

MODULE : IAD & AI

L'Intelligence Artificielle Distribuée & Agent Intelligent

Master 1 : Sciences 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.**

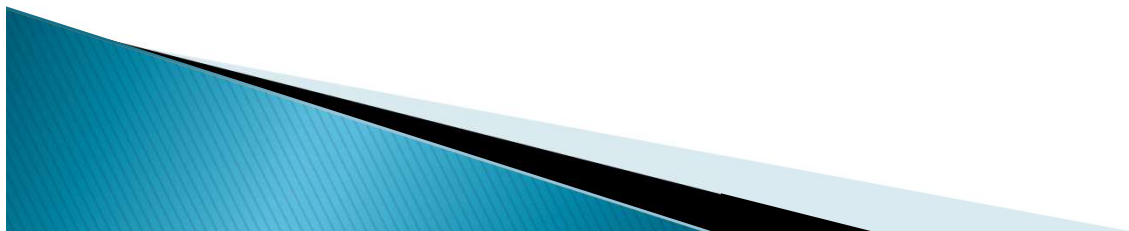
- Fonction de Valeur (Value Function).

- Algorithmes d'Agent :

- L'Algorithme : Q-Learning.
- L'Algorithme : Deep Q-Learning.
- L'Algorithme : Policy Gradient.

L'Apprentissage par Renforcement

- Les algorithmes d'apprentissage par renforcement comprennent souvent des éléments tels que :
 - **Des politiques** : stratégies que l'agent suit pour prendre des décisions.
 - **Des valeurs d'état** : estimations de la "valeur" de chaque état de l'environnement.
 - **Des fonctions de récompense** : déterminant les récompenses ou les pénalités associées à chaque action ou état.



L'Apprentissage par Renforcement

► Concepts Fondamentaux :

- **Environnement** : Le monde avec lequel l'agent interagit, défini par un ensemble d'états et d'actions possibles.
- **Agent** : L'entité qui prend des décisions dans l'environnement pour maximiser une récompense.
- **Récompense** : Un signal de retour fourni par l'environnement pour évaluer la qualité des actions prises par l'agent.
- **Politique** : La stratégie ou le plan que l'agent suit pour choisir des actions dans un état donné.

