EPI EDUCATIONAL GROUP

Module: Machine Learning

Machine Leraning

Niveau : GL4

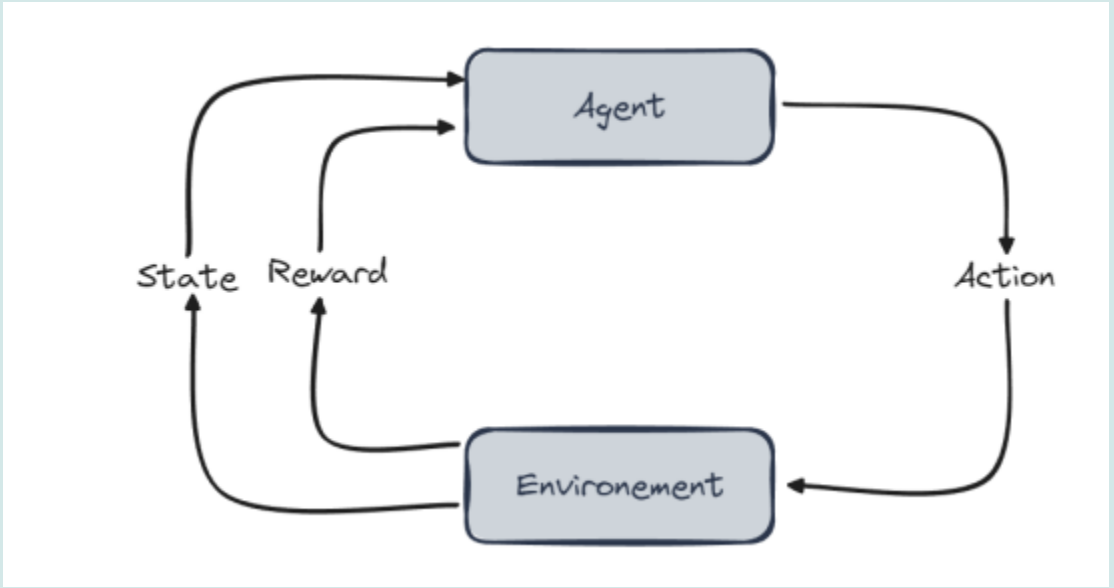
Enseignante: Bènène Fradi Boumiza

Année universitaire: 2025/2026

Apprentissage Par Renforcement

- créer un agent, libre d'entreprendre des actions dans un environnement.
- Ces actions modifient l'état (state) de l'agent, et ce changement d'état s'accompagne d'une récompense (reward).
- Pour l'agent, le but du jeu est de maximiser ses récompenses, ce qui le pousse à apprendre quelles actions effectuer pour obtenir le plus de récompenses.

➔Apprendre à un **agent** à atteindre un but à partir de son **expérience**

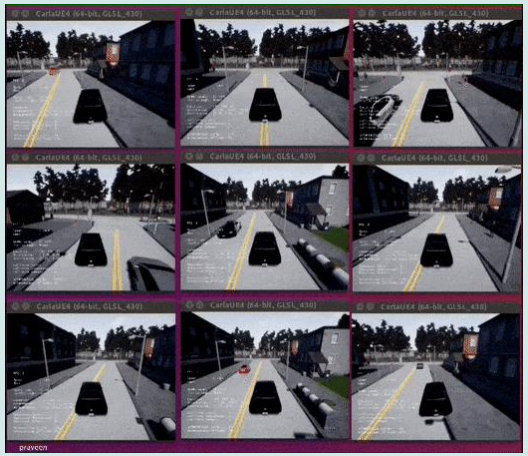


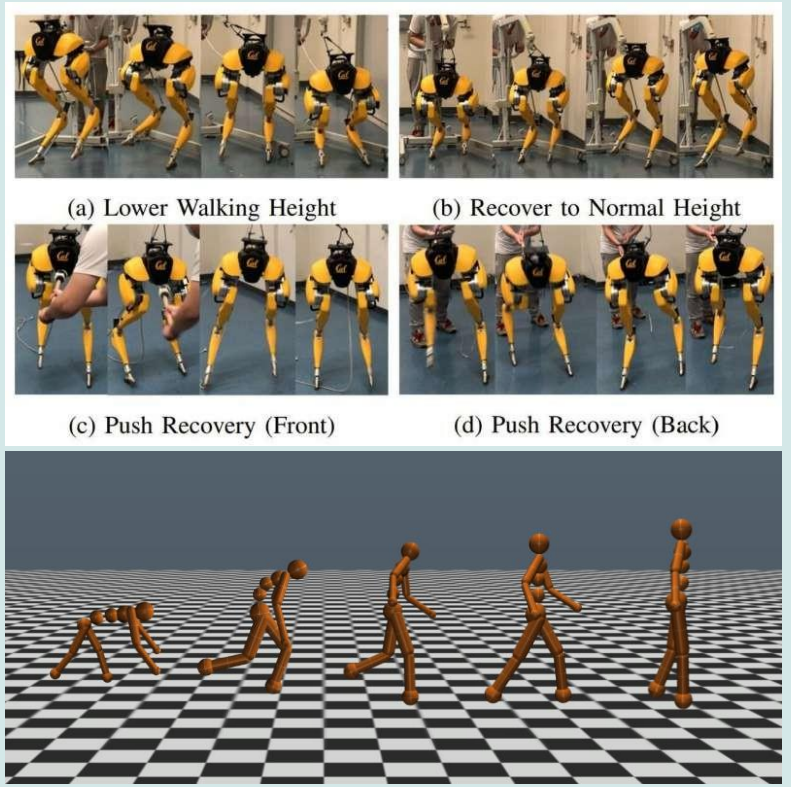
Apprentissage Par Renforcement

- Cette discipline est très utilisée pour les applications suivantes :
  - Robotique
  - Véhicules autonomes (voitures, drones)
  - Trading
  - Algorithmes de prise de décision

