Rapport sur l'implémentation d'agents d'apprentissage par renforcement pour le jeu Taxi-v3

Jason Perez
Samy Hadj-said

20 octobre 2024

1 Introduction

Ce rapport présente l'implémentation et les résultats de trois agents d'apprentissage par renforcement pour le jeu *Taxi-v3* de OpenAI Gym : **Q-Learning**, **Q-Learning avec planification d'epsilon**, et **SARSA**. L'objectif est de comparer l'efficacité des algorithmes en termes de temps d'apprentissage et de performance, tout en analysant les logs pour suivre les évolutions de performances.

2 Choix d'implémentation

2.1 Q-Learning et SARSA

Les deux algorithmes ont été implémentés avec les caractéristiques suivantes :

- **Utilisation d'un dictionnaire** pour stocker les Q-valeurs, permettant une mise à jour et un accès efficaces.
- **Normalisation des récompenses** (division par 20) pour stabiliser l'apprentissage et éviter des mises à jour excessives.
- **Clipping de l'erreur TD** entre -1 et 1 pour éviter des changements brusques dans les Q-valeurs.
- **Stratégie d'exploration epsilon-greedy** avec une décroissance exponentielle pour le Q-Learning standard, favorisant l'exploration au début et l'exploitation à la fin.
- Pour SARSA, une **décroissance plus lente de l'epsilon** pour maintenir un certain niveau d'exploration plus longtemps.

2.2 Q-Learning avec planification d'epsilon

Cet agent étend le Q-Learning standard avec :

— Une **décroissance linéaire de l'epsilon** sur un nombre fixe d'étapes, offrant un contrôle plus précis de la transition entre exploration et exploitation.
— Un **epsilon minimal** pour maintenir une exploration résiduelle même après la période de décroissance.

2.3 Optimisations communes

Pour tous les agents :
— **Réinitialisation de l'agent** en cas de stagnation des performances, permettant de sortir des optima locaux.
— Utilisation de **moyennes mobiles** pour le suivi des performances, offrant une vue plus stable de l'évolution de l'apprentissage.

3 Résultats et analyse des logs

Les logs obtenus lors de l'exécution des agents montrent l'évolution des récompenses moyennes, l'amélioration continue des performances et les situations de stagnation des agents. Voici une synthèse des performances observées à partir des logs :

3.1 Performance de l'agent Q-Learning

— Initialement, l'agent Q-Learning avait une récompense moyenne de -1955 à l'épisode 0.
— À l'épisode 100, la récompense moyenne s'améliore à -539, indiquant une progression continue.
— Après plusieurs épisodes d'entraînement, la récompense moyenne atteint 0.92 à l'épisode 1000, marquant une convergence satisfaisante des performances.

3.2 Performance de l'agent Q-Learning avec planification d'epsilon

— La récompense moyenne de départ était de -3769 à l'épisode 0, reflétant les explorations initiales.
— À l'épisode 100, l'agent améliore sa récompense moyenne à -1674.
— Finalement, après l'épisode 1000, la récompense moyenne atteint 2.13, démontrant l'efficacité de la planification de l'epsilon dans la stratégie d'exploration.

3.3 Performance de l'agent SARSA

— L'agent SARSA débute avec une récompense moyenne de -3132 à l'épisode 0.

— Au fil des épisodes, les récompenses moyennes augmentent, atteignant -6.21 à l'épisode 900.
— À l'épisode 1000, une récompense moyenne de 0.28 indique une stabilisation des performances et une convergence vers une politique efficace.

