# 1. Introduction

(à écrire à la fin)

# 2. Apprentissage Par Renforcement.

L'apprentissage par renforcement consiste à apprendre ce qu'il faut faire et comment associer des situations à des actions, de manière à maximiser un signal de récompense numérique. On ne dit pas à l'apprenant quelles actions il doit entreprendre, comme dans la plupart des formes d'apprentissage automatique, mais il doit découvrir les actions qui rapportent le plus en les essayant. Dans les cas les plus intéressants et les plus difficiles, les actions peuvent affecter non seulement la récompense immédiate, mais aussi la situation suivante et, à travers elle, toutes les récompenses ultérieures. Ces caractéristiques, recherche par essais et erreurs et récompense différée sont les plus importantes distinctives de l'apprentissage par renforcement.

L'apprentissage par renforcement est défini non pas en caractérisant les méthodes d'apprentissage, mais en caractérisant un problème d'apprentissage. Toute méthode qui est bien adaptée à la résolution de ce problème est considérée comme une méthode d'apprentissage par renforcement. L'idée de base est simplement de capturer les aspects les plus importants du problème réel auquel est confronté un agent d'apprentissage qui interagit avec son environnement pour atteindre un objectif. Il est clair qu'un tel agent doit être capable de détecter l'état de l'environnement dans une certaine mesure et doit être capable de prendre des mesures qui affectent cet état. L'agent doit également avoir un ou plusieurs objectifs liés à l'état de l'environnement.

L'apprentissage par renforcement est différent d’autres types d'apprentissage automatique, par exemple l'apprentissage supervisé consiste à apprendre à partir d'exemples fournis par un superviseur externe bien informé. Il s'agit d'un type d'apprentissage important, mais il n'est pas suffisant pour apprendre à partir d'une interaction. Dans les problèmes interactifs, il est souvent peu pratique d'obtenir des exemples de comportement souhaité qui soient à la fois corrects et représentatifs de toutes les situations dans lesquelles l'agent doit agir. En territoire inconnu où l'on s'attend à ce que l'apprentissage soit le plus bénéfique, un agent doit être capable d'apprendre de sa propre expérience.

L'un des défis qui se pose dans l'apprentissage par renforcement et non dans d'autres types d'apprentissage est le compromis entre l'exploration et l'exploitation. Pour obtenir beaucoup de récompenses, un agent d'apprentissage par renforcement doit préférer les actions qu'il a essayées dans le passé et qui se sont avérées efficaces pour produire des récompenses. Mais pour découvrir ces actions, il doit essayer des actions qu'il n'a pas sélectionnées auparavant. L'agent doit exploiter ce qu'il sait déjà afin d'obtenir une récompense, mais il doit également explorer afin de faire de meilleures sélections d'actions à l'avenir. Le dilemme est que ni l'exploration ni l'exploitation ne peuvent être poursuivies exclusivement sans échouer dans la tâche. L'agent doit essayer une variété d'actions et favoriser progressivement celles qui semblent les meilleures. Dans une tâche stochastique, chaque action doit être essayée de nombreuses fois pour obtenir une estimation fiable de sa récompense attendue. Nous notons simplement que la question de l'équilibre entre l'exploration et l'exploitation ne se pose même pas dans l'apprentissage supervisé tel qu'il est généralement défini.

## 2.1. Environnement

Le problème de l'apprentissage par renforcement est censé être une formulation simple du problème de l'apprentissage par interaction pour atteindre un objectif. L'apprenant et le décideur est appelé l'agent. L'objet avec lequel il interagit, comprenant tout ce qui est extérieur à l'agent, est appelé l'environnement. Ils interagissent continuellement, l'agent choisissant des actions et l'environnement réagissant à ces actions et présentant de nouvelles situations à l'agent. L'environnement donne également lieu à des récompenses, des valeurs numériques spéciales que l'agent essaie de maximiser au fil du temps. Une spécification complète d'un environnement définit une tâche, une instance du problème de l'apprentissage par renforcement.

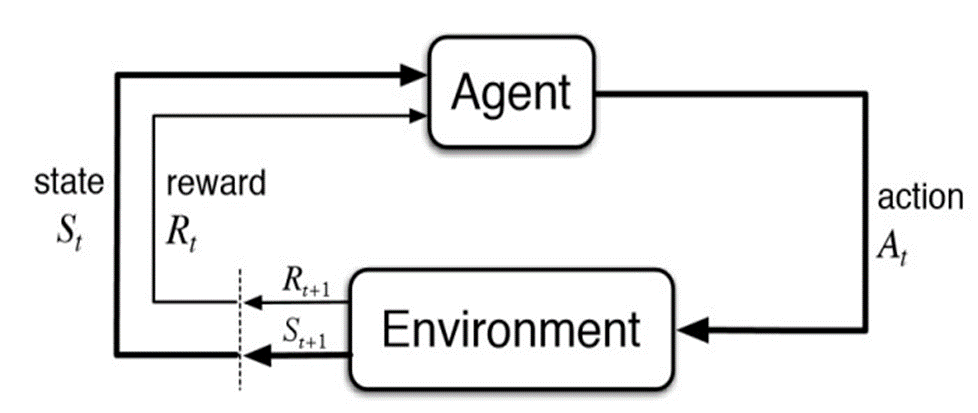


Figure 1 Interaction agent-environnement

Comme le montre la « figure1 », l'agent et l'environnement interagissent à chacun d'une séquence de pas de temps discrets, t = 0, 1, 2, 3, ... . À chaque pas de temps t, l'agent reçoit une représentation de l'état de l'environnement, st ∈ S, où S est l'ensemble des états possibles, et sur cette base, il choisit une action, at ∈ A(St), où A(St) est l'ensemble des actions disponibles dans l'état st. En conséquence de son action, l'agent reçoit une récompense numérique, rt+1, et se retrouve dans un nouvel état, st+1.

A chaque pas de temps, l'agent implémente une correspondance entre les états et les probabilités de sélection de chaque action possible. Cette correspondance est appelée la politique de l'agent et est notée πt, où πt(s,a)est la probabilité que at = a et st = s. Les méthodes d'apprentissage par renforcement précisent comment l'agent modifie sa politique à la suite de son expérience. L'objectif de l'agent, en gros, est de maximiser le montant total de la récompense qu'il reçoit sur le long terme.

Ce cadre est abstrait et flexible et peut être appliqué à de nombreux problèmes différents et de nombreuses manières différentes. Par exemple, les pas de temps ne doivent pas nécessairement faire référence à des intervalles fixes de temps réel ; ils peuvent faire référence à des étapes successives arbitraires de prise de décision et d'action. De même, les états peuvent prendre une grande variété de formes. Ils peuvent être entièrement déterminés par des sensations de bas niveau, comme les lectures directes de capteurs, ou être plus élevés et abstraits, comme les descriptions symboliques d'objets dans une pièce. Une partie de ce qui constitue un état peut être basée sur la mémoire de sensations passées. En général, les actions peuvent être toutes les décisions que l’agent veut apprendre à prendre.

La règle générale que nous suivons est que tout ce qui ne peut pas être modifié arbitrairement par l'agent est considéré comme extérieur à celui-ci et fait donc partie de son environnement. Nous ne supposons pas que tout ce qui se trouve dans l'environnement est inconnu de l'agent. Par exemple, l'agent en sait souvent assez long sur la façon dont ses récompenses sont calculées en fonction de ses actions et des états dans lesquels elles sont prises. Mais nous considérons toujours que le calcul des récompenses est externe à l'agent car il définit la tâche à laquelle l'agent est confronté et doit donc être au-delà de sa capacité à changer arbitrairement. En fait, dans certains cas, l'agent peut tout savoir sur le fonctionnement de son environnement et être confronté à une tâche d'apprentissage par renforcement difficile. La frontière agent-environnement représente la limite du contrôle absolu de l'agent, et non de ses connaissances.

La frontière entre l'agent et l'environnement peut être située à différents endroits et à différentes fins. Dans un robot complexe, de nombreux agents différents peuvent fonctionner en même temps, chacun ayant sa propre limite. Par exemple, un agent peut prendre des décisions de haut niveau qui font partie des états auxquels est confronté un agent de niveau inférieur qui met en œuvre les décisions de haut niveau. En pratique, la limite agent-environnement est déterminée une fois que l'on a sélectionné des états, des actions et des récompenses particuliers, et que l'on a donc identifié une tâche décisionnelle spécifique.

Le cadre de l'apprentissage par renforcement est une abstraction considérable du problème de l'apprentissage dirigé vers un but à partir de l'interaction. Il propose que, quels que soient les détails de l'appareil sensoriel, de la mémoire et du contrôle, et quel que soit l'objectif que l'on essaie d'atteindre, tout problème d'apprentissage d'un comportement dirigé vers un but peut être réduit à trois signaux qui vont et viennent entre un agent et son environnement : un signal pour représenter les choix faits par l'agent (les actions), un signal pour représenter la base sur laquelle les choix sont faits (les états), et un signal pour définir le but de l'agent (les récompenses). Ce cadre n'est peut-être pas suffisant pour représenter utilement tous les problèmes d'apprentissage de la décision, mais il s'est avéré largement utile et applicable.

Bien sûr, les états et actions particuliers varient considérablement d'une application à l'autre, et la manière dont ils sont représentés peut fortement affecter les performances.

## 2.2. Objectifs et Récompenses

Dans l'apprentissage par renforcement, le but ou l'objectif de l'agent est formalisé en termes d'un signal de récompense particulier passant de l'environnement à l'agent. À chaque pas de temps, la récompense est un simple nombre, rt . L'objectif de l'agent est de maximiser le montant total de la récompense qu'il reçoit. Ceci Cela signifie maximiser non pas la récompense immédiate, mais la récompense cumulative à long terme.

L'utilisation d'un signal de récompense pour formaliser l'idée d'un objectif est l'une des caractéristiques les plus distinctives de l'apprentissage par renforcement. Bien que cette façon de formuler les objectifs puisse à première vue sembler restrictive, elle s'est avérée flexible et largement applicable dans la pratique. La meilleure façon de s'en rendre compte est d'examiner des exemples de la façon dont elle a été, ou pourrait être, utilisée. Par exemple, pour qu'un robot apprenne à marcher, les chercheurs ont prévu une récompense à chaque pas de temps, proportionnelle à la progression du robot. Pour qu'un robot apprenne à s'échapper d'un labyrinthe, la récompense est souvent nulle jusqu'à ce qu'il s'échappe, après quoi elle devient +1. Une autre approche courante dans l'apprentissage des labyrinthes consiste à donner une récompense de -1 pour chaque pas de temps qui précède l'évasion ; cela encourage l'agent à s'échapper aussi vite que possible. Pour qu'un robot apprenne à trouver et à collecter des canettes de soda vides pour les recycler, on peut lui donner une récompense de zéro la plupart du temps, puis une récompense de +1 pour chaque canette collectée (et confirmée comme vide). On pourrait également donner au robot des récompenses négatives lorsqu'il se cogne contre des objets ou lorsque quelqu'un lui crie dessus. Pour qu'un agent apprenne à jouer aux dames ou aux échecs, les récompenses naturelles sont +1 pour une victoire, -1 pour une défaite, et 0 pour un match nul et pour toutes les positions non terminales, dans tous ces exemples. L'agent apprend toujours à maximiser sa récompense.

## 2.3. Le Rendement

En général, nous cherchons à maximiser le rendement attendu, où le rendement, Rt, est défini comme une fonction spécifique de la séquence de récompenses. Dans le cas le plus simple, le rendement est la somme des récompenses :

où *T* est un pas de temps final. Cette approche est utile dans les applications où il existe une notion naturelle de pas de temps final, c'est-à-dire lorsque l'interaction agent-environnement se décompose naturellement en sous-séquences, que nous appelons épisodes. Chaque épisode se termine dans un état spécial appelé état terminal, suivi d'une réinitialisation à un état de départ standard ou à un échantillon d'une distribution standard d'états de départ. Les tâches avec des épisodes de ce type sont appelées tâches épisodiques.

D'autre part, dans de nombreux cas, l'interaction agent-environnement ne se décompose pas naturellement en épisodes identifiables, mais se poursuit continuellement sans limite. Par exemple, ce serait la façon naturelle de formuler une tâche de contrôle de processus continu, ou une application à un robot ayant une longue durée de vie. Nous appelons cela des tâches continues. La formulation du retour est problématique pour les tâches continues car le dernier pas de temps serait de T = ∞, et le retour, qui est ce que nous essayons de maximiser, pourrait facilement être infini.

Le concept supplémentaire dont nous avons besoin est celui de la remise. Selon cette approche, l'agent tente de sélectionner des actions de manière à maximiser la somme des récompenses actualisées qu'il reçoit dans le futur. En particulier, il choisit *at* pour maximiser le rendement attendu :

Où γ est un paramètre, 0 ≤ γ ≤ 1, appelé le facteur de réduction.

Un facteur de réduction peut être considéré comme un paramètre d'apprentissage, variant de 0 ≤ γ ≤ 1. Chaque récompense future au pas temporel *t* est actualisée par γ. Si γ s'approche de zéro, notre agent considère les récompenses proches de l'état actuel comme beaucoup plus précieuses. En revanche, un γ est proche de un change notre agent à considérer les récompenses futures comme tout aussi importantes.

## 2.4. Politique

Un comportement se traduit par une politique de contrôle, une carte déterminant l'action à entreprendre à chaque état. Une politique peut être déterministe, lorsque l'exécution d'une action est garantie, ou stochastique, lorsqu'une certaine probabilité est impliquée. Par conséquent, une politique déterministe est mathématiquement représentée par ce qui suit :

Si les actions sont stochastiques, la politique se transforme en une distribution de probabilité, alors l'information doit être incluse dans la politique :

Théoriquement, la politique rassemblant la plus grande quantité de récompenses pour un monde particulier est considérée comme une politique optimale, dénotée par π∗. De plus, après avoir appliqué un algorithme d'apprentissage couplé à une stratégie d'exploration appropriée jusqu'à convergence, compte tenu d'épisodes suffisants, la politique obtenue peut être considérée comme "optimale" ou "sous-optimale". Cette dernière est également référencée comme le principe d'optimalité de Bellman [1].

## 2.5. Value Function

Une fonction de valeur évalue l'utilité d'une politique, compte tenu d'un état, st ∈ S et en suivant la même politique π par la suite. L'utilité d'une politique comprend une somme de récompenses.

Nous définissons la valeur de l'action *a* dans l'état *s* sous une politique *π* comme le rendement espéré en partant de s, en prenant l'action a, et en suivant ensuite la politique π :

Où *Eπ* désigne la fonction de valeur espérée étant donné la politique π, et t est un pas de temps quelconque.

## 2.6 La propriété de Markov

Dans le cadre de l'apprentissage par renforcement, l'agent prend ses décisions en fonction d'un signal provenant de l'environnement, appelé état de l'environnement. La propriété de Markov [2] est une forme mathématiquement idéalisée du problème de l'apprentissage par renforcement pour lequel des déclarations théoriques précises peuvent être faites.

Une tâche d'apprentissage par renforcement qui satisfait à la propriété de Markov est appelée un processus de décision de Markov, ou PDM. Si les espaces d'état et d'action sont finis, on parle alors de processus de décision de Markov fini (PDM fini). Les PDM finis sont particulièrement importants pour la théorie de l'apprentissage par renforcement.

Un MDP fini particulier est défini par ses ensembles d'états et d'actions et par la dynamique à un pas de l'environnement. Pour tout état et action, et, la probabilité de chaque état suivant possible, s', est la suivante

C'est ce qu'on appelle les probabilités de transition. De même, étant donné tout état et action actuels, s et a, ainsi que tout état suivant, s', la valeur attendue de la récompense suivante est

and , spécifient complètement les aspects les plus importants de la dynamique d'un MDP fini .

## 2.7 Apprentissage par différence temporelle

L'apprentissage par différence temporelle [3] (ou TD) interpole les idées de la programmation dynamique [4] (DP) et des méthodes de Monte Carlo [5]. Les algorithmes TD sont capables d'apprendre directement à partir d'expériences brutes, sans modèle particulier de l'environnement. Alors que dans les méthodes de Monte Carlo, un épisode doit être complet pour mettre à jour une fonction de valeur, l'apprentissage par différence temporelle est capable d'apprendre (mettre à jour) la fonction de valeur dans chaque expérience (ou étape). Le prix à payer pour pouvoir modifier régulièrement la fonction de valeur est la nécessité de mettre à jour les estimations sur la base d'autres estimations apprises (idée de l'apprentissage par différence temporelle). Alors que dans la PD, un modèle de la dynamique de l'environnement est nécessaire, les approches Monte Carlo et TD sont plus adaptées aux tâches incertaines et imprévisibles. Comme la TD apprend de chaque transition (état, récompense, action, état suivant, récompense suivante), il n'est pas nécessaire d'ignorer ou de négliger certains épisodes comme dans les algorithmes de Monte Carlo.

## 2.8 Exploration / Exploitation

Le problème du bandit à plusieurs bras [6] est un problème dans lequel un ensemble limité de ressources doit être alloué entre des choix concurrents (alternatifs) de manière à maximiser leur gain attendu, lorsque les propriétés de chaque choix ne sont que partiellement connues au moment de l'allocation, et peuvent être mieux comprises au fil du temps, ou en allouant des ressources au choix. Il s'agit d'un problème classique d'apprentissage par renforcement qui illustre le dilemme du compromis exploration-exploitation. Le nom vient de l'imagination d'un joueur devant une rangée de machines à argent (parfois appelées "bandits manchots"), qui doit décider quelles machines jouer, combien de fois jouer chaque machine et dans quel ordre les jouer, et s'il faut continuer avec la machine actuelle ou essayer une autre machine. Le problème du bandit manchot entre également dans la vaste catégorie de l'ordonnancement stochastique.

Dans ce problème, chaque machine fournit une récompense aléatoire issