

2022

Q-Learning



En cadre par:

Pr. **YAHYAOU** Ali

Chellak El Mehdi

MIDVI S3

01/01/2022

Definition :

Q-learning est un algorithme d'apprentissage *par renforcement sans modèle*.

Q-learning est un algorithme d'apprentissage **basé sur des valeurs**. Les algorithmes basés sur la valeur mettent à jour la fonction de valeur en fonction d'une équation (en particulier l'équation de Bellman). Alors que l'autre type, **basé sur** la politique, estime la fonction de valeur avec une politique gourmande obtenue à partir de la dernière amélioration de la politique.

Qu'est-ce que ce « Q » ?

Le « Q » dans Q-Learning est synonyme de qualité. La qualité représente ici l'utilité d'une action donnée pour obtenir une récompense future.

Présentation de la Q-Table

Q-Table est la structure de données utilisée pour calculer les récompenses futures maximales attendues pour l'action à chaque état.

Fondamentalement, ce tableau nous guidera vers la meilleure action à chaque état. Pour apprendre chaque valeur de la table Q, l'algorithme Q-Learning est utilisé.

Fonction Q

La fonction Q utilise l'équation de Bellman et prend deux entrées : l'état (s) et l'action (a).

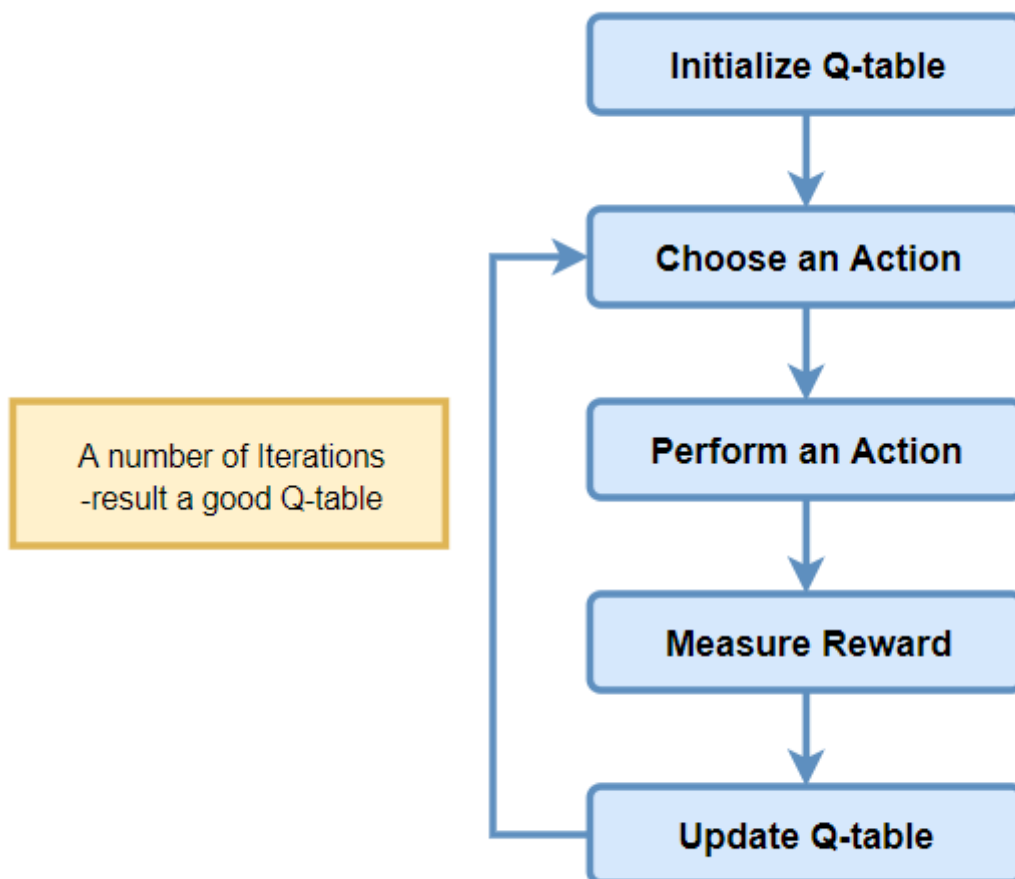
$$Q^{\pi}(s_t, a_t) = \underline{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | s_t, a_t]$$

Q-Values for the state given a particular state

Expected discounted cumulative reward




Given the state and action

Processus de l'algorithme Q-learning

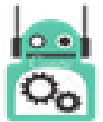










Étape 1 : initialiser la Q-Table

Nous allons d'abord construire une table Q. Il y a n colonnes, où n= nombre d'actions. Il y a m lignes, où m= nombre d'états. Nous allons initialiser les valeurs à 0.

Actions :    

Start	0	0	0	0
Nothing / Blank	0	0	0	0
Power	0	0	0	0
Mines	0	0	0	0
END	0	0	0	0

					
					
					
					
				End	

Dans notre exemple de robot, nous avons quatre actions ($a = 4$) et cinq états ($s = 5$). Nous allons donc construire un tableau avec quatre colonnes et cinq lignes.

Étapes 2 et 3 : choisir et effectuer une action

Cette combinaison d'étapes est effectuée pour une durée indéfinie. Cela signifie que cette étape s'exécute jusqu'au moment où nous arrêtons la formation ou que la boucle d'entraînement s'arrête comme défini dans le code.