4 Résultats visuels

Les graphiques ci-dessous montrent l'évolution des récompenses totales pour chaque agent sur 1 000 épisodes :

FIGURE 1 – Comparaison des performances des agents

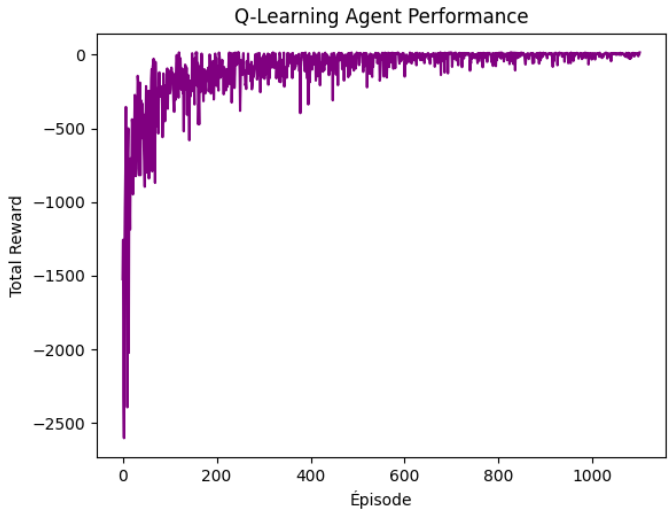

FIGURE 2 – Performance de l'agent Q-Learning

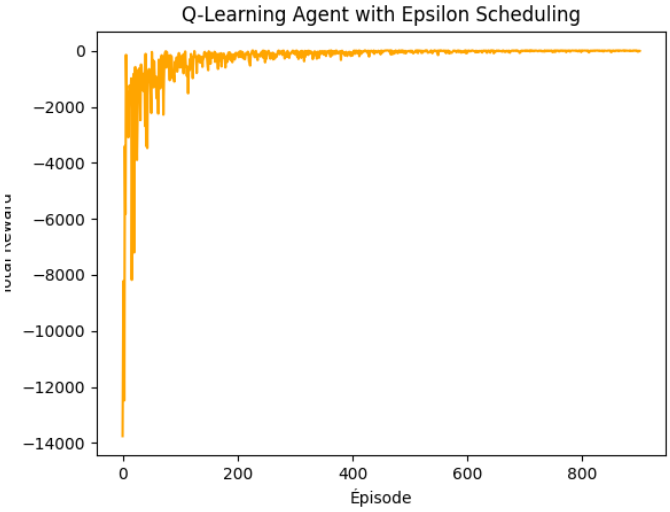

FIGURE 3 – Performance de l'agent Q-Learning avec planification d'epsilon

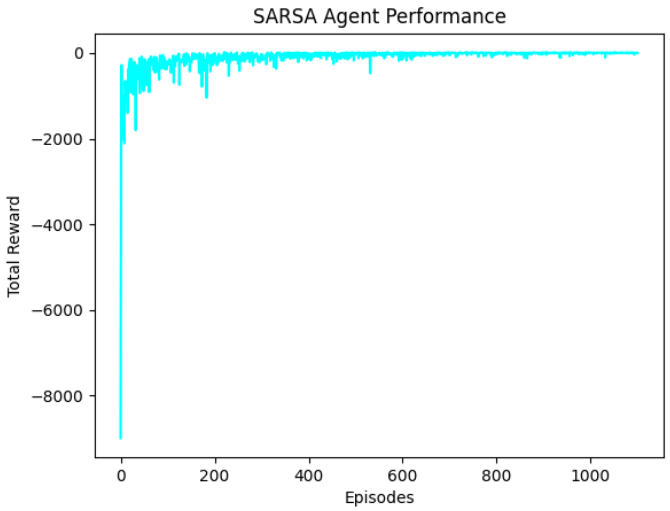

FIGURE 4 – Performance de l'agent SARSA

5 Conclusion

Les trois agents ont montré une amélioration significative de leurs performances sur le jeu *Taxi-v3*, chacun suivant une dynamique d'apprentissage différente. Le Q-Learning avec planification d'epsilon a offert une convergence plus douce, tandis que l'agent SARSA a montré une convergence stable mais plus lente. Le Q-Learning standard a quant à lui démontré une montée en performance plus rapide, se montrant efficace pour ce problème spécifique.

6 Perspectives

Plusieurs pistes d'amélioration peuvent être explorées :

— Étendre l'implémentation vers des algorithmes de type DQN pour gérer des espaces d'états plus complexes.
— Tester des stratégies d'exploration plus sophistiquées pour accélérer l'apprentissage.
— Expérimenter avec des environnements plus variés pour évaluer la capacité de généralisation des agents.