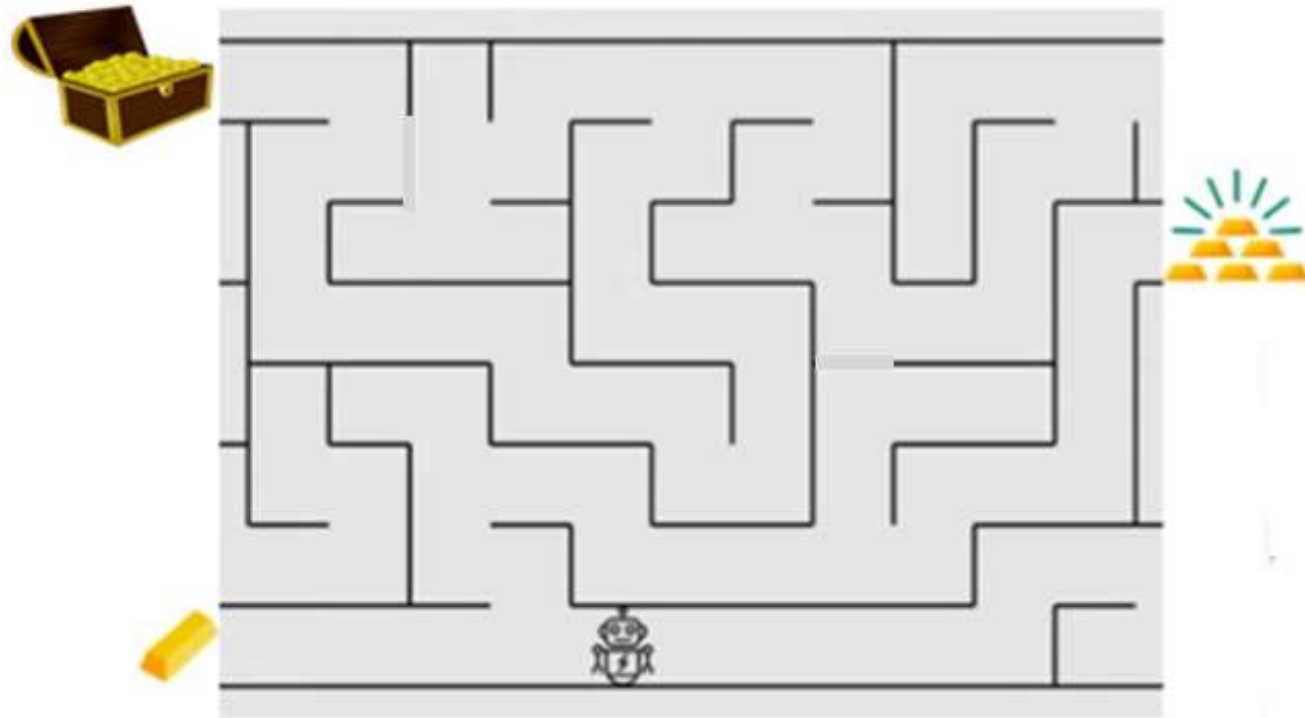


Exploration & Exploitation

- Ce sont deux aspects fondamentaux de l'apprentissage par renforcement, jouant un rôle crucial dans la manière dont un agent apprend à prendre des décisions dans un environnement donné.
- En équilibrant habilement l'exploration et l'exploitation, l'agent apprend à **maximiser sa récompense** cumulative dans son environnement.
- Une **exploration excessive peut retarder** la convergence vers une solution optimale, tandis qu'une **exploitation excessive peut conduire à une stagnation précoce**.
- Trouver le bon équilibre entre ces deux aspects est crucial pour le succès de l'apprentissage par renforcement.



