UNIVERSITÉ
Abdelhamid MEHRI
CONSTANTINE 2

LISIA
Laboratoire Informatique en Sciences de données et Intelligence Artificielle

Module : IAD & AI

L'Intelligence Artificielle Distribuée

& Agent Intelligent

Master 1 : Science de Données et Intelligence Artificielle

2023 - 2024

Plan de Présentation

- Apprentissage par Renforcement :

  Définitions , Types d'Apprentissage, Comparaison, RL pour Agent ..

- Exploration & Exploitation.
- La Stratégie : ε-Greedy.
- Valeur d'Etat : V(s).
- Valeur d'Action : Q(s,a).
- **L'Algorithme : Q-Learning.**
- **L'Algorithme : Deep Q-Learning.**
- L'Algorithme : Policy Gradient.

Q-Learning

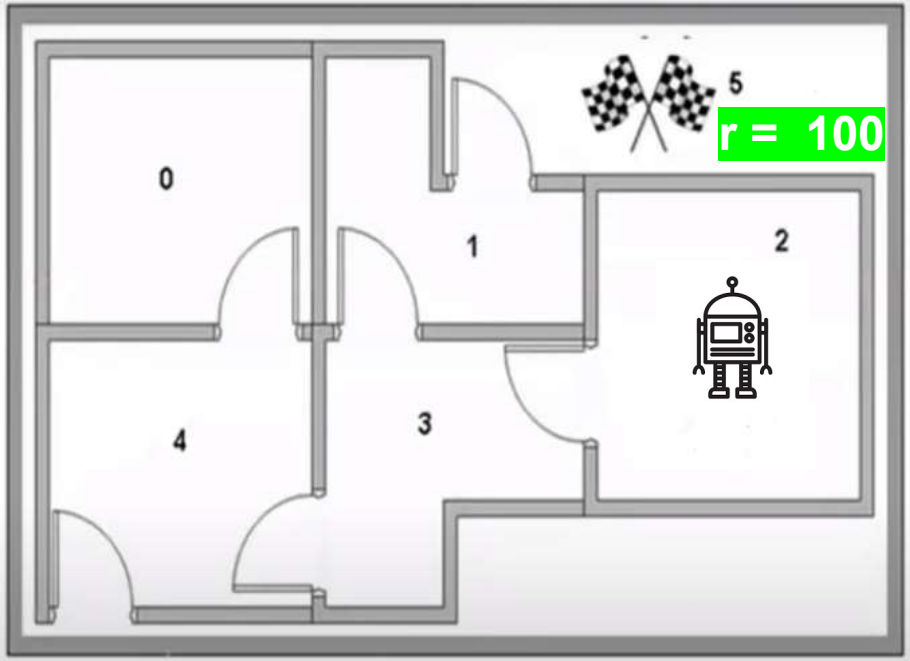
- Le Q-Learning est un type d'algorithme d'apprentissage par renforcement utilisé en apprentissage automatique.

