Implementation of RL Algorithms in OpenAI Gym

Douglas Trajano, Sirleno Vidaletti
Pontifical Catholic University of Rio Grande do Sul - PUCRS
School of Technology. Porto Alegre, Brazil
{douglas.trajano, sirleno.vidaletti}@edu.pucrs.br

Abstract

LunarLander is an electronic game created in 2013. The objective in the game is to earn as many points as possible by controlling a bird, without letting it crash into the pipes. In this work, reinforcement learning techniques will be used to automatically divert the yellow bird between the pipes. The OpenAI Gym toolkit will be used. OpenAI Gym is a toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games.

Introduction

Reinforcement learning (RL) is a general framework for adaptive control, which has proven to be efficient in many domains, e.g., board games, video games or autonomous vehicles. In such problems, an agent faces a sequential decision-making problem where, at every time step, it observes its state, performs an action, receives a reward and moves to a new state.(Buffet, Dutech, and Charpillet 2007)

RL agents interacts with an environment by selecting actions and receiving rewards. The environment is typically a simulator that provides a state and a reward for each action taken. The RL agent can learn from its experience and improve its decision-making ability. In this work, we will use the LunarLander environment provided by OpenAI Gym.

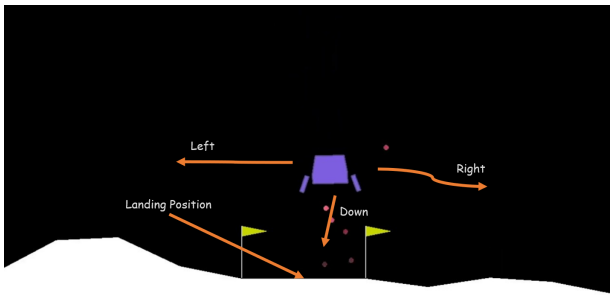

Figure 1: LunarLander-v2 - OpenAI Gym

The basic idea behind many reinforcement learning algorithms is to estimate the action-value function. The goal of the agent (powered by a reinforcement learning algorithm) is to interact with the emulator by selecting actions in a way that maximises future rewards.(Mnih et al. 2013)

In this work, we will develop some reinforcement learning algorithms that will be used to train our agents in the OpenAI Gym environment. OpenAI Gym is a toolkit for reinforcement learning research. It includes a growing collection of benchmark problems that expose a common interface.(Brockman et al. 2016)

Technical Approach

Our technical approach consists of two parts. The first part is the definition of the environment. The environment that will be explored in this project is provided by OpenAI Gym. The second part is the reinforcement learning algorithm, which we will be develop by ourselves.

Environment

The environment is a Python class that simulates the game and returns the observations and rewards. It uses the OpenAI Gym API.

The state observation is a tuple with 8 elements representing the spaceship position, velocity, lander angle and angular velocity, left and right landing marks.

The available actions are:

- do nothing
- fire left orientation engine
- fire main engine
- fire right orientation engine

RL Algorithms

The RL algorithms will be implemented in Python. The algorithms are responsible for deciding the actions of the agent. We will develop a base class with a random policy, which will be used to define the API between the environment and the algorithms. We want to develop at least two different RL algorithms:

Q-Learning is an off-policy TD control algorithm. The learned action-value function, Q, directly approximates q_*, the optimal action-value function, independent of the policy being followed.(Sutton and Barto 2018)

Deep Q-Network (DQN) combines Q-Learning with deep neural networks to let RL work for complex, high-dimensional environments, like video games, or robotics. A critical component of DQN-style algorithms is memory buffer known as experience replay, it holds the most recent transitions collected by the policy.(Fedus et al. 2020)

Two different approaches of the experience replay will be develop and tested.

- **Experience Replay**: The most basic sampling strategy, it uses uniform sampling, whereby each transition in the buffer is sampled with equal probability.(Fedus et al. 2020)
- **Prioritized Experience Replay**: Extends experience replay function by learning to replay memories where the real reward significantly diverges from the expected reward, letting the agent adjust itself in response to developing incorrect assumptions.(Schaul et al. 2015)

At the end of the project, we will compare the performance of the developed algorithms and buffers.

Project Management

We want to validate some reinforcement learning algorithms in the Lunar Lander OpenAI Gym environment. We will work under the agile methodology of Scrum, so in each sprint we will have a planning session to define the scope of the work, and a sprint review to validate the work. The deadline for submitting papers is Dec 7, 2021. Below you can see a table with the expected sprints and the start and end dates.

Sprint	Start	End
1	Oct 25, 2021	Oct 31, 2021
2	Nov 1st, 2021	Nov 7, 2021
3	Nov 8, 2021	Nov 14, 2021
4	Nov 15, 2021	Nov 21, 2021

An estimate of tasks is provided below

- **Sprint 1**: Validate OpenAI environment and develop Deep Reinforcement Learning algorithm.
- **Sprint 2**: Develop a Temporal-Difference algorithm.
- **Sprint 3**: Evaluate the results.
- **Sprint 4**: Revision and paper development.

Conclusion

In this work, we will develop reinforcement learning algorithms and compare them with each other. We will use the OpenAI Gym environment to train our RL agents. We believe that the algorithms will be able to learn how to act in the proposed environment. At the end of the project, we will compare the performance of each algorithm.

References

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. Openai gym.

Buffet, O.; Dutech, A.; and Charpillet, F. 2007. Shaping multi-agent systems with gradient reinforcement learning. *Autonomous Agents and Multi-Agent Systems* 15(2):197–220.

Fedus, W.; Ramachandran, P.; Agarwal, R.; Bengio, Y.; Larochelle, H.; Rowland, M.; and Dabney, W. 2020. Revisiting fundamentals of experience replay. In *International Conference on Machine Learning*, 3061–3071. PMLR.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Schaul, T.; Quan, J.; Antonoglou, I.; and Silver, D. 2015. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

Sutton, R. S., and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.