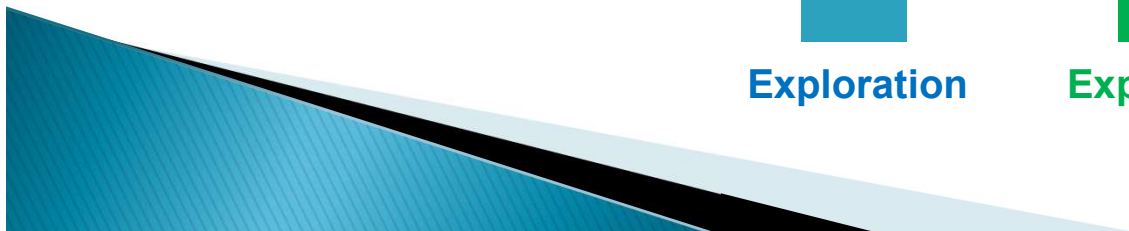
Exploration & Exploitation



Exploration



Exploitation

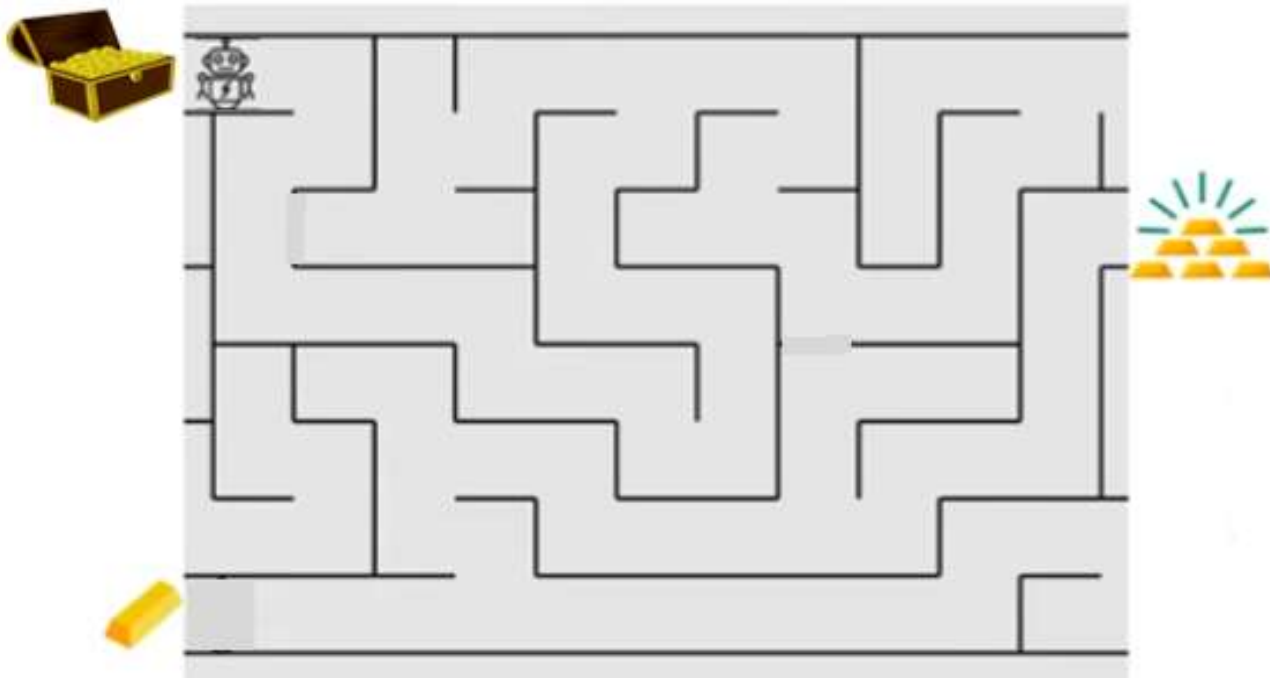


Exploration

- ▶ L'exploration est le processus par lequel l'agent découvre de nouvelles informations sur son environnement en essayant des actions qui **ne sont pas nécessairement optimales** selon sa connaissance actuelle.
- ▶ Cela peut sembler contre-intuitif, car l'agent **sacrifie des récompenses immédiates** pour explorer des régions inconnues de l'espace d'action. Cependant, **sans exploration**, l'agent risque de **rester coincé** dans des politiques sous-optimales, car il n'aurait jamais découvert des actions plus rentables.



Exploration



Exploration

Exploitation



Stratégies de l'Exploration

- **UCB (Upper Confidence Bound)** : Cette approche privilégie les actions qui ont un potentiel de récompense élevé mais qui n'ont pas été suffisamment explorées. Elle utilise une forme de calcul de l'incertitude pour guider l'exploration vers des actions dont la récompense est moins certaine.
- **Thompson Sampling** : Une approche probabiliste où l'agent choisit les actions en fonction d'une distribution de probabilité sur les récompenses des actions. L'agent explore des actions avec des récompenses élevées selon la distribution tout en continuant à mettre à jour ses croyances sur les récompenses.



Stratégies de l'Exploration

- **Stratégie Aléatoire** : L'agent choisit ses actions de manière totalement aléatoire, explorant ainsi de manière uniforme l'espace des actions. Cette approche est simple mais peut être inefficace car elle n'apprend pas des actions qui semblent prometteuses.
- **ϵ -Greedy** : C'est l'une des stratégies les plus courantes. L'agent choisit l'action optimale avec une probabilité élevée $(1-\epsilon)$ et une action aléatoire avec une probabilité ϵ . Cela permet à l'agent d'exploiter les actions connues tout en explorant de nouvelles actions avec une probabilité ϵ .



