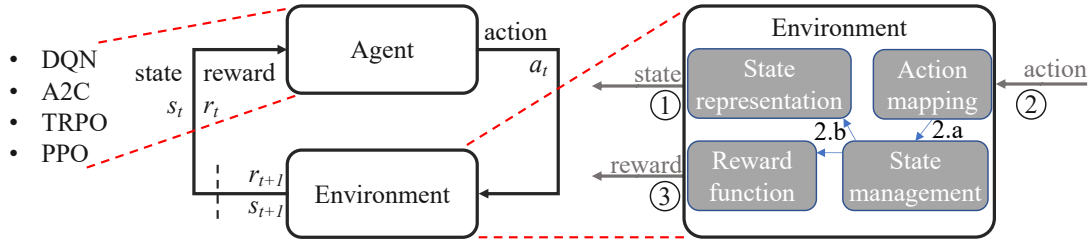


Figure 2: Main components of a RL framework and their interactions. On the right side a breakdown of the environment entities. On the left side a list of possible RL agent implementations.

The RL agent is responsible for observing the environment and taking actions to maximize the cumulative reward. In the literature, there are several models already available for the implementation of a RL agent, e.g., Q-Learning, Deep Q Network (DQN), synchronous Advantage Actor-Critic (A2C), Trust Region Policy Optimization (TRPO) and Proximal Policy Optimization (PPO).

The environment represents the use case to be learned by the agent. Usually, it is formalized in terms of a Partially Observable Markov Decision Process (POMDP). The environment comprises several entities with rules that govern their interactions and the reactions that each action has within the environment. More specifically, the environment interacts with the agent through three interfaces. The first one is the *state representation*, which exposes relevant information about the current state of the environment to the agent. At this stage, the agent analyzes the state and outputs the appropriate action to be taken. The second interface is the *action mapping*, which defines how each possible action may influence the environment, i.e., by changing its state. Finally, the *reward function* defines the value associated with the action selected. The value of the reward can be based either on the type of changes in the environment state caused by the action or by a value associated with the current environment state as a result of the action. Internally, the action mapping changes the environment state, which, in turn, affects both the reward function and the state representation of the next interaction with the agent.

The implementation of RL environments requires a deep knowledge of the specific use case of interest, which usually falls outside the expertise of RL researchers. The OpenAI Gym [9] filled this gap by establishing what is now the *de-facto* standard followed by several other projects, enabling RL researchers to quickly benchmark their models using a variety of use cases. However, there are no environments currently available for optical network use cases. Meanwhile, the implementation of RL agents requires deep knowledge of the specific model of interest, which usually falls outside the expertise of optical network practitioners. There are several frameworks focused on providing state-of-the-art RL implementations while not requiring deep knowledge about RL models, such as Stable Baselines [10], which extends the OpenAI Baselines [11]. These implementations assume that the RL environment is provided with an interface following the OpenAI Gym definition. Therefore, implementing a RL environment that follows the OpenAI Gym interfaces definition enables the use of a number of already available and validated implementations of RL agents, e.g., DQN [12], A2C [13], TRPO [14] and PPO [15].

The Optical RL-Gym toolkit is built following the principles established by the OpenAI Gym. As a result, the proposed toolkit provides a set of fully featured optical network environments for the quick setting up, training, and testing of different RL agent options. By following the software architecture established by the OpenAI Gym, the environments already provided in the Optical RL-Gym can be quickly extended starting from the (already provided) basic functionalities according to the properties of the specific use case to be studied.

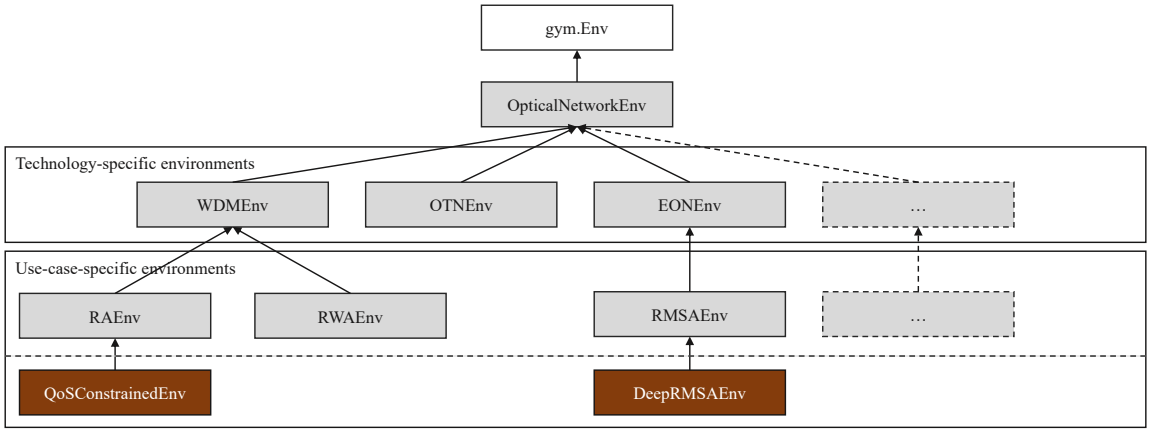


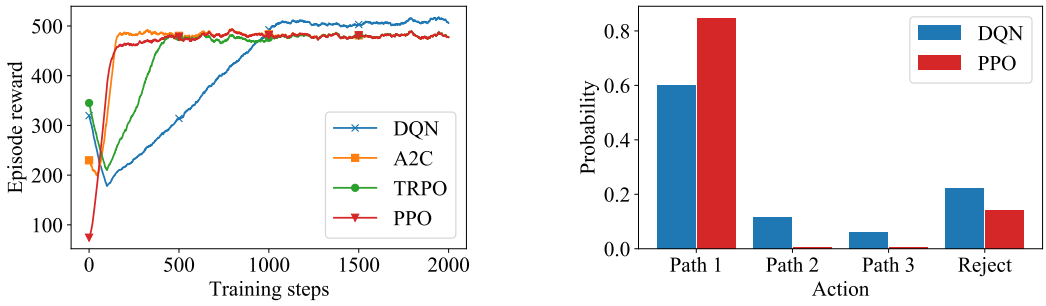
Figure 3: Hierarchy of the Optical RL-Gym environments.

Fig. 3 shows the hierarchy of the environments envisioned for the proposed Optical RL-Gym toolkit. The environments extend the methods defined by the *gym.Env* class. The Optical RL-Gym envisions four levels of environments. At the top level, the *OpticalNetworkEnv* environment defines the essential optical network components (e.g., network topology composed of nodes and links, with links composed of spectral resources) and implements elementary functionalities for optical networks (e.g., booking/releasing of resource). Technology-specific environments extend the functionalities of *OpticalNetworkEnv*, including specific properties of the technology, mainly characterized by the way spectral resources are managed. For instance, WDM optical networks (represented by the *WDMEnv* environment) divide their spectral resources into wavelengths that, in turn, can be used to created optical channels with fixed and pre-defined bandwidth. EONs (represented by the *EONEnv*environment), on the other hand, assume spectrum slots that can be grouped to establish optical channels with variable bandwidth values.

Use-case-specific environments extend the technology-specific ones by implementing the particular properties of the problem at hand. For instance, resource assignment problems in optical networks are usually modeled by random service request asking for the establishment of optical channels between two network nodes. In WDM optical networks, this problem is known as dynamic RWA. In contrast, in EONs this problem is known as dynamic Routing, Modulation format and Spectrum Assignment (RMSA). Some simplifications can be made, such as removing the spectrum continuity constraint while solving the dynamic RWA problem, suitable for opaque WDM optical networks, resulting in the Routing Assignment (RA) problem modeled by the *RAEnv* environment. Finally, more specific use cases can extend the functionalities implemented by the built-in Optical RL-Gym use cases. For instance, the *QoSConstrainedEnv* environment [4] implements a use case where each service request has specific Quality of Service (QoS) requirements and generate different revenue level when provisioned. Another example is the *DeepRMSAEnv* environment [5] that implements specific state representation, action mapping, and reward function for the dynamic RMSA use case. In the following section, both the *QoSConstrainedEnv* and the *DeepRMSAEnv* environments are used to illustrate the functionalities and the benefits of using the Optical RL-Gym.