Apprentissage Par Renforcement

❑ Exemples : contrôle





Apprentissage Par Renforcement

❑ Exemples : jeux

Apprentissage Par Renforcement

❑ Caractéristiques de l'apprentissage par renforcement

- ❑ Pas de supervision, juste une **récompense**
- ❑ Le *feedback* est retardé
- ❑ Le processus est séquentiel : les données ne sont **pas *iid***

  Chaque état dépend :de l'état précédent, de l'action choisie avant

➔Donc les données sont **liées dans le temps**.

- ❑ Les actions de l'agent affectent l'environnement

➔La récompense (*reward*) R_t : mesure à quel point l'agent est performant à l'étape t

Apprentissage Par Renforcement

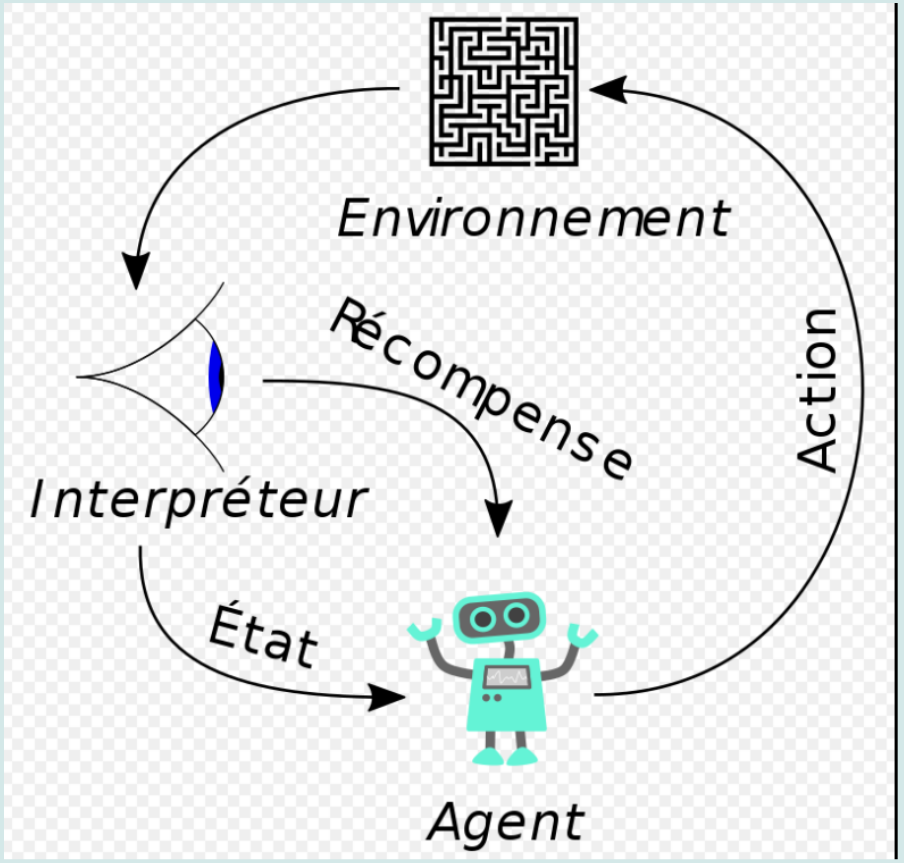
❑ Exemples de rewards

- ❑ Manoeuvre de pilotage/conduite :
  - + R / —R si l'agent suit la bonne trajectoire/ dévie
  - —∞ en cas d'accident
- ❑ Jeu de plateau :
  - + R en cas de victoire, —R en cas de défaite
- ❑ Jeu vidéo :
  - + R / —R par augmentation/diminution de score
    —∞ en cas de défaite

Apprentissage Par Renforcement

- **L'idée centrale est:**
  - De créer un agent libre
  - D'entreprendre des actions dans un environnement.
  - Ces actions modifient l'état (state) de l'agent,
  - Ce changement d'état s'accompagne d'une récompense (reward).