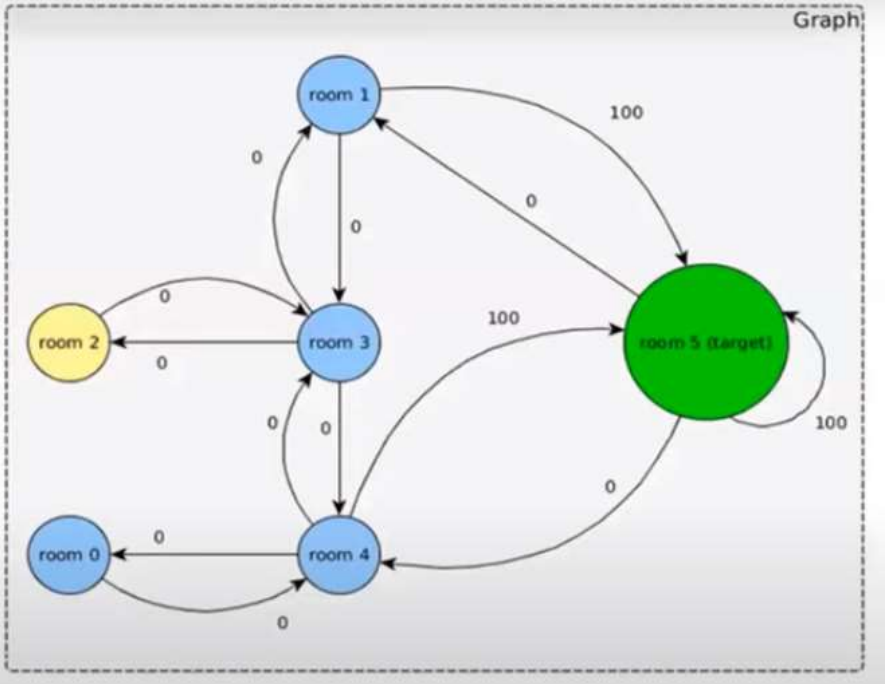
- Il s'agit d'une technique d'apprentissage par renforcement sans modèle, ce qui signifie qu'elle ne nécessite pas de modèle de l'environnement et apprend plutôt par interaction par essais et erreurs.

- Le "Q" dans Q-learning signifie : Qualité, représentant la qualité d'une action particulière dans un état donné. L'algorithme vise à apprendre une politique optimale pour sélectionner des actions afin de maximiser les récompenses cumulatives au fil du temps.

Fonctionnement de Q-Learning

- **L'étape Primordiale** : Transformer l'environnement (E), en un graphe (Modélisation MDP : Marcov Decision Process).
  - **Nœuds :** Les états.
  - **Arcs :** Les récompenses d'actions.





Fonctionnement de Q-Learning

1. **Initialisation** : Initialiser la table Q, une matrice où les lignes représentent les états et les colonnes représentent les actions. Initialement, les valeurs dans la table Q sont arbitraires.

2. **Exploration vs Exploitation** : À chaque pas de temps, l'agent décide s'il doit explorer de nouvelles actions ou exploiter ses connaissances actuelles.

- ✓ L'exploration est importante pour découvrir de nouvelles stratégies, tandis que l'exploitation maximise les récompenses en fonction des connaissances actuelles.

Fonctionnement de Q-Learning

3. **Sélection de l'action :** L'agent sélectionne une action à prendre dans l'état actuel en fonction d'une stratégie d'exploration-exploitation.

- ✓ Cela pourrait être epsilon-greedy, où l'agent choisit une action aléatoire avec une probabilité epsilon et l'action avec la valeur Q la plus élevée avec une probabilité (1 - epsilon).

Fonctionnement de Q-Learning

4. **Mise à jour des valeurs Q :** Après avoir pris une action et observé la récompense résultante et l'état suivant, l'agent met à jour la valeur Q pour la paire état-action actuelle en utilisant la règle de mise à jour Q-learning.

