Reinforcement Learning TP3 report

Juliette Jacquot, Matis Braun

Introduction

In this report, we teach some reinforcement learning models to play to the game "Taxi-v3" from OpenAI Gym.

This game contains a 5 by 5 grid, a taxi a passenger, and a building. The taxi can move in four directions (up, down, left, right), pick up a passenger on its current spot, or drop a passenger off on its current spot. The goal of the game is to pick up the passenger and bring them to the building as fast as possible.

Each model is trained over a thousand iterations of the game, and is allowed to choose up to 200 actions to execute in the span of a game.

Models

QLearning

ε-greedy

Surprisingly, as seen in figure 1, trying to add exploration to the QLearning algorithm is detrimental to the model's performance. Perhaps the size of the game environment and the constant negative rewards applied until the end of the game prevent the model from finding a local maximum reward, making the added exploration unnecessary.

On the other hand, the learning rate does not seem to have much of an influence on the final result, as seen in figure 2. Indeed, the final reward seems to stabilize around a singular value as soon as $\alpha \geq 0.2$.

As seen in figures 3 and 4, setting the parameter ε at 0 stabilizes the rewards almost immediately once the model has an idea of how to play the game.

ε scheduling

In the ε scheduling version of the QLearning algorithm, the chance of a random choice occurring in the model decreases as time goes on. Still, as seen in figure 5, the model gets better results when no random choice occurs at all. Similarly, the learning rate behaves the same way as in the ε-greedy variant of the model, as seen in figure 6. While the difference in stability between the model with dummy parameters (figure 7) and the one with optimized parameters (figure 8) is reduced, there is still a noticeable improvement when $\varepsilon = 0$.

SARSA

Unlike the QLearning algorithm, SARSA does not use a random chance to explore. However, figure 9 shows the same stabilization of rewards depending on α than the other algorithms. As such, there is not an easily perceived difference between the base model (figure 10) and the optimized one (figure 11).

Conclusion

The optimized versions of each algorithm seem to behave similarly, both in their learning and in their final results as seen in figure 12.

Appendix

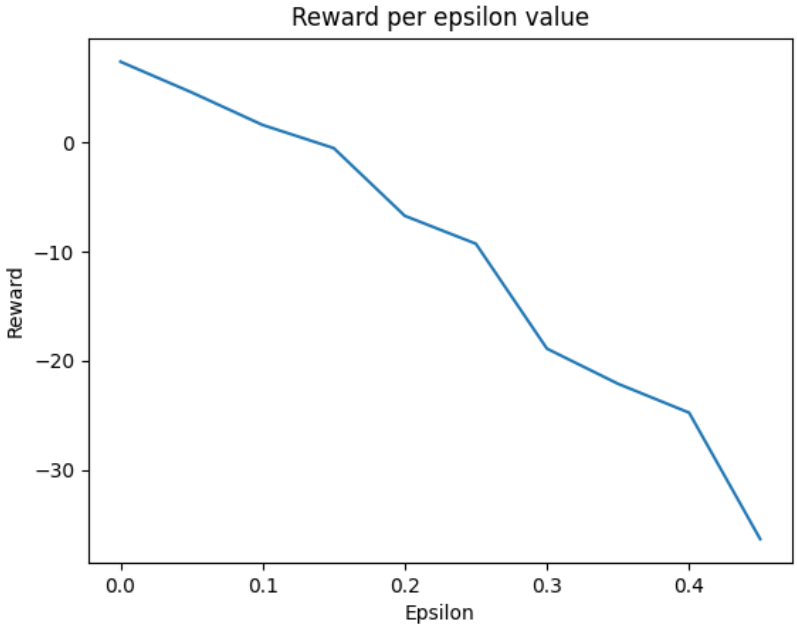

Figure 1: Average final rewards for QLearning per ε ($\alpha = 0.5$)

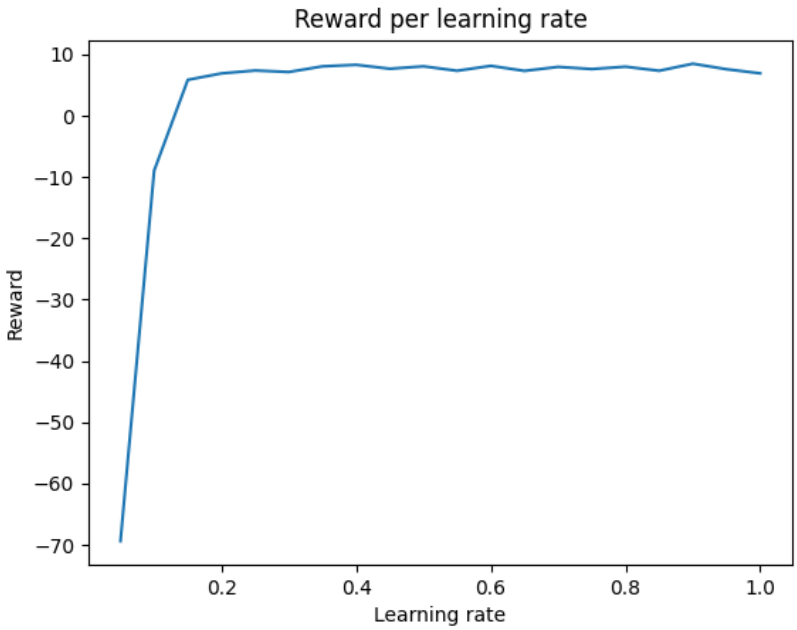

Figure 2: Average final rewards for QLearning per α ($\varepsilon = 0$)

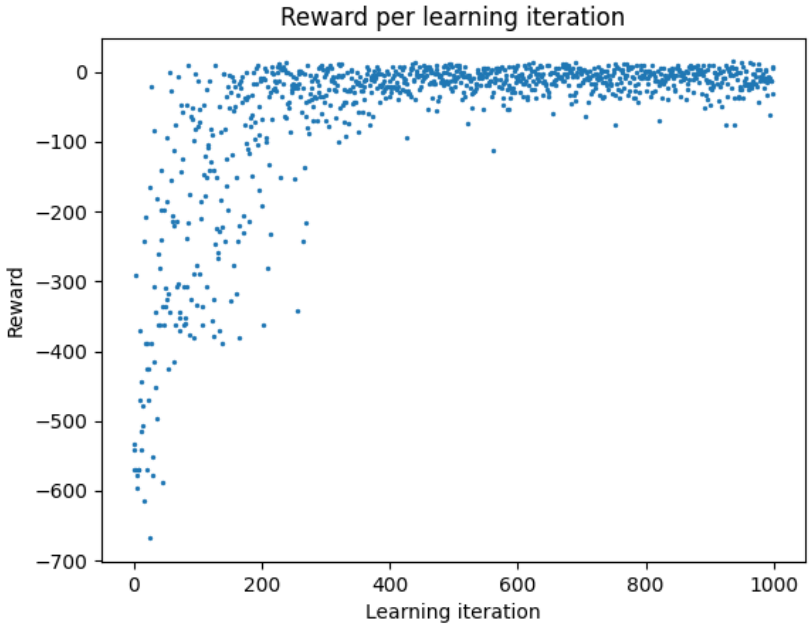

Figure 3: QLearning rewards during learning process ($\alpha = 0.5$, $\varepsilon = 0.25$)

Figure 4: QLearning rewards during learning process ($\alpha = 0.9$, $\varepsilon = 0$)

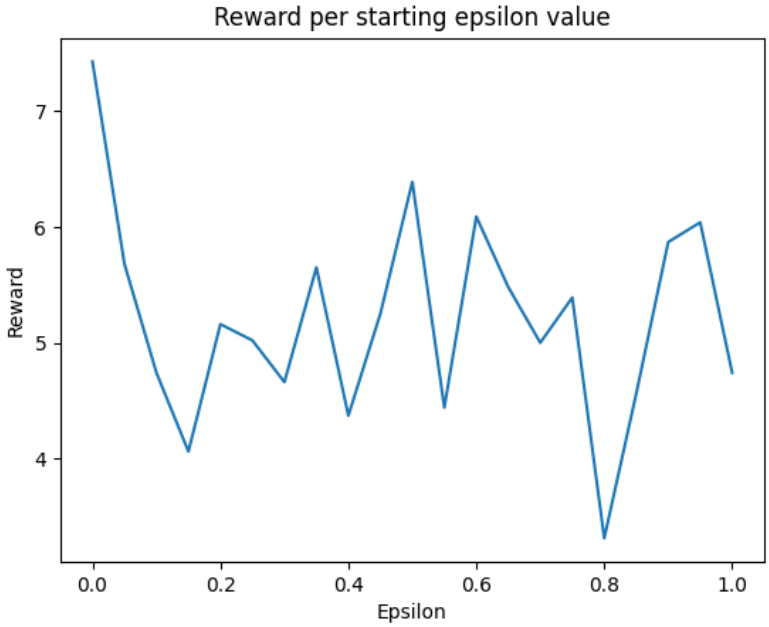

Figure 5: Average final rewards for QLearning with ε scheduling per $\varepsilon_{\text{start}}(\alpha = 0.5)$