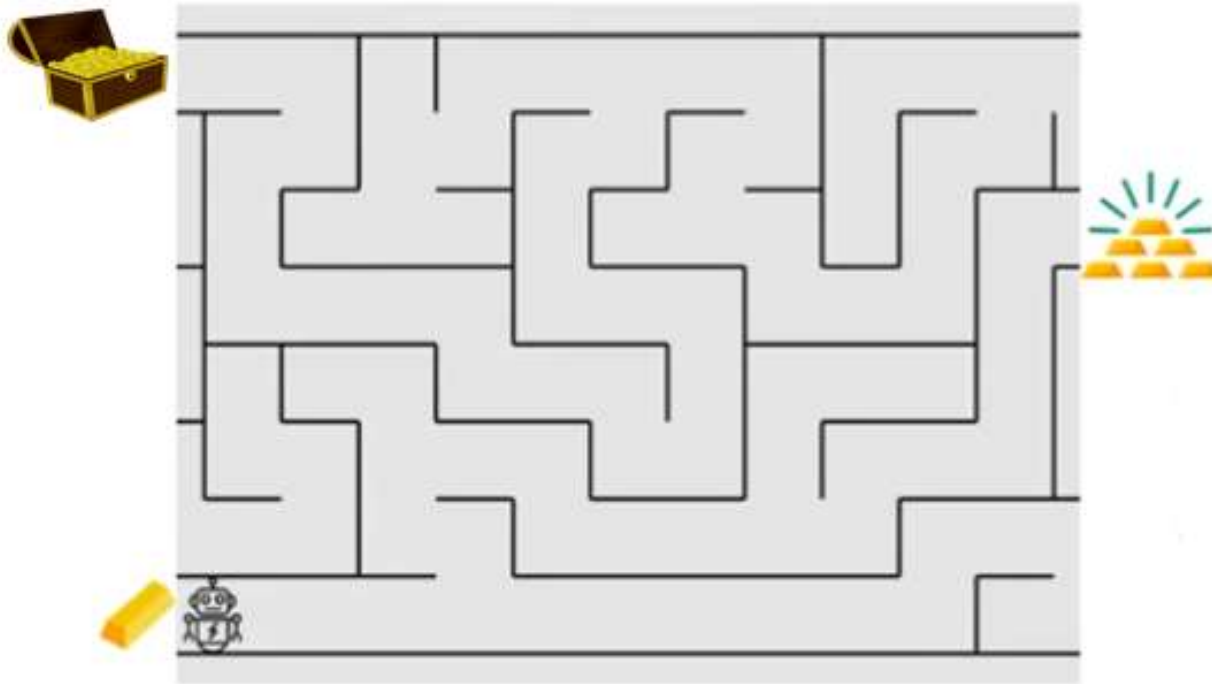
Exploitation

L'exploitation consiste à utiliser les **informations déjà acquises** pour **maximiser la récompense immédiate**. Une fois que l'agent a acquis une certaine connaissance de son environnement, il utilise cette connaissance pour prendre des décisions qui lui rapportent la plus grande récompense possible.


Il est important de noter que l'**exploitation seule peut conduire à une stagnation précoce** dans une politique sous-optimale, car l'**agent pourrait manquer de nouvelles opportunités** plus intéressantes.

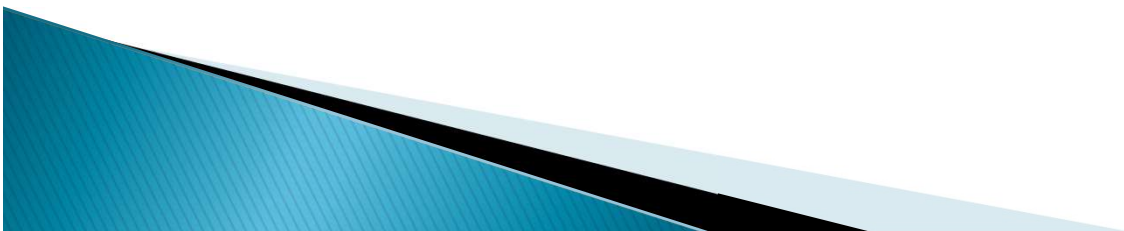


Exploitation




Exploration


Exploitation



Stratégies de l'Exploitation

- **Greedification** : Dans cette stratégie, l'agent choisit simplement l'action avec la récompense immédiate la plus élevée selon sa connaissance actuelle. C'est une approche purement exploitante qui peut entraîner une stagnation dans des politiques sous-optimales.
- **Exploitation de la Connaissance** : L'agent utilise des connaissances préalablement acquises sur l'environnement pour prendre des décisions. Cela peut inclure des politiques apprises précédemment ou des règles expertes fournies par un humain.
- **Stratégies d'Optimisation de Politique** : Ces approches visent à améliorer progressivement la politique de l'agent en fonction de sa performance passée. Cela peut impliquer l'utilisation de techniques telles que la descente de gradient pour ajuster les paramètres de la politique afin de maximiser la récompense cumulative.