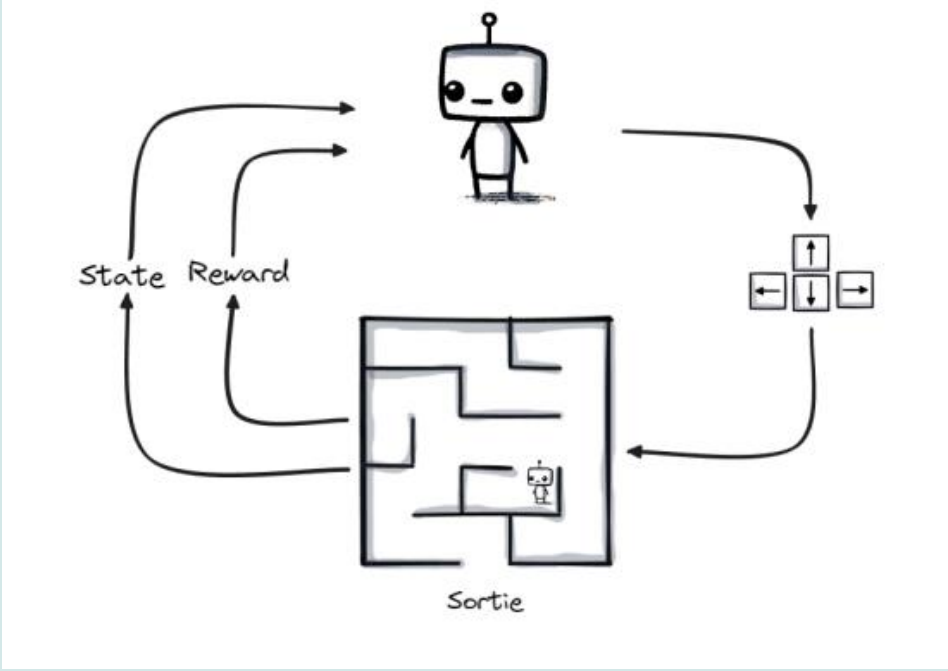
- Pour l'agent, le but du jeu est de maximiser ses récompenses, ce qui le pousse à apprendre quelles actions effectuer pour obtenir le plus de récompenses.



Apprentissage Par Renforcement

❑Exemple:

- Un petit robot pour qu'il apprenne à sortir d'un labyrinthe le plus rapidement possible
- Construire un algorithme de recherche du chemin le plus court basé sur la théorie des graphes, mais cela reviendrait à coder explicitement le comportement de la machine.
- Programmer la machine pour qu'elle développe elle-même la stratégie lui permettant de quitter le labyrinthe.



➔Utiliser une approche d'apprentissage par renforcement.

- Agent : le petit robot
- le labyrinthe l'environnement dans lequel il évolue.

Apprentissage Par Renforcement

- **L'état (State)** correspond à la position de **l'agent** dans le **labyrinthe**, et les actions à ses déplacements possibles (haut, bas, droite, gauche)

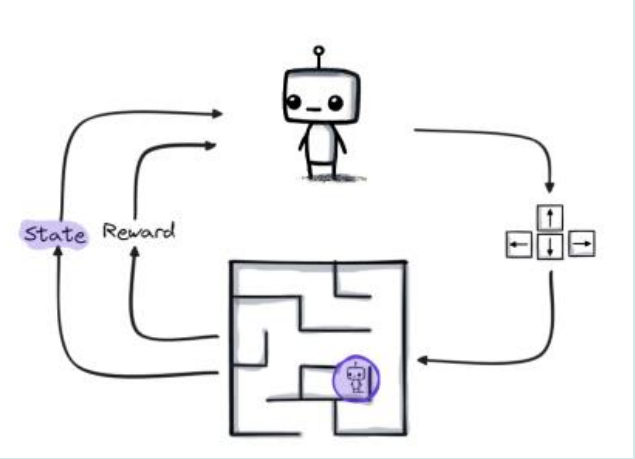
- Depuis cet état, l'agent peut choisir une action, par exemple "aller en haut".

- En effectuant cette action, l'agent change son état, ce qui s'accompagne d'une récompense.

Ici, nous attribuons à la machine un score de -1 point.

- Objectif : "trouver la sortie le plus vite possible" : but étant de maximiser son score final

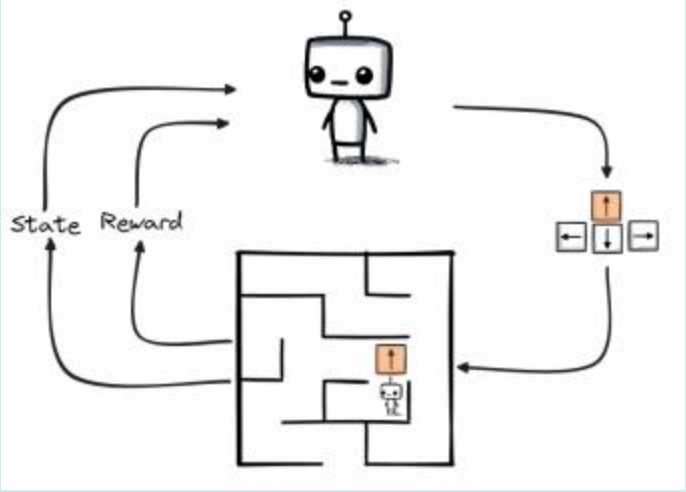
- Elle cherchera donc le moyen le plus rapide de rejoindre la sortie, préférant marquer -10 points plutôt que -15.



Apprentissage Par Renforcement

- Apprendre à la machine, quelle action effectuer lorsqu'elle se situe dans un état donné.
- nous cherchons à développer une fonction f(s) = a qui prend en entrée l'état s et produit une action a.

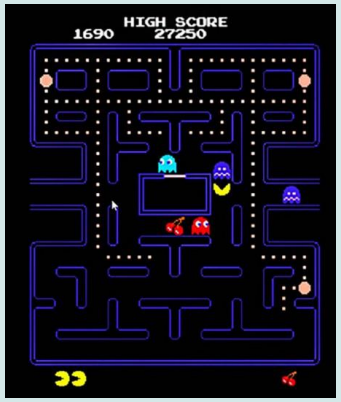
➔Cette fonction est couramment nommée politique d'action.



Apprentissage Par Renforcement

❑ Environnement

- Complètement ou partiellement connu
- Conditionne les actions de l'agent
- Émet les récompenses pour l'agent



Apprentissage Par Renforcement

❑ Agent

- ❑ L'agent désigne la machine, le robot ou l'algorithme qui doit apprendre la politique d'action optimale pour atteindre le but fixé.

