Reinforcement Learning

Projet : Entraîner un agent à rejoindre une sortie en 2D et en 3D

L'objectif de ce projet est d'appliquer un algorithme d'apprentissage par renforcement en utilisant une table de Q-Learning.

La première partie consistera en l'apprentissage d'une table de Q-Learning pour un cas 2D (une grille) et de proposer une visualisation de test de cette table.

La seconde partie consistera en l'adaptation du code 2D sur un environnement 3D (un cube). Ensuite, nous entraînerons une table de Q-Learning dans cet environnement.

Table of Contents

Principes du RL 2
- Agent 2
- Environnement 2
- Etats 2
- Actions 2
- Récompenses 2
- Episode 2
- Q-Learning 2
- Paramètres 3
  - Learning Rate 3
  - Exploration Rate 3
  - Discount Factor 3

RL sur une grille (2D) 3
- Définition de l'environnement 3
  - Grille 3
  - Actions 3
  - Etat 3
  - Q-Table 3
- Entraînement 3

RL sur un cube (3D) 4
- Environnement 4
- Visualisation 4

Principes du RL

L'apprentissage par renforcement est une méthode d'apprentissage automatique qui consiste à apprendre à un agent par essais et erreurs à partir d'interactions avec un environnement.

Voici les éléments de base du RL :

Agent

L'agent est l'entité qui interagit avec son environnement pour atteindre un objectif spécifique. Il prend des décisions en fonction des informations qu'il reçoit de son environnement. La manière dont il interagira avec son environnement dépendra de l'algorithme de renforcement qu'il utilisera (Q-Learning, réseau de neurones etc.)

Environnement

L'environnement est l'endroit avec lequel l'agent évolue et interagie. Ici l'environnement est la grille ou le cube.

Etats

Les états représentent les différentes configurations possibles dans lesquels l'agent peut se trouver. Dans chaque état, l'agent dispose de différentes actions possibles qu'il choisira en fonction de l'algorithme d'apprentissage qui lui a été fourni. Dans notre cas, les états représentent les différentes positions/coordonnées dans la grille/cube dans lesquelles se trouve l'agent.

Actions

Les actions représentent les différents choix que peut prendre l'agent dans les différents états. Dans notre cas les actions pourront être haut/bas/gauche/droite, elles représentent un déplacement dans l'environnement.

Récompenses

Les récompenses sont les informations/feedback/reward que reçoit l'agent de l'environnement. Ces récompenses sont positives ou négatives et sont définies par rapport à l'action que choisit l'agent. Par exemple, si l'agent se trouve sur la position de sortie, la reward sera positive alors que si c'est un obstacle ou un mur elle sera négative.

Episode

Un épisode représente un ensemble d'interactions (actions) avec l'environnement.
Lors d'une phase d'entraînement, un nombre d'épisode peut être défini ce qui permettra à l'agent de recommencer les étapes d'entraînement à partir de sa position de départ jusqu'à ce qu'il atteigne son état objectif.

Q-Learning

Le Q-Learning est un algorithme d'apprentissage par renforcement qui apprend à prendre des décisions en apprenant une fonction de valeur d'action, appelée fonction Q. L'objectif est de maximiser la somme des récompenses attendues à long terme. L'agent utilise cette fonction Q pour prendre des décisions optimales en choisissant l'action ayant la plus grande valeur de Q dans chaque état.

Paramètres

Learning Rate

Ce paramètre permet de contrôler l'importance de nouvelles valeurs (les rewards) par rapport à ce qui est connu. Il détermine à quel point les rewards ont de l'importance dans la mise à jour de la Q-Table.

Exploration Rate

Ce paramètre contrôle la proportion d'actions à prendre en compte pour l'exploration. Plus il est élevé, plus la possibilité de prendre une action, même si elle n'est pas optimale, est élevée.

Discount Factor

Ce paramètre contrôle l'importance des récompenses futures par rapport aux récompenses immédiates. Il est utilisé pour estimer la récompense totale que l'agent peut accumuler en suivant une politique donnée à partir d'un état donné. Si le facteur de remise est proche de zéro, l'agent accorde une importance élevée aux récompenses immédiates et ne prend pas en compte les récompenses futures et inversement pour un facteur élevé.

RL sur une grille (2D)

L'objectif ici est d'apprendre à un agent à rejoindre une cellule de la grille.
L'agent commencera au coin haut gauche de la grille et la sortie sera en bas à droite.
Il y aura des obstacles pour gêner l'agent.

Définition de l'environnement

Grille

La grille est représentée par une matrice de taille 10x10.

Actions

Il est possible pour l'agent de réaliser 4 actions : haut, bas, gauche et droite. L'agent peut se déplacer sur la grille en choisissant l'une de ces actions.

Etat

Chaque état représente une position sur la grille (x, y). Pour chaque état, il est possible de réaliser 1 une action et cette action sera choisie parmi les 4 actions possibles.

Q-Table

La table de Q-Learning contiendra alors les 10x10=100 états (positions) et pour chaque positions 4 états.

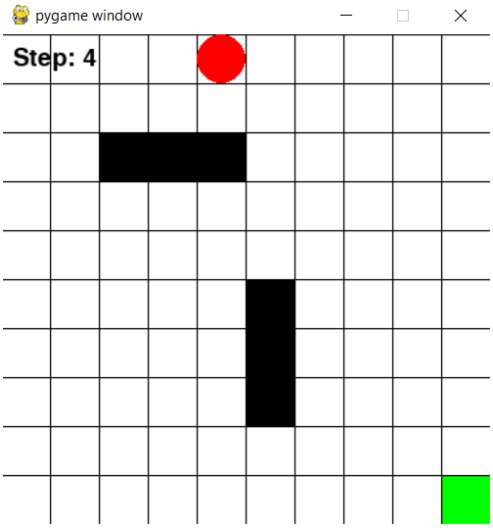
Pour chaque position, la valeur contenue pour chaque état représente l'estimation de la qualité de l'action à prendre (Q-value). C'est la somme des récompenses sur laquelle l'algorithme va se baser pour prendre sa décision en fonction des paramètres initiaux (learning rate, exploraiton rate, discount factor).

Entraînement

Pour l'entraînement, nous définissions les paramètres principaux et lançons l'entraînement de la Q-Table.

Ensuite, nous utilisons la Q-Table pour voir comment l'agent interagit avec l'environnement avec ce qu'il a appris.

Pour visualisation des résultats, voir et lancer le notebook « RL_2D.ipynb ».



RL sur un cube (3D)

Maintenant que nous avons défini notre environnement comme étant une grille 2D et que nous avons réussi à entraîner et à tester une table issue du Q-Learning sur cet environnement, nous allons modifier l'environnement pour qu'il corresponde à un cube 3d où chaque état représentera une position (x, y, z) de ce cube.

L'objectif sera d'entraîner la Q-Table afin d'obtenir la trajectoire optimale que l'on visualisera en 3D.

Environnement

L'environnement est maintenant un cube 3D de dimensions 10x10x10.

L'agent a le choix entre 6 actions : haut, bas, gauche, droite, avant arrière.

L'agent se trouve à l'origine du cube et l'objectif à l'extrémité du cube.

Visualisation

Trajectoire optimale de 2 points de vue différents. (points bleus = obstacles)