3. USING THE OPTICAL RL-GYM

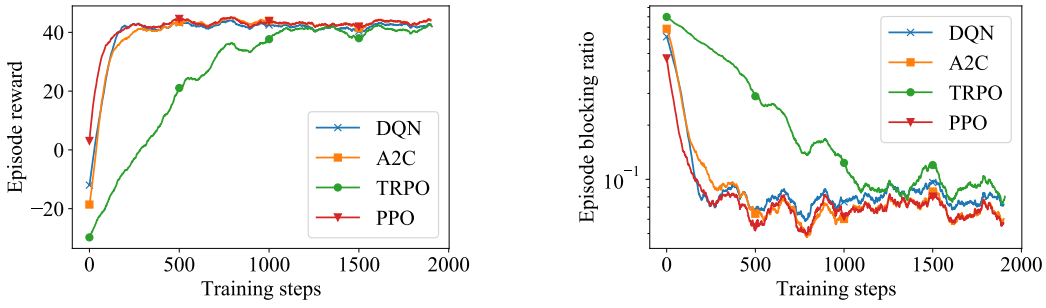
In this section, we demonstrate the flexibility and ease of use of the Optical RL-Gym by benchmarking two resource assignment use cases with four state-of-the-art RL agents taken from the literature. The results presented in this section were obtained using the RL agent implementations provided by the Stable Baselines [10]: DQN [12], A2C [13], TRPO [14] and PPO [15]. For all the agents, only two parameters were changed from their default value, i.e., the learning rate was set to 10^{-5}, and the neural network used has 4 layers with 150 neurons each. We are particularly interested in the RL agent performance in the initial stages of training. Therefore, we limit our evaluation to up to 2,000 training steps. The convergence of these models may take millions of training steps, and this analysis is left for future work.



(a) (b)

Figure 4: Training results usign different RL agents with the *QoSConstrainedEnv* environment.

Fig. 4 shows the results for the *QoSConstrainedEnv* environment, which implements the use case described in [4]. This use case considers two types of requests: the low-priority ones can use any path and result in a low reward; the high-priority ones only accept the shortest path but result in a high reward. A particular aspect of this environment is that the agent has the choice to select one of precomputed k-shortest-paths (k=3) or to proactively reject the request on purpose (i.e., even there are enough resources in the network to provision the request). We use a 6-nodes, 8-links topology with 16 connectivity resource units per link. The episode length is set to 100 (as opposed to 1,000 in [4]). Fig. 4 shows that A2C and PPO quickly reach reward near 500, while DQN takes longer to increase the reward, but achieves a higher reward from 1,000 training steps. Fig. 4b shows the probability across different actions, which explains the better performance of DQN (A2C and TRPO are not shown since they present similar probabilities values as PPO). While PPO uses mostly the shortest path (path 1), DQN uses more the second and third path option. This allows DQN to accommodate more QoS-constrained requests with an overall higher reward as a result. Moreover, we can observe that DQN rejects more requests than PPO, and, based on the achieved reward, we can assume that these rejections are for low-priority requests, making space for high-priority ones.



(a) (b)

Figure 5: Training results using different RL agents using the *DeepRMSAEnv* environment.

Fig. 5 shows some results related to the *DeepRMSAEnv* environment which implements the use case described in [5]. We use the NSFnet topology with the same modulation formats, traffic intensity, number of available spectrum blocks (J), and episode length as in [5]. Fig. 5a shows that the PPO model is the first one to reach an episode reward equal to 40 and that it maintains a slight advantage over the other models. Interestingly, the TRPO model takes much longer to reach the same reward level as the other models and it is the one with the worst performance. Fig. 5b illustrates the performance of the various agents in terms of blocking ratio. A high

reward obtained in Fig. 5a translates into a lower blocking ratio. PPO obtains a better blocking ratio at the beginning of the training, with A2C obtaining a similar performance after 1000 training steps. DQN and TRPO show a slightly higher blocking probability, which indicates that further training and hyperparameter tuning are necessary.