- ❑ Évolue dans l'environnement en se basant sur un ensemble d'action A

- ❑ Reçois une récompense à chaque étape

➔Le but de l'agent est de maximiser l'espérance des récompenses cumulées

Apprentissage Par Renforcement

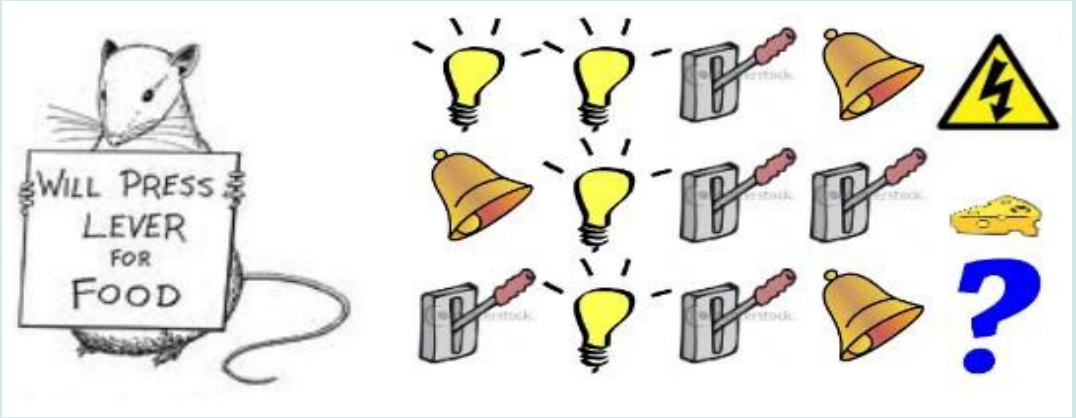
❑ État

Un **état** s_t est une fonction de l'historique

$H_t = \{O_0, R_0, A_0, \cdots, A_{t-1}, O_t, R_t\}$:

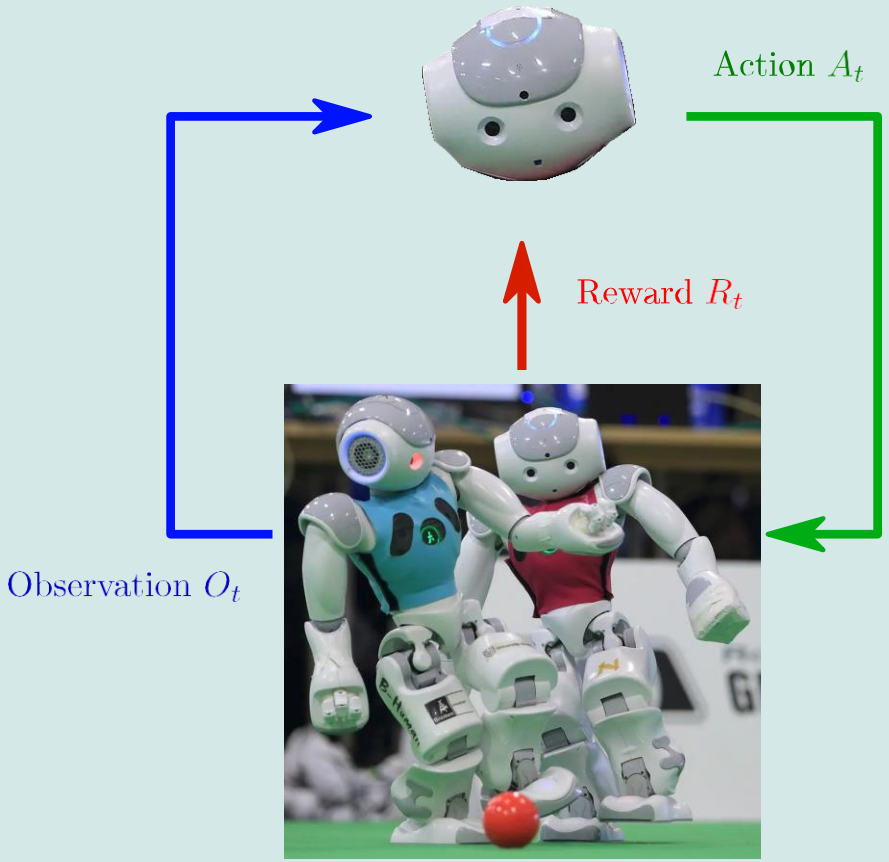
$$S_t = f(H_t)$$

Il contient toute l'inforamtion utilisée par l'agent pour décider son action prochaine.



Apprentissage Par Renforcement

❑ Schéma classique en RL



foreach Time step t do

The agent :

- gets an observation O_t and a reward R_t
- Executes action A_t
-

The environment :

- Receives action A_t
- Emits observation O_{t+1} and reward R_{t+1}
-

end

Apprentissage Par Renforcement

❑ Formalisation en RL

En apprentissage par renforcement, la description formelle d'un environnement ainsi que les états S et actions A de l'agent se fait au travers d'un **Processus de Décisions Markovien (MDP)**

Une vaste majorité des problèmes de RL peuvent être caractérisés par un MDP

Apprentissage Par Renforcement

❑ Processus Markovien

- ❑ Un processus Markovien est un procéssus aléatoire sans mémoire.
- ❑ une succession d'états Markovien

Un **processus Markovien** est défini par le couple (S, P) où :