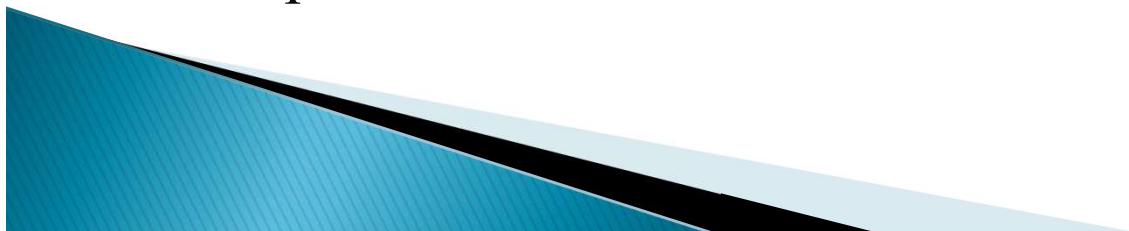


Exploitation



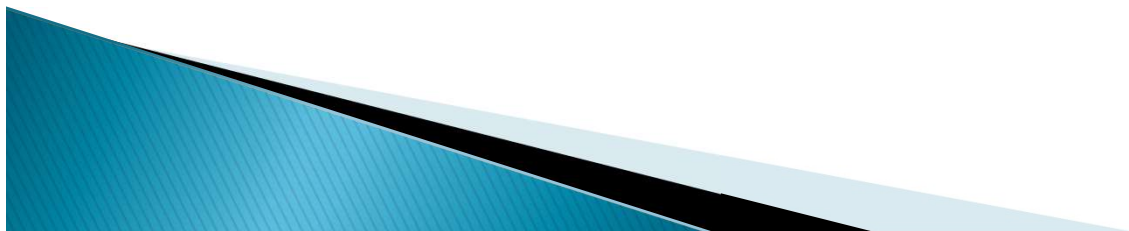
Exploration

- ▶ Trouver le bon équilibre entre exploration et exploitation est un défi majeur en apprentissage par renforcement.
- ▶ Une politique **trop exploratoire peut retarder la convergence vers une solution optimale**, tandis qu'une politique trop exploiteuse peut **entraîner une stagnation dans une solution sous-optimale**.
- ▶ Les stratégies telles que l' ϵ -Greedy, UCB et Thompson Sampling sont des exemples de tentatives pour équilibrer ces deux aspects de manière efficace.
- ▶ Aller vers des techniques qui permettent à l'agent d'apprendre rapidement et de manière efficace tout en maximisant sa récompense cumulative.



L'Algorithme ϵ -Greedy

- ▶ L'objectif de l'algorithme ϵ -Greedy est de trouver un équilibre entre exploration et exploitation.
- ▶ Lorsque ϵ est proche de 0, l'agent privilégie l'exploitation, c'est-à-dire qu'il choisit les actions qui semblent être les meilleures en fonction de sa connaissance actuelle.
- ▶ À l'inverse, lorsque ϵ est proche de 1, l'agent privilégie l'exploration, c'est-à-dire qu'il choisit des actions au hasard pour explorer de nouvelles possibilités.



L'Algorithme ϵ -Greedy

- ▶ Le choix de la valeur de ϵ est crucial dans cet algorithme.
- ▶ Si ϵ est trop élevé, l'agent explore trop et risque de ne pas tirer parti des actions les plus prometteuses qu'il a déjà découvertes.
- ▶ Si ϵ est trop faible, l'agent n'explore pas suffisamment et risque de manquer des actions potentiellement meilleures.
- ▶ Trouver le bon équilibre dépend souvent du problème spécifique et peut nécessiter des expérimentations pour déterminer la valeur optimale de ϵ .



L'Algorithme ϵ -Greedy

1. **Initialisation** : Lorsque l'agent commence son apprentissage, il définit un paramètre ϵ (epsilon), qui détermine la proportion d'actions qui seront choisies de manière aléatoire plutôt que selon la politique actuelle.
2. **Choix de l'action** : À chaque étape, l'agent doit choisir une action à exécuter. Il génère un nombre aléatoire entre 0 et 1. Si ce nombre est inférieur ou égal à ϵ , l'agent choisit une action aléatoire parmi toutes les actions possibles. Sinon, il choisit l'action avec la plus grande valeur selon la politique actuelle (exploitation).



