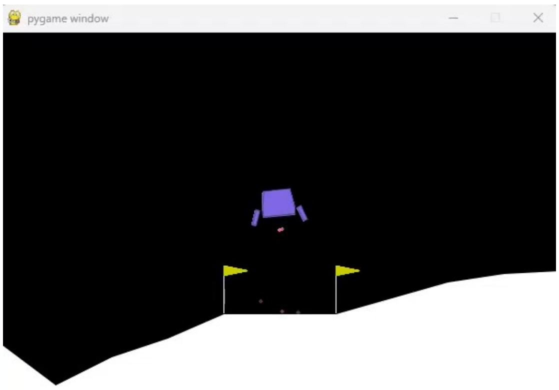
Project for the AAS course – DQN Lunar Lander

Matteo Rossi Reich - matteo.rossireich@studio.unibo.it

July 2023



Abstract

The aim of this project is to train an agent able to solve the Lunar Lander environment provided by Gymnasium with the best results.

1 Introduction

1.1 The environment

The environment is part of the Box2D environments contained in the gymnasium project. It consists of a simple 2D physics simulation, where a lander spacecraft has to land between two flags using its three engines.
The action space is discrete and consists of four actions:

- 0: do nothing
- 1: fire left orientation engine
- 2: fire main engine
- 3: fire right orientation engine

The observation space is described in the project page as follows ”The state is an 8-dimensional vector: the coordinates of the lander in x & y, its linear velocities in x & y, its angle, its angular velocity, and two booleans that represent whether each leg is in contact with the ground or not.”

1.2 The Reward

The reward function is predefined and described as follows:

- is increased/decreased the closer/further the lander is to the landing pad.
- is increased/decreased the slower/faster the lander is moving.

- is decreased the more the lander is tilted (angle not horizontal).
- is increased by 10 points for each leg that is in contact with the ground.
- is decreased by 0.03 points each frame a side engine is firing.
- is decreased by 0.3 points each frame the main engine is firing.

Moreover at the end of the episode an additional reward of -100 or +100 points for crashing or landing safely respectively.

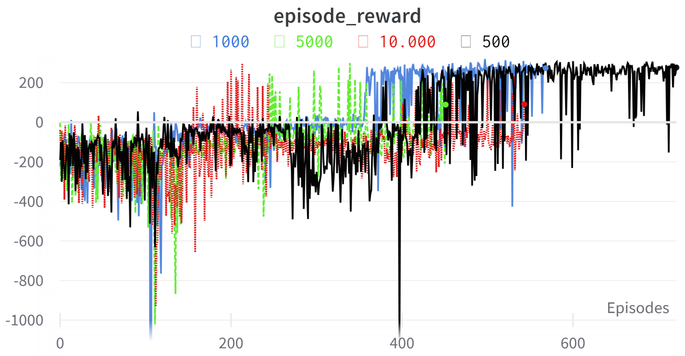
2 DQN

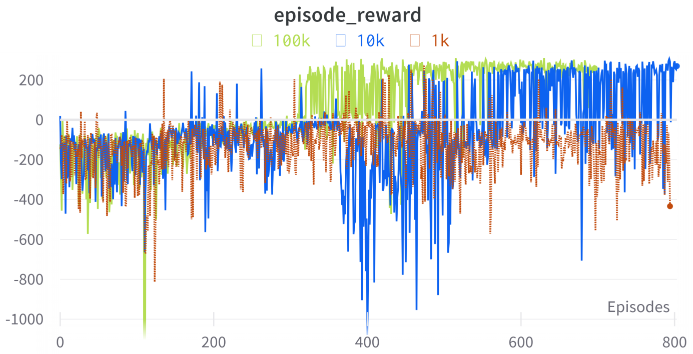
Deep Q-learning is implementing following the paper from Deep Mind. Both experience replay to break the correlation between consecutive samples and improve the learning efficiency and target network to provide more stable targets are used.

2.1 Hyper-parameter Search

In order to track and save the results of all runs and visualize progress Weight and Biases was used. It is which allows to track ML projects very conveniently and allows to perform hyper-parameter searches as well with W&B Sweeps. For this project multiple runs where performed to find the best combination of hyper-parameters, the size of the experience replay, the sampling starting step, the learning rate and the number of steps between the synchronization of the two networks were controlled. The number of steps between synchronization of the networks is the parameter which had the biggest impact on the learning process. The following graph illustrates it well, showing how syncing each 1000 seems to be the right choice.

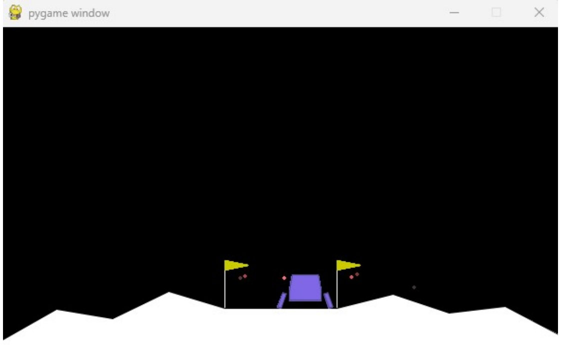
Moreover also the size of the experience replay has a noticeable effect on the learning process.





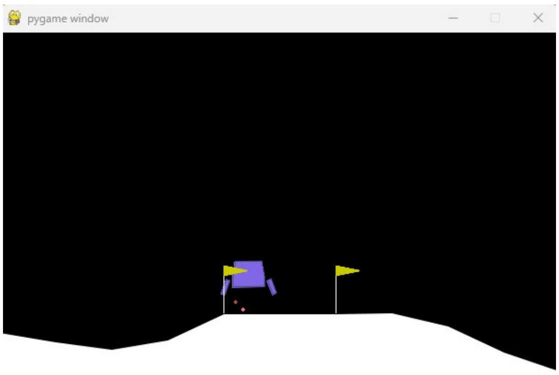
2.2 Reward Shaping

Once the agent had been trained I realized that while it did achieve a good score it had one problem, it wasn't turning off the engines after landing. I don't know why it was showing such behaviour, but I decided to modify the reward function to make it stop polluting the moon by spacecraft idling. As it is possible to see it is still firing both the right and left engine even though it reached its goal already. So I modified the reward function to give a negative reward when firing the engines with both legs touching the ground. The following is the reward I have added: state[6] and state[7] are booleans indicating contact with the ground, while state[4] is the angle of the spacecraft relative to the ground. The area between the two flags is flat, however before taking into consideration the angle I have seen the spacecraft land with a leg slightly off-targept and then slide on the side without turining on the engines.



```
    reward -= (
        m_power * state[6] * state[7] * 2 * (1 if abs(state[4]) < 0.1 else 0)
    )
    reward -= s_power * state[6] * state[7] * 2 * (1 if abs(state[4]) < 0.1 else 0)
```

The multiplicative factor **2** is chosen arbitrarily, however its is worth noting that giving a value to high (e.g. 30) it detrimental because the agent can literally get stuck inches from the ground in "fear" of the negative reward. The following image is a frame from an uncompleted attempt where the lander never made it to the ground.



3 Conclusion

The algorithm wasn't too hard to implement, it was however very challenging to find the right combination of hyper-parameters.