- S est un ensemble fini d'états
- P est la matrice de transition associée à S

Apprentissage Par Renforcement

❑ État Markovien

Un **état Markovien** contient toute l'information utile du passé pour prédire le future

Un état S_t est **Markovien** si et seulement si

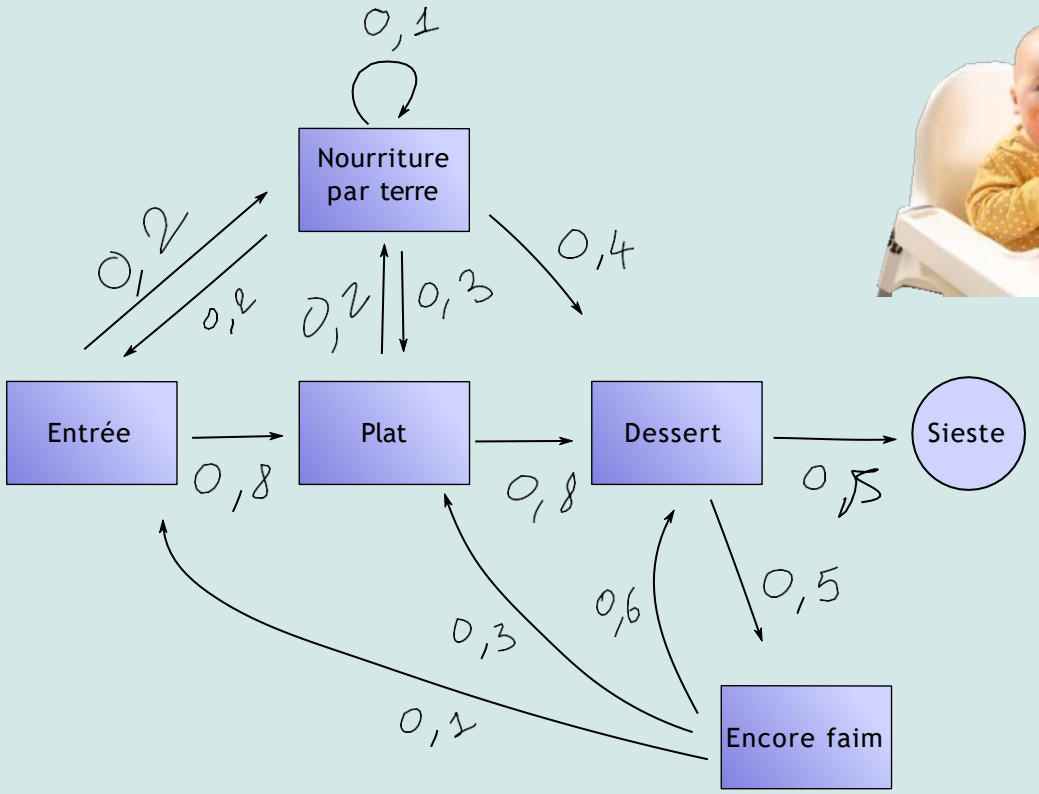
$$p_{ss^0} = P[S_{t+1} = s^0/S_t = s] = P[S_{t+1}/S_1, \cdots, S_t]$$

La **matrice de transition** associée à un ensemble d'états S collecte toutes les probabilité de passer d'un état à un autre :

$$\mathcal{P} = \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & & \vdots \\ p_{n1} & \cdots & p_{nn} \end{bmatrix}$$