L'Algorithme ϵ -Greedy

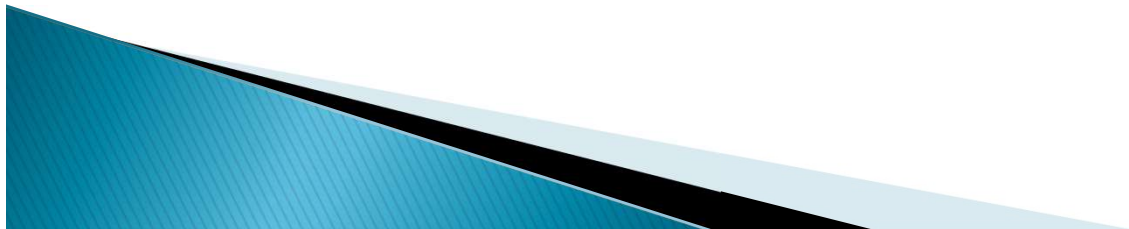
3. **Mise à jour de la politique** : Après avoir exécuté une action et reçu une récompense de l'environnement, l'agent met à jour ses estimations de la valeur des actions et peut ajuster sa politique en conséquence.

Cette mise à jour peut se faire selon différentes méthodes, telles que la mise à jour de la valeur Q dans le cas de l'apprentissage par renforcement basé sur les valeurs.



Choix de ϵ

- 1. Exploration initiale élevée :** Au début de l'apprentissage, il peut être bénéfique d'avoir une exploration élevée pour permettre à l'agent de découvrir différentes actions et stratégies. Ainsi, ϵ peut être initialement fixé à une valeur élevée, comme 0.9 ou 0.8.
- 2. Décroissance exponentielle :** Une approche courante consiste à diminuer ϵ au fil du temps à mesure que l'agent gagne de l'expérience. Par exemple, ϵ peut être réduit exponentiellement à chaque épisode ou à chaque pas de temps. Cette décroissance peut être contrôlée par un paramètre tel que le taux de décroissance.
- 3. Décroissance linéaire :** ϵ peut également être réduit linéairement à chaque épisode ou à chaque pas de temps. Par exemple, on peut définir un taux de décroissance fixe pour diminuer ϵ progressivement jusqu'à un certain seuil, en dessous duquel il reste constant.



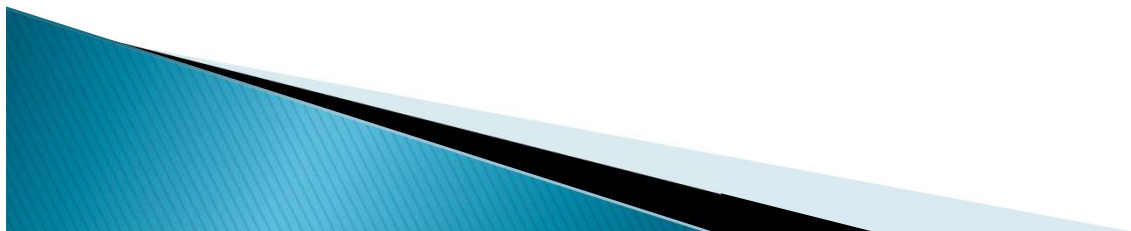
Choix de ϵ

4. **Planification adaptative** : Certains algorithmes adaptent dynamiquement la valeur de ϵ en fonction de la performance de l'agent. Par exemple, si l'agent commence à bien performer, ϵ peut être réduit plus rapidement pour privilégier l'exploitation. À l'inverse, si l'agent semble mal performer, ϵ peut être maintenu à un niveau plus élevé pour encourager davantage d'exploration.
5. **Paramètres de l'environnement** : Dans certains cas, des informations spécifiques sur l'environnement ou le problème peuvent guider le choix de ϵ . Par exemple, si l'environnement est complexe ou comporte de nombreuses actions, il peut être utile d'avoir une exploration plus élevée.



Choix de ϵ

- ▶ En pratique, le choix de ϵ **dépend souvent d'expérimentations** et de tests sur l'environnement spécifique et **peut nécessiter des ajustements au fil du temps** en fonction des performances de l'agent.
- ▶ L'objectif est de **trouver un équilibre entre exploration et exploitation** qui permette à l'agent d'apprendre efficacement et de maximiser les récompenses à long terme.



Exemple

- ▶ Supposons le fameux problème de bandits multi-bras où un agent doit choisir parmi 5 bras différents pour maximiser sa récompense cumulative.
- ▶ L'agent utilise l'algorithme ϵ -Greedy avec $\epsilon = 0.2$ pour gérer l'exploration et l'exploitation.



