Reinforcement Learning

L'objectif de ce TP est de découvrir l'apprentissage par renforcement.

Installation

On dispose d'un agent situé sur une grille 4x4. La grille est représentée par un tableau de 16 cases.

3 cases sont spéciales : - un mur (en gris) - un puit (en rouge) - une arrivée (en vert)

L'agent peut se déplacer dans les 4 directions (haut, bas, gauche, droite). Il ne peut pas traverser les murs. Lorsqu'il arrive sur une case puit, il perd la partie. Lorsqu'il arrive sur la case arrivée, il gagne la partie.

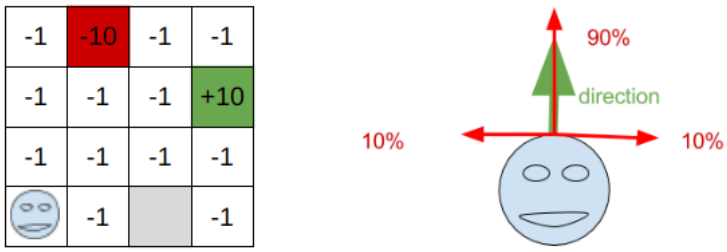

Figure 1: Gridworld

Voir le fichier `environment.py` pour plus de détails.

Q-learning

Objectif

Dans cette partie du TP, vous allez implémenter l'algorithme Q-learning pour entraîner l'agent à naviguer dans la grille. Le but est de maximiser les récompenses en trouvant le chemin le plus court vers la case arrivée, tout en évitant le puit.

Théorie

Le Q-learning est une méthode d'apprentissage par renforcement sans modèle, où l'agent apprend à estimer la valeur de l'action dans un certain état. Il utilise une

table de Q-valeurs pour stocker ces estimations. Chaque fois que l'agent prend une action et reçoit une récompense, il met à jour la valeur Q correspondant à l'état-action en utilisant la formule de mise à jour Q-learning :

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a) \right)$$

où α est le taux d'apprentissage, γ est le facteur de remise, R(s, a) est la récompense reçue après avoir exécuté l'action a dans l'état s, et s' est le nouvel état après l'action.

Instructions

1. **Initialisation**: Créez une table Q-valeurs initialement remplie de zéros. Elle doit avoir une dimension de 16×4 correspondant aux 16 états (cases) et 4 actions possibles.
2. **Boucle d'apprentissage**: Pour chaque épisode d'apprentissage :
   - Réinitialisez l'environnement (agent au départ).
   - Tant que la partie n'est pas finie :
     - Choisissez une action (au hasard ou en utilisant la politique actuelle basée sur les Q-valeurs).
     - Appliquez cette action et observez la récompense et le nouvel état.
     - Mettez à jour la table Q-valeurs en utilisant la formule de Q-learning.
     - Si l'agent tombe dans le puit ou atteint l'arrivée, terminez l'épisode.
3. **Politique de choix d'action**: Utilisez une stratégie d'exploration/exploitation, comme espilon-greedy, où l'agent choisit parfois une action au hasard pour explorer.
4. **Paramètres d'apprentissage**: Vous pouvez commencer avec un $\alpha = 0.1$ et un $\gamma = 0.9$. Ajustez ces paramètres au besoin pour améliorer les performances de l'agent.

Tâches

- Implémentez l'algorithme Q-learning.
- Testez votre agent sur 1000 épisodes et observez son apprentissage.
- Analysez l'évolution des performances de l'agent (par exemple, le nombre de mouvements nécessaires pour gagner au fil des épisodes).

Questions de réflexion

1. Comment la valeur de epsilon (dans la stratégie epsilon-greedy) influence-t-elle l'apprentissage de l'agent ?

2. Quelle est l'importance du taux d'apprentissage α et du facteur de remise γ ?
3. Comment interpréter la table des Q-valeurs finale ?