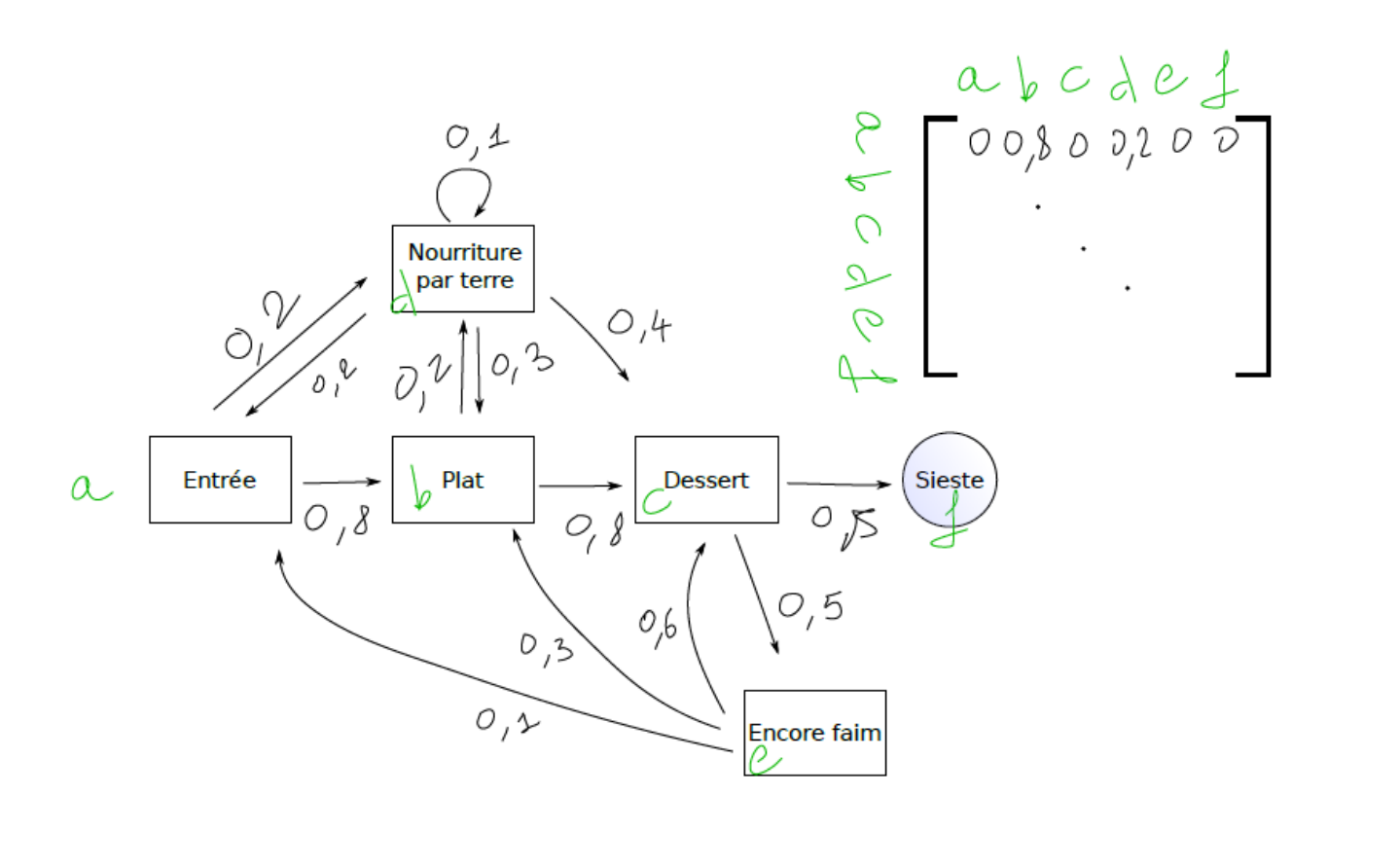
Apprentissage Par Renforcement

❑ Exemple



Apprentissage Par Renforcement

❑ Exemple



Apprentissage Par Renforcement

❑ Markov Reward Process (MRP)

Un **Processus de Récompense Markovien (MRP)** est défini par :

- S un ensemble fini d'états
- P la matrice de transition associée
- R la récompense moyenne associée à chaque état :

$$R_s = E[R_t / S_t = s]$$

R_{t+1}

- ц un facteur de réduction temporel

Apprentissage Par Renforcement

❑ Markov Decision Process

Un Processus de décision est un MRP auquel on rajoute les actions que peut prendre l'agent.

Un **Processus de Décision Markovien (MDP)** est défini par :

S un ensemble fini d'états

A un ensemble fini d'actions

P la matrice de transition d'un couple (s, a) 2 $S \twoheadrightarrow A$ vers S :

$$P^{a}_{ss'} = P[S_{t+1} = s' / S_t = s, A_t = a]$$

- R la récompense moyenne associée à chaque couple état - action :

$$R^{a}_{s} = E[R_t / S_t = s, A_t = a]$$

- цun facteur de réduction temporel

Apprentissage Par Renforcement

- **Méthodes d'apprentissage par renforcement**

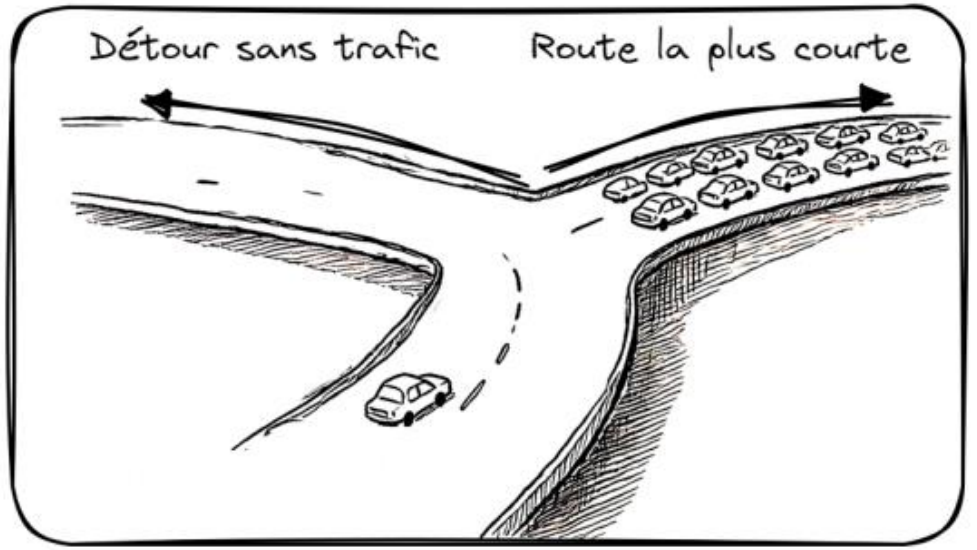
  Ce sont des **algorithmes** qui apprennent à agir dans un MDP.