Nous choisirons une action (a) dans l'état (s) en fonction de la Q-Table. Mais, comme mentionné précédemment, lorsque l'épisode commence initialement, chaque valeur Q est 0.

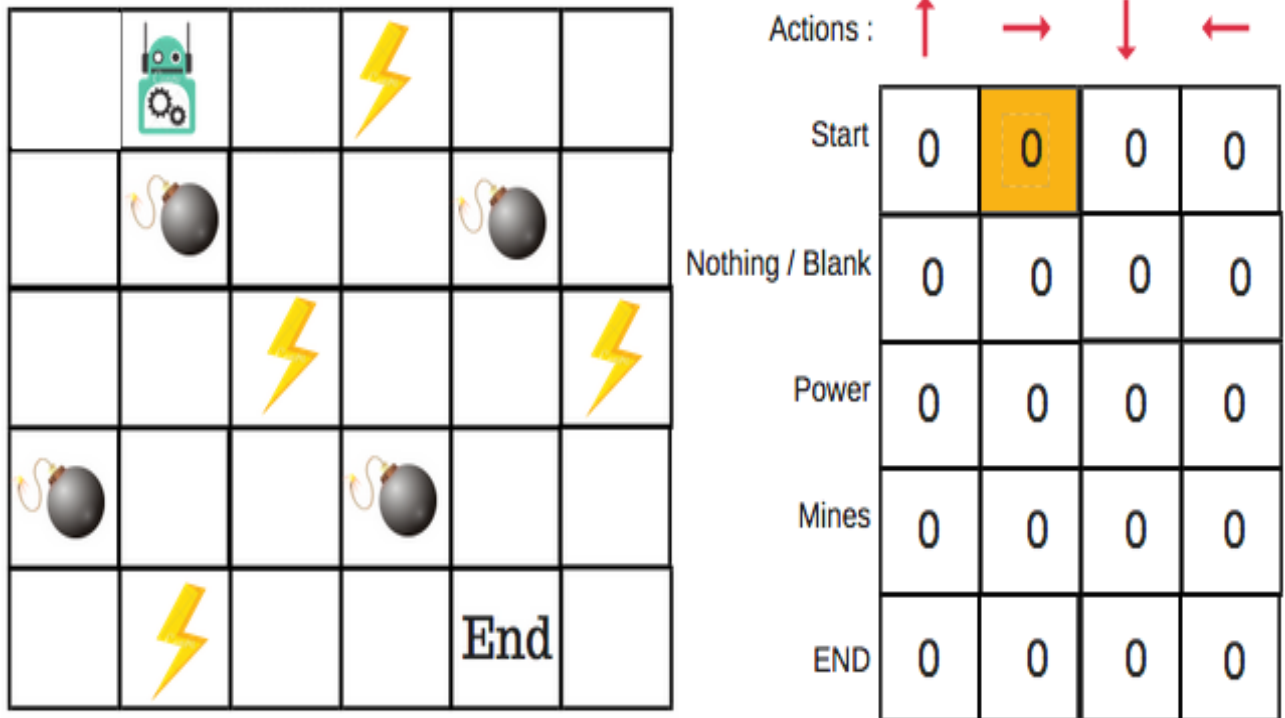
Nous utiliserons ce qu'on appelle la stratégie gourmande epsilon.

Au début, les taux d'epsilon seront plus élevés. Le robot explorera l'environnement et choisira des actions au hasard. La logique derrière cela est que le robot ne sait rien de l'environnement.

Au fur et à mesure que le robot explore l'environnement, le taux d'epsilon diminue et le robot commence à exploiter l'environnement.

Au cours du processus d'exploration, le robot devient progressivement plus confiant dans l'estimation des valeurs Q .

Pour l'exemple du robot, vous avez le choix entre quatre actions : haut, bas, gauche et droite. Nous commençons la formation maintenant – notre robot ne sait rien de l'environnement. Donc, le robot choisit une action aléatoire, disons à droite.



Nous pouvons maintenant mettre à jour les valeurs Q pour être au début et se déplacer vers la droite en utilisant l'équation de Bellman.

Étapes 4 et 5 : évaluer

Maintenant, nous avons pris une mesure et observé un résultat et une récompense. Nous devons mettre à jour la fonction $Q(s,a)$.

$$\text{New } Q(s,a) = Q(s,a) + \alpha [R(s,a) + \gamma \max_{a'} Q'(s',a') - Q(s,a)]$$

- New Q Value for that state and the action
- Learning Rate
- Reward for taking that action at that state
- Current Q Values
- Maximum expected future reward given the new state (s') and all possible actions at that new state.
- Discount Rate

Conclusion :

- Q-Learning est un algorithme d'apprentissage par renforcement basé sur la valeur qui est utilisé pour trouver la politique optimale de sélection d'action à l'aide d'une fonction Q.
- Notre objectif est de maximiser la fonction de valeur Q.
- La table Q nous aide à trouver la meilleure action pour chaque état.
- Il aide à maximiser la récompense attendue en sélectionnant la meilleure de toutes les actions possibles.
- $Q(\text{état}, \text{action})$ renvoie la récompense future attendue de cette action à cet état.
- Cette fonction peut être estimée à l'aide de Q-Learning, qui met à jour de manière itérative $Q(s,a)$ à l'aide de **l'équation de Bellman**.
- Dans un premier temps, nous explorons l'environnement et mettons à jour la Q-Table. Lorsque la Q-Table est prête, l'agent commence à exploiter l'environnement et commence à prendre de meilleures mesures.