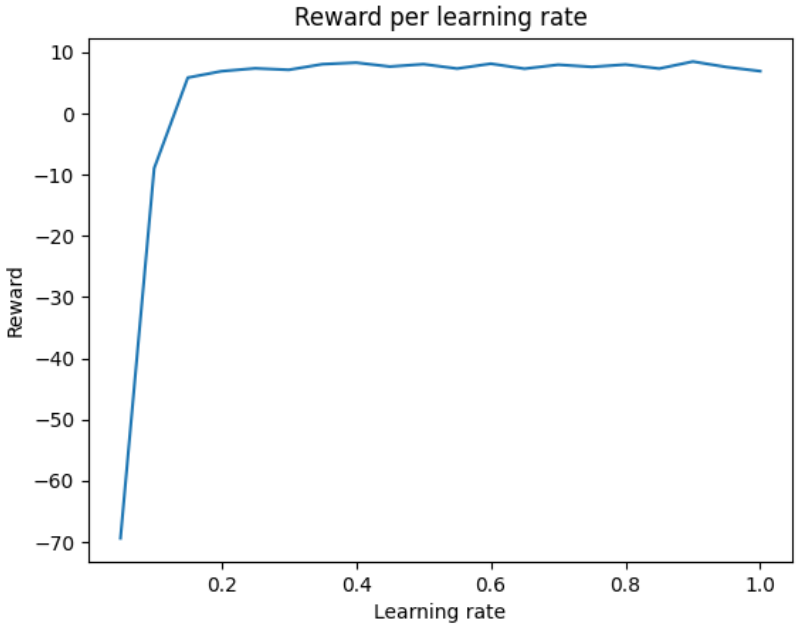

Figure 6: Average final rewards for QLearning with ε scheduling per $\alpha(\varepsilon = 0)$

Figure 7: QLearning with ε scheduling rewards during learning process ($\alpha = 0.5$, $\varepsilon_{\text{start}} = 0.25$)

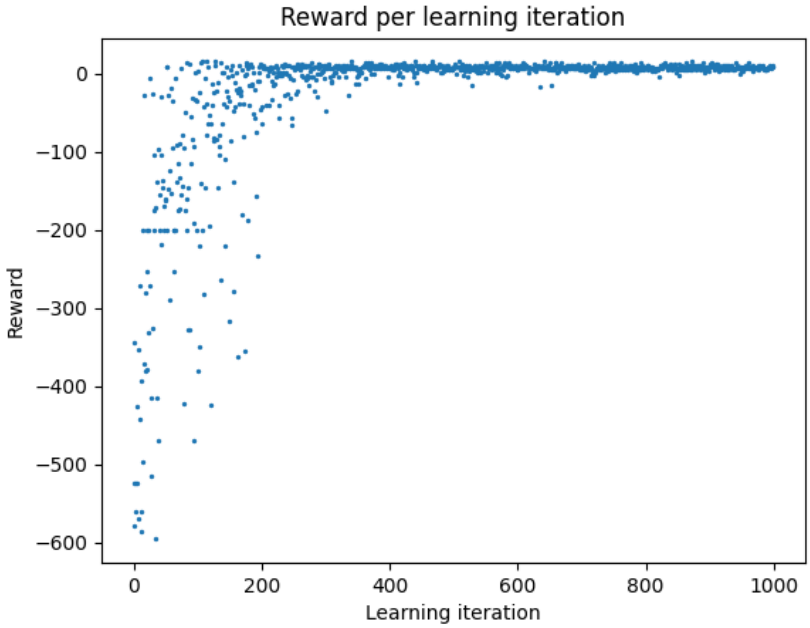

Figure 8: QLearning with ε scheduling rewards during learning process ($\alpha = 0.55$, $\varepsilon_{\text{start}} = 0$)

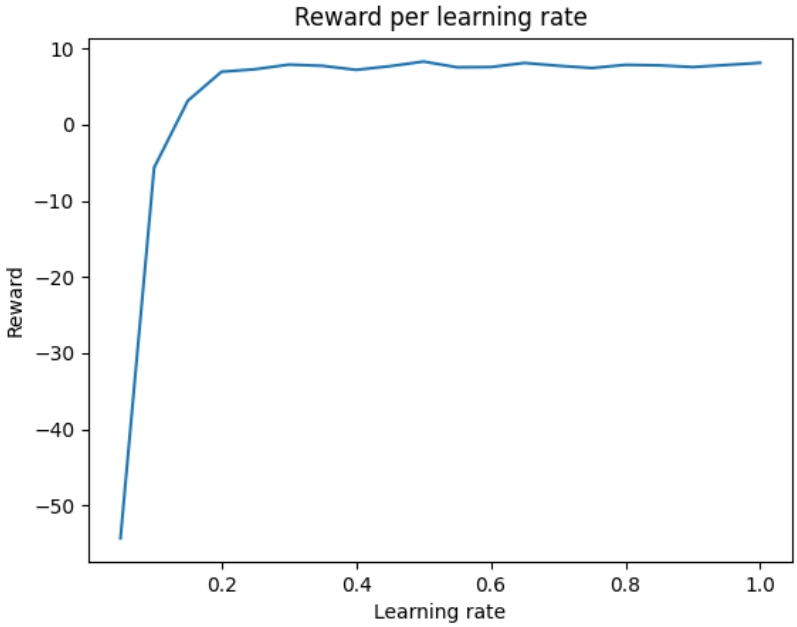

Figure 9: Average final rewards for Sarsa per α

Figure 10: Sarsa rewards during learning process ($\alpha = 0.5$)

Figure 11: Sarsa rewards during learning process ($\alpha = 0.55$)

	ε-greedy QLearning	QLearning with ε scheduling	Sarsa
Reward	7.78	7.69	7.49

Figure 12: Final rewards for all models