Exemples :

- Q-Learning
- SARSA
- Policy Gradient
- Deep Q-Network (DQN)

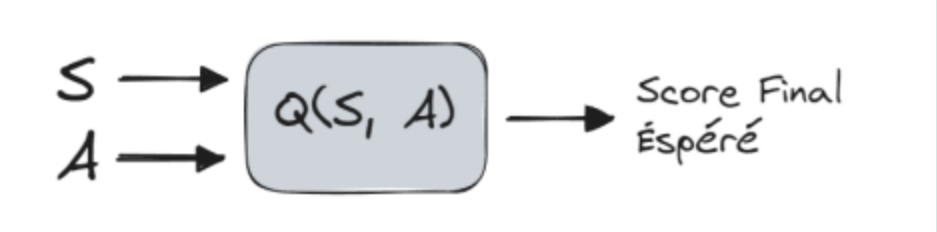
Apprentissage Par Renforcement

- Le fonctionnement du Q-Learning Le Q-Learning consiste à apprendre quelle est la valeur d'une action A dans un état S donné.
- Par exemple, je mets habituellement 30 minutes en voiture pour aller à mon travail. Aujourd'hui, cette route est encombrée par le trafic.
- cependant qu'une autre route, un peu plus longue, est également possible.
- Deux choix s'offrent : l'action de tourner à gauche ou à droite.
- Pour choisir laquelle emprunter, je vais évaluer la valeur de chacune



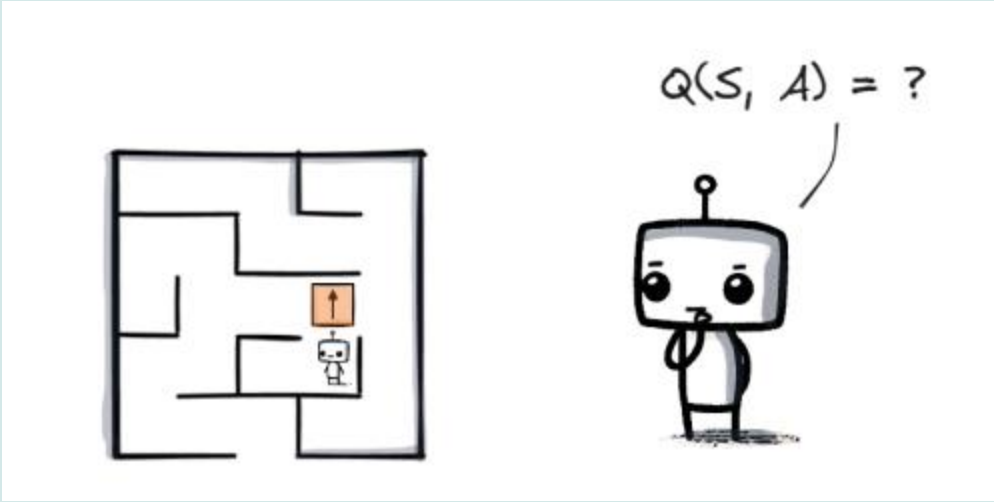
Apprentissage Par Renforcement

- Apprendre une fonction Q qui prend en entrée un état S et une action A, et qui prédit le score final que l'on peut espérer obtenir si tout se passe pour le mieux par la suite.

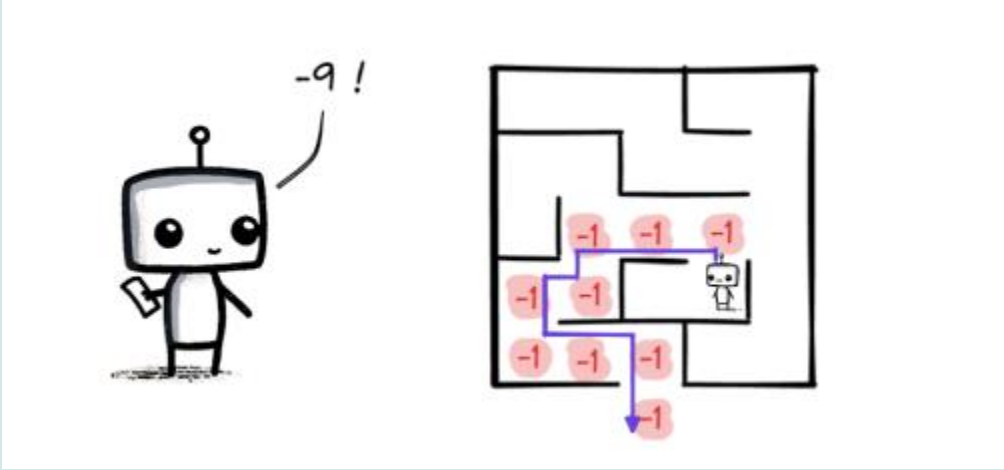


Apprentissage Par Renforcement

- Dans la position actuelle, quel est le meilleur score que la machine puisse espérer obtenir si elle choisit l'action "allez en haut" ?

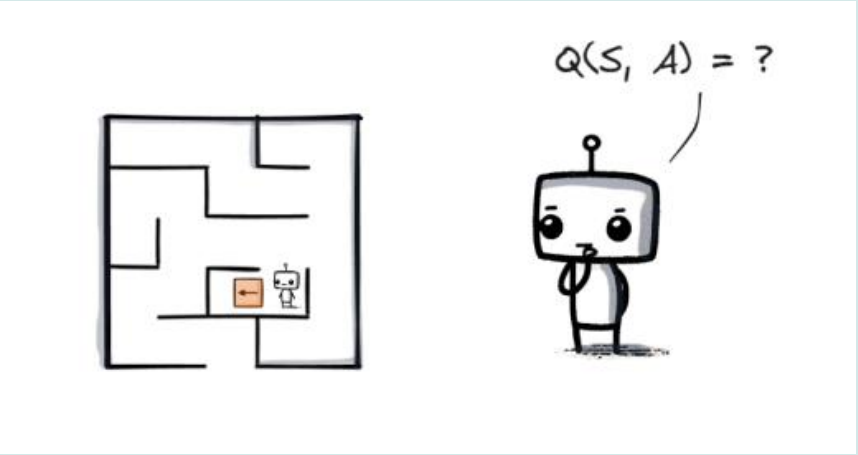


- En tenant compte du fait que la machine perd un point pour chaque déplacement, le meilleur score qu'elle puisse espérer obtenir est -9.



