Rapport du TP3

Alexandre Bailly and Lina Farchado

October 18, 2024

1 Implémentations

- **Q Learning:** Nous avons implémenté une politique epsilon-greedy pour équilibrer exploration et exploitation, où l'agent choisit une action aléatoire avec une probabilité epsilon et la meilleure action connue sinon. Nous avons sélectionné aléatoirement une action lors de l'exploration. Pendant l'étape de mise à jour, nous avons calculé la cible de différence temporelle en utilisant la Q-valeur la plus élevée de l'état suivant.
- **Q-Learning Agent with Epsilon Scheduling:** Epsilon scheduling a été implémentée en utilisant une formule de décroissance exponentielle, assurant une réduction progressive de la valeur initiale à une valeur finale sur un nombre prédéfini d'étapes.
- **Sarsa:** La principale différence avec Q-Learning est que SARSA utilise la Q-valeur de l'action choisie dans l'état suivant pour sa mise à jour, ce qui en fait une méthode en-politique. Nous avons utilisé une approche epsilon-greedy similaire pour la sélection des actions.

2 Résultats

Nous avons comparé les performances des algorithmes Q-Learning et SARSA dans l'environnement Taxi-v3. Les résultats mettent en évidence les points suivants :

Après 1000 épisodes, les deux algorithmes ont obtenu des récompenses moyennes positives. Q-Learning a convergé plus rapidement vers des comportements optimaux, surtout grâce à l'epsilon scheduling, qui a permis de mieux équilibrer exploration et exploitation. Avec une politique epsilon-greedy standard, Q-Learning a montré une amélioration initiale rapide, mais plus de fluctuations après la phase d'apprentissage initiale, suggérant qu'un ajustement d'epsilon ou l'epsilon scheduling est crucial pour stabiliser les performances à long terme.

SARSA a utilisé une approche epsilon-greedy similaire, avec la meilleure performance obtenue pour epsilon = 0.003, menant à une récompense moyenne maximale de 7.65. La nature on-policy de SARSA l'a rendu plus stable après convergence, avec moins de fluctuations dans les récompenses comparé à Q-Learning. Cette stabilité est due au fait que l'algorithme tient compte de l'action réellement exécutée dans l'état suivant. *(cf les graphiques dans le dossier RewardsPlot)*

3 Conclusion

En conclusion, l'ajout d'un mécanisme d'epsilon scheduling a significativement amélioré les performances de l'agent en Q-Learning, lui permettant de mieux explorer l'environnement dans les premières étapes et d'adopter une politique plus stable à long terme. Bien que SARSA ait montré des performances légèrement inférieures en termes de récompenses moyennes, sa nature conservatrice lui a permis d'atteindre une meilleure stabilité après convergence.