Exemple

1. Calcul des estimations des valeurs Q pour chaque bras :

- Pour le bras 1 : Le nombre de fois où il a été choisi est de 300, et sa récompense moyenne est de 3. Donc,
 $(1) = 300/1000 = 0.3$ $Q(1) = 1000/300 = 0.3$.
- Pour le bras 2 : $(2) = 200/1000 = 0.2$ $Q(2) = 1000/200 = 0.2$.
- Pour le bras 3 : $(3) = 150/1000 = 0.15$ $Q(3) = 1000/150 = 0.15$.
- Pour le bras 4 : $(4) = 100/1000 = 0.1$ $Q(4) = 1000/100 = 0.1$.
- Pour le bras 5 : $(5) = 250/1000 = 0.25$ $Q(5) = 1000/250 = 0.25$.



Exemple

1. Calcul des probabilités d'exploration et d'exploitation :

- Probabilité d'exploration : $\epsilon = 0.2$, donc la probabilité que l'agent choisisse une action au hasard est de 0.2.
- Probabilité d'exploitation : $1 - \epsilon = 1 - 0.2 = 0.8$, donc la probabilité que l'agent choisisse l'action exploitante est de 0.8.



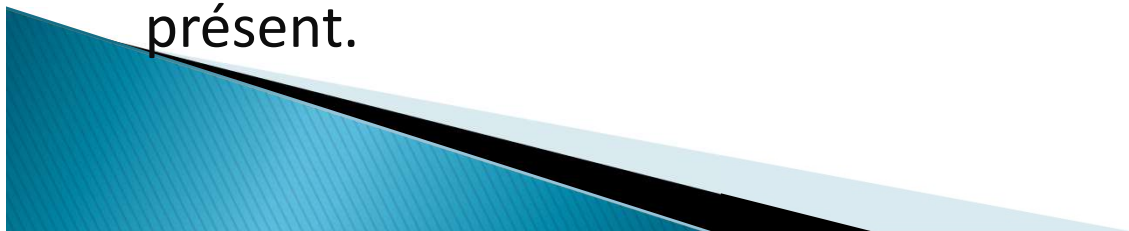
Exemple

- ▶ Chaque bras génère une récompense aléatoire selon une distribution spécifique et l'objectif de l'agent est d'apprendre à choisir les bras qui offrent les récompenses les plus élevées.
- ▶ On applique : ϵ -Greedy avec $\epsilon = 0.2$ pour gérer l'exploration et l'exploitation dans ce contexte :

1. Initialisation :

L'agent initialise une table Q qui stocke les valeurs Q pour chaque bras. Au début, toutes les valeurs Q sont initialisées à zéro.

L'agent initialise également un compteur pour chaque bras, indiquant le nombre de fois que ce bras a été sélectionné jusqu'à présent.



Exemple

2. Interaction avec l'environnement :

- À chaque étape, l'agent doit choisir un bras à tirer.
- L'agent utilise la stratégie ϵ -Greedy pour choisir le bras :
 - Avec une probabilité $\epsilon = 0.2$, l'agent choisit un bras aléatoire parmi les bras disponibles (exploration).
 - Avec une probabilité $1 - \epsilon = 0.8$, l'agent choisit le bras avec la plus grande valeur Q (exploitation).
- Une fois le bras choisi, l'agent tire ce bras et observe la récompense générée.



Exemple

3. Mise à jour de la valeur Q :

- Après avoir tiré un bras et reçu une récompense, l'agent met à jour la valeur Q pour ce bras en utilisant la formule de mise à jour de Q-Learning.
- Par exemple, si le bras i a été choisi et a généré une récompense r , la mise à jour de la valeur Q pour ce bras serait : $Q_i = Q_i + \alpha \cdot (r - Q_i)$
- où α est le **taux d'apprentissage**, qui peut être une valeur fixe ou décroissante avec le temps.



Exemple

4. Répéter :

- Ce processus se répète pour un certain nombre d'étapes ou jusqu'à ce que l'agent **atteigne un critère d'arrêt** prédéfini, tel qu'un **nombre fixe d'itérations** ou **une récompense cumulée suffisamment élevée**.
- En utilisant cette approche, l'agent peut apprendre à **sélectionner les bras les plus prometteurs** tout en **explorant de nouveaux bras** pour améliorer ses estimations de leur valeur.
- Avec le temps, l'agent devrait converger vers une politique qui maximise la récompense cumulative à long terme.





***Merci pour votre
attention ..***