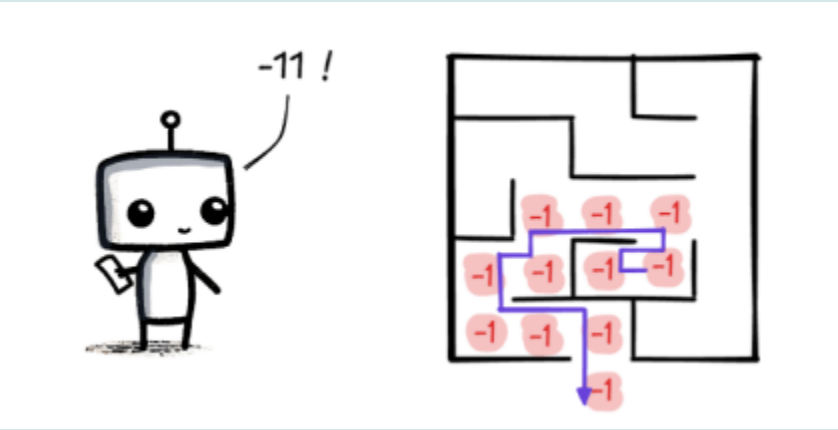
Apprentissage Par Renforcement

❑ Et dans le cas où la machine choisit d'aller à gauche ?



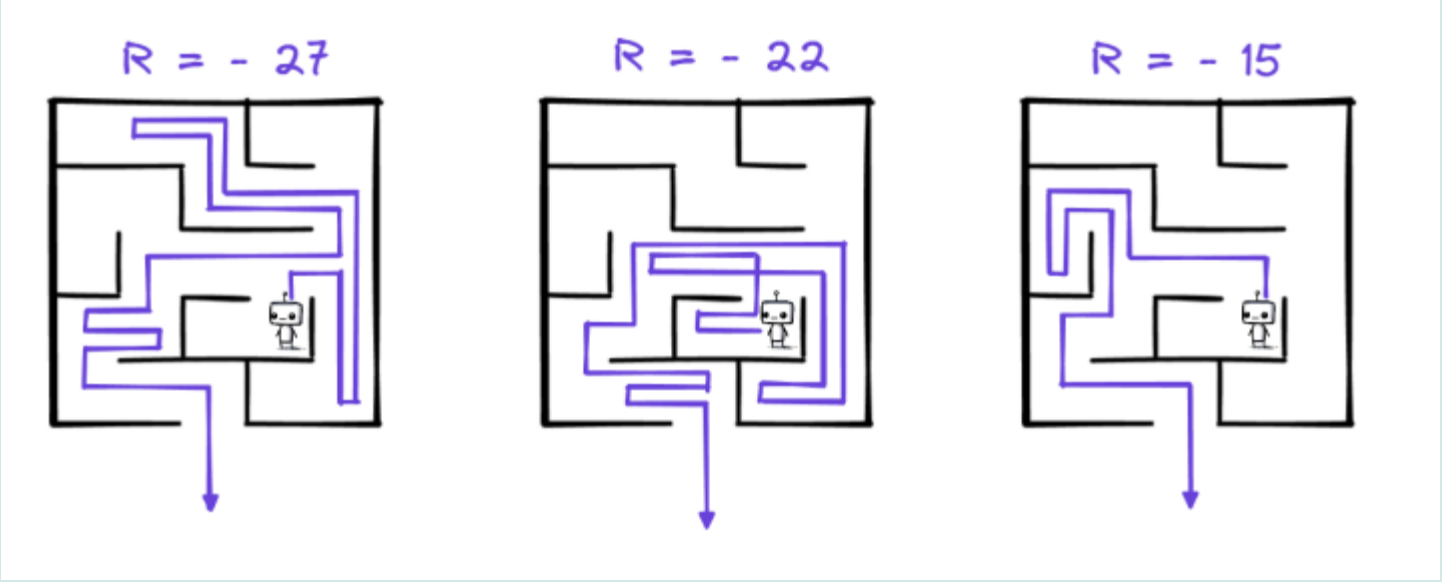
❑ -11, car la machine devra revenir sur son chemin.

Ainsi, il est préférable de choisir l'action "aller en haut" car elle donne un score final plus élevé (-9) que l'action "aller à gauche" (-11)



Apprentissage Par Renforcement

Pour l’apprentissage de la fonction Q Pour apprendre à prédire la valeur de ses actions, la machine doit explorer son environnement en s’y déplaçant au hasard, afin de générer des données (S, A, R) (état, action, récompense).



Apprentissage Par Renforcement

- Ces données sont ensuite utilisées pour mettre à jour un tableau Q(S, A) qui informe la machine des récompenses qu'elle est censée obtenir à l'avenir en choisissant telle ou telle action dans l'état présent.
- Au début de son apprentissage, ce tableau est rempli de valeurs aléatoires
- Une formule, connue sous le nom de l'équation de Bellman, permet d'exprimer le score total que l'on peut espérer obtenir à l'avenir.
- Ce score est égal à la récompense obtenue immédiatement dans l'état S, à laquelle on ajoute le même score pour l'état suivant S', si l'on choisit la meilleure action possible dans ce nouvel état.