4. Final Remarks, Future Work and Open Challenges

This paper introduces the Optical RL-Gym, a toolkit that facilitates the use of RL-based methods to optimize the performance of optical networks. This is done by allowing an easy translation of different optical network use cases in terms of RL environments that can be easily integrated with already existing RL agents.

The smooth integration between the Optical RL-Gym and the RL agents is demonstrated by evaluating four state-of-the-art agent models over two optical networking use cases taken from the literature. We show that, depending on the use case, some models show better performance than others, confirming the benefits of evaluating different agents to identify the one with best performance.

In the future, we will focus on extending the functionalities of the Optical RL-Gym with relevant technologies and use-case-related environments. We also intend to create environment wrappers that will allow users to modify specific aspects of the environment without the need to extend the entire environment.

There are several interesting challenges in optical networks where RL has the potential to surpass the performance of currently available heuristics. For instance, existing RMSA solutions based on RL allow for very few options for the assignment of the spectral resources, e.g., most of the time they consider using only the first-fit algorithm. Allowing the RL agent to manage the spectral resources more intelligently might result in better blocking ratio results. Following the same reasoning, it will be possible to incorporate in the RL framework also other performance metrics, i.e., in addition to considering only the blocking ratio as it has been the case so far. Moreover, existing works only assess the performance of RL agents against a pre-defined and constant reward function. At the same time, one of the crucial advantages of RL is the possibility to dynamically adapt to changes in the value associated with an action. As an example, the introduction of new services while a network is in operation might change the value of the revenue functions thus requiring the resource assignment algorithm to adapt dynamically to the new network conditions. With RL this is possible by changing in the reward function, thus allowing the network management system to react dynamically to the new environment.

Acknowledgments

This work was supported by the Celtic-Plus sub-project AI-NET-ANIARA funded by VINNOVA.

References

[1] J. Mata *et al.*, "Artificial intelligence (AI) methods in optical networks: A comprehensive survey," *Optical Switching and Networking*, vol. 28, pp. 43 – 57, 2018, DOI: 10.1016/j.osn.2017.12.006.

[2] F. Musumeci *et al.*, "An overview on application of machine learning techniques in optical networks," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1383–1408, 2019, DOI: 10.1109/COMST.2018.2880039.

[3] Y. Pointurier and F. Heidari, "Reinforcement learning based routing in all-optical networks," in *Int. Conf. on Broadband Communications, Networks and Systems (BROADNETS)*, Sep. 2007, pp. 919–921, DOI: 10.1109/BROADNETS.2007.4550533.

[4] C. Natalino *et al.*, "Machine-learning-based routing of QoS-constrained connectivity services in optical transport networks," in *OSA Networks*, 2018, p. NeW3F.5, DOI: 10.1364/NETWORKS.2018.NeW3F.5.

[5] X. Chen *et al.*, "DeepRMSA: A deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks," *J. Lightw. Technol.*, vol. 37, no. 16, pp. 4155–4163, Aug 2019, DOI: 10.1109/JLT.2019.2923615.

[6] J. Suárez-Varela *et al.*, "Routing in optical transport networks with deep reinforcement learning," *J. Opt. Commun. Netw.*, vol. 11, no. 11, pp. 547–558, Nov 2019, DOI: 10.1364/JOCN.11.000547.

[7] N. C. Luong *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 2019, DOI: 10.1109/COMST.2019.2916583.

[8] P. Henderson *et al.*, "Deep reinforcement learning that matters," in *AAAI Conf. on Artificial Intelligence*, 2018. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16669

[9] G. Brockman *et al.*, "OpenAI Gym," 2016, arXiv: 1606.01540.

[10] A. Hill *et al.*, "Stable baselines," https://github.com/hill-a/stable-baselines, 2018.

[11] P. Dhariwal *et al.*, "OpenAI baselines," https://github.com/openai/baselines, 2017.

[12] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," 2013, arXiv: 1312.5602.

[13] ——, "Asynchronous methods for deep reinforcement learning," 2016, arXiv: 1602.01783.

[14] J. Schulman *et al.*, "Trust region policy optimization," 2015, arXiv: 1502.05477.

[15] ——, "Proximal policy optimization algorithms," 2017, arXiv: 1707.06347.