Rapport TP3 - Florian TIGOULET

Q-Learning (Version Classique)

- J'ai utilisé une politique **epsilon-greedy** pour choisir l'action aléatoirement avec probabilité epsilon, et optimalement autrement.
- Après plusieurs tests, la meilleure valeur d'epsilon était **0.1**. Avec **epsilon = 0.25**, la récompense moyenne était de **-8.68**, tandis qu'avec **epsilon = 0.1** et un **learning rate de 0.5**, elle a atteint **2.79**, montrant une nette amélioration.

Q-Learning avec Epsilon Scheduling

- **Epsilon Scheduling** : J'ai ajouté un mécanisme de décroissance linéaire d'epsilon pour favoriser davantage l'exploration au début de l'apprentissage et l'exploitation par la suite. Les paramètres utilisés sont :
    - **epsilon_start = 1.0** (exploration maximale au début),
    - **epsilon_end = 0.001** (exploration minimale à la fin),
    - **epsilon_decay_steps = 20,000** pour contrôler la vitesse de décroissance.

Les tests avec différents taux de décroissance ont montré que **epsilon_end = 0.001** et **epsilon_decay_steps = 20,000** donnaient les meilleurs résultats avec une récompense moyenne finale de **8.3**. Ce mécanisme a significativement amélioré les performances de l'agent, lui permettant de mieux explorer l'environnement initialement et d'adopter une politique plus stable à long terme.

SARSA

L'algorithme SARSA est une approche plus conservatrice que le Q-Learning car il prend en compte l'action que l'agent va réellement suivre dans l'état suivant. L'équation d'update est :

SARSA est combiné avec un **epsilon-greedy** pour la sélection des actions. J'ai testé plusieurs valeurs pour **epsilon**, et la meilleure performance a été obtenue avec **epsilon = 0.001**.

Pour SARSA, la récompense moyenne maximale a atteint **8.26** avec **epsilon = 0.001**, ce qui est similaire à Q-Learning avec epsilon scheduling. Cela montre que SARSA fonctionne bien dans ce contexte, même si sa nature conservatrice ralentit parfois l'apprentissage.

Conclusion

L'implémentation et l'optimisation des deux algorithmes, Q-Learning et SARSA, ont montré que l'ajout d'un mécanisme de scheduling pour epsilon permet une meilleure exploration initiale et améliore les performances. SARSA, bien que conservateur, a également montré de bonnes performances avec une politique epsilon-greedy fine-tunée. Pour les deux algorithmes, une exploration limitée au départ, suivie d'une exploitation contrôlée, semble être la stratégie optimale.