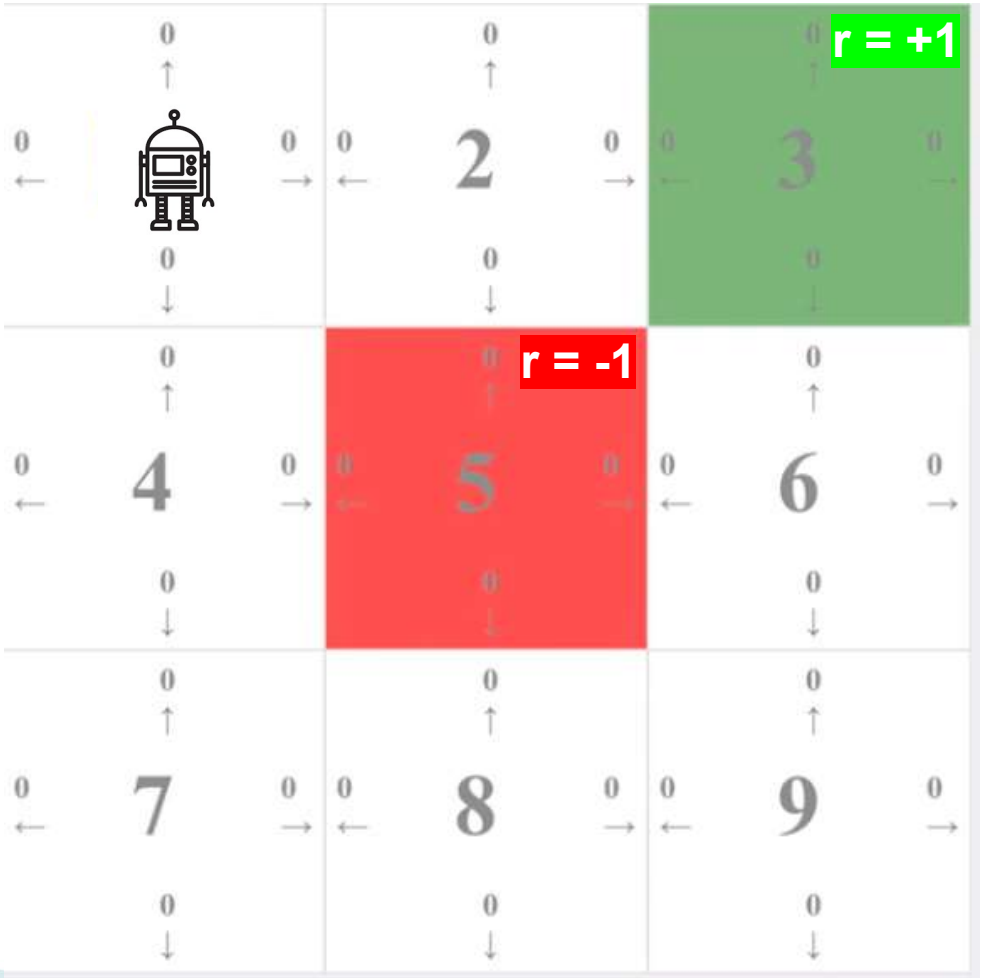
5. **Répétition :** L'agent continue d'interagir avec l'environnement, mettant à jour les valeurs Q en fonction des récompenses observées, jusqu'à la convergence ou un nombre prédéfini d'itérations.

Domaine d'Application : Q-Learning

- Le Q-learning est un algorithme puissant et a été appliqué à diverses tâches, telles que les jeux, le contrôle robotique et les problèmes d'optimisation.
- Cependant, il présente certaines limitations, telles que le besoin d'espaces d'états et d'actions discrets, et peut être lent à converger dans de grands espaces d'états.
- Des extensions comme les Deep Q-Networks (DQN) abordent certaines de ces limitations en utilisant des réseaux neuronaux pour approximer les valeurs Q.

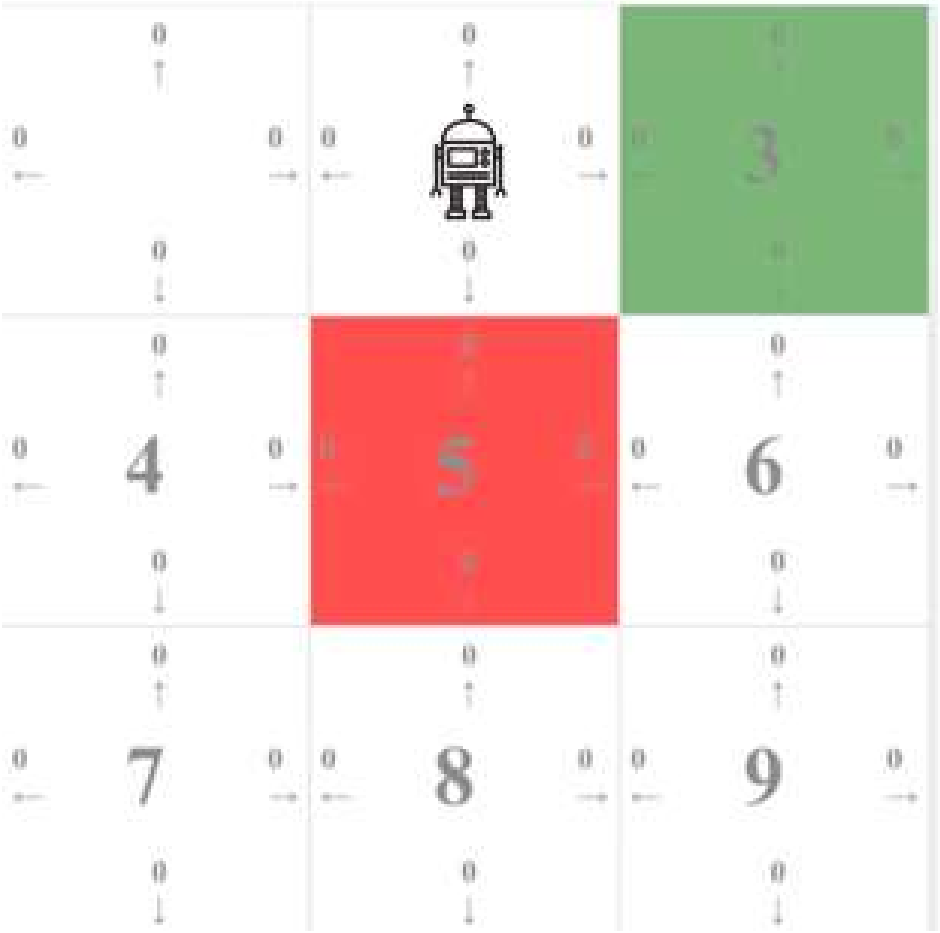
Exemple : Q-Learning

$$Q(s_t, a_t)_{new} = Q(s_t, a_t)_{old} + \alpha[r + \gamma max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)_{old}]$$



Exemple : Q-Learning

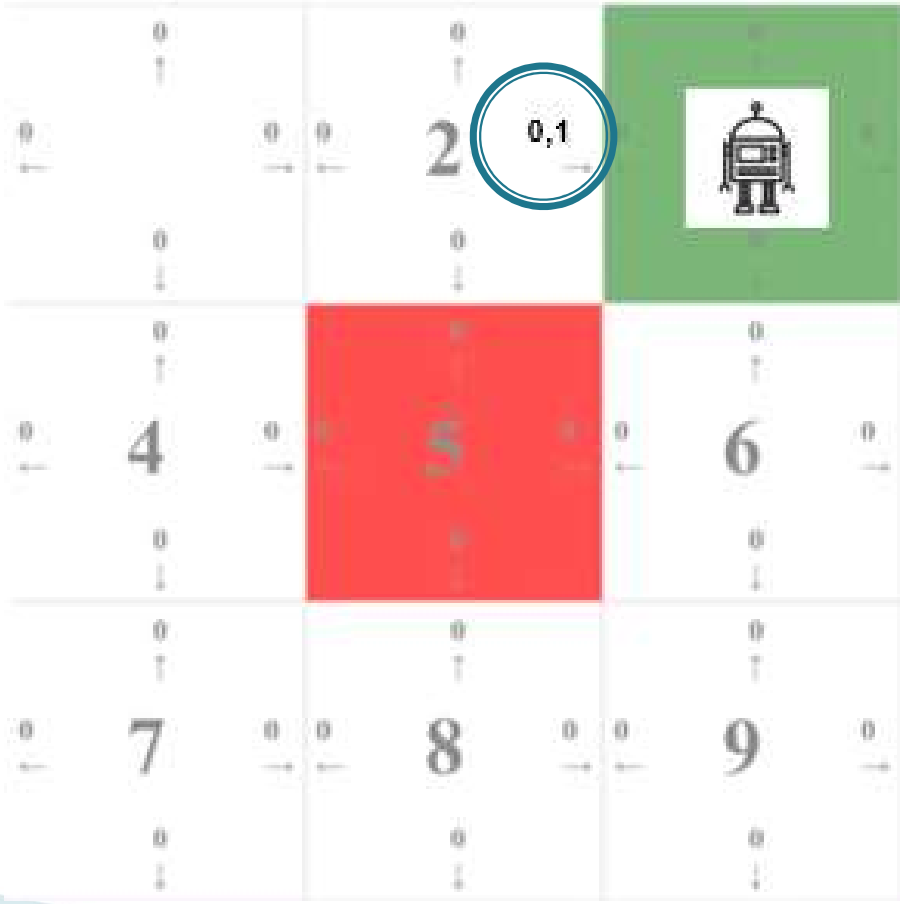
- *Aucune mise à jour dans la Q-Table par rapport à ce déplacement :*



Exemple : Q-Learning

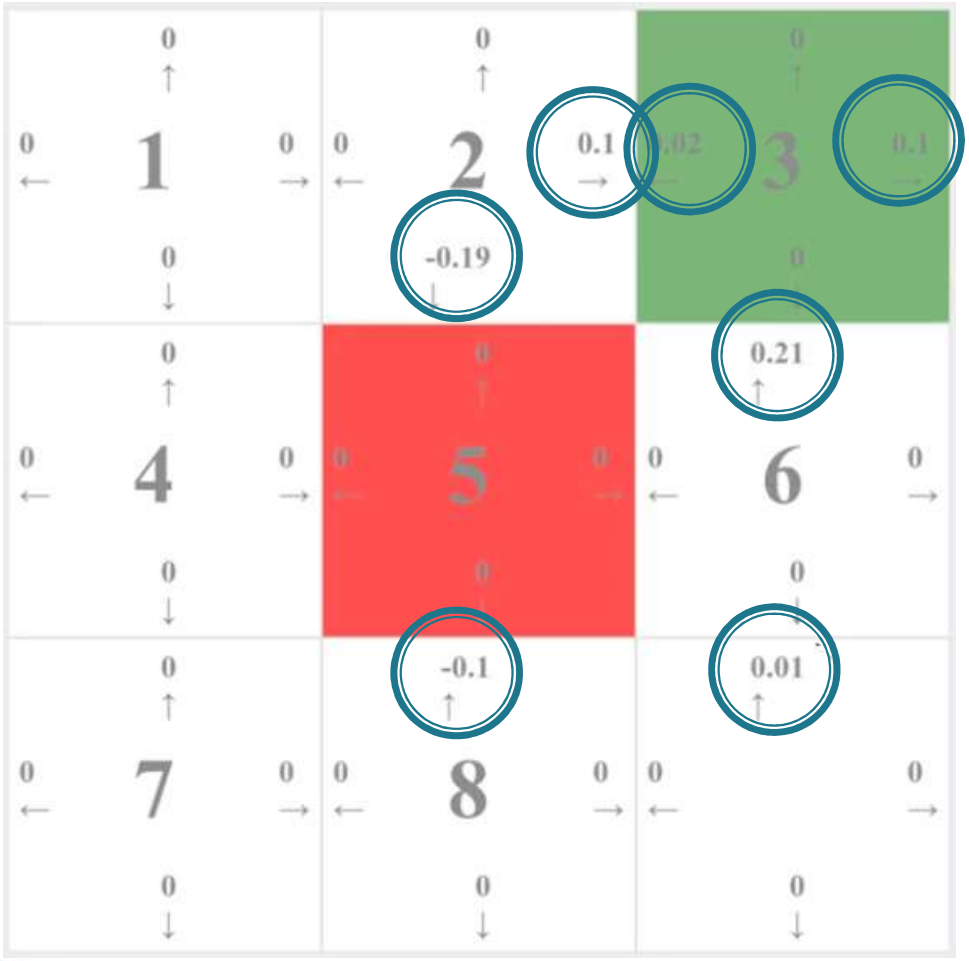
- *En appliquant l'ancienne formule, avec :* ***α = 0,1*** *et* ***γ = 0,9*** *:*

$$Q(s_t, a_t)_{new} = Q(s_t, a_t)_{old} + \alpha[r + \gamma max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)_{old}]$$



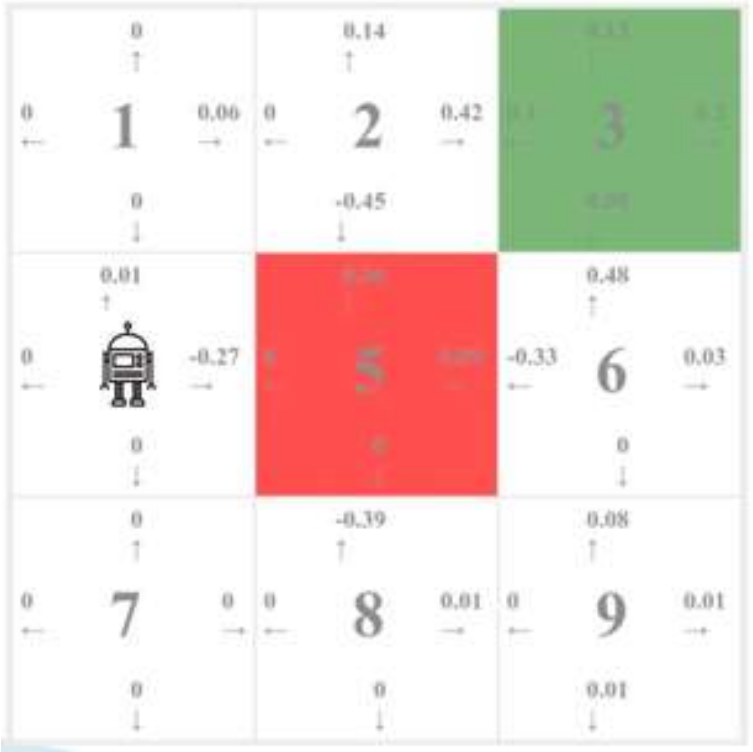
Exemple : Q-Learning

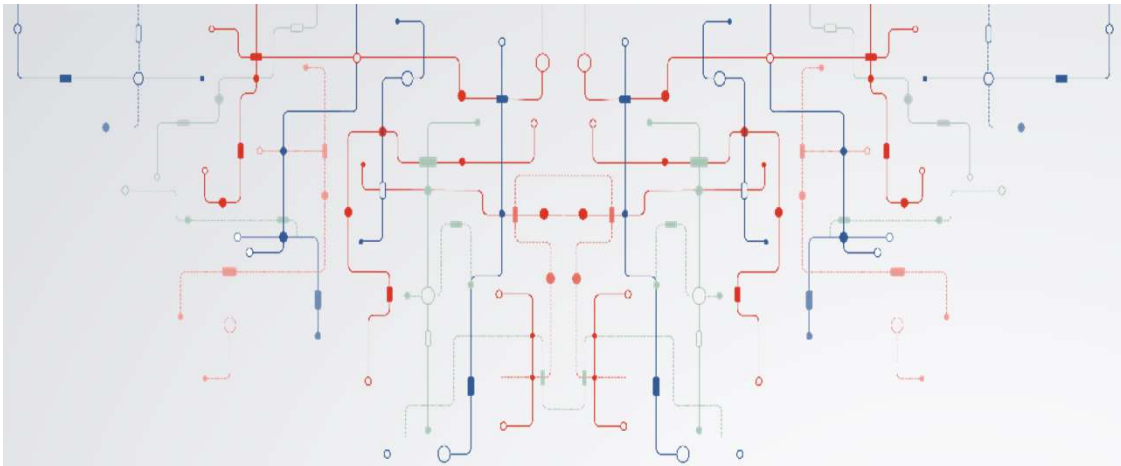
- *En lançant l'algorithme plusieurs fois, les valeurs vont commencer à se propager au fur et à mesure dans les états précédents :*



Exemple : Q-Learning

- *Jusqu'à l'arrivée des valeurs finales de notre Grille (MDP).*
- ***Question :*** *Cette simple Grille peut être projetée dans un environnement complexe ? ..*
- ***Tendance :*** *Allers vers Deep Q-Learning ..*





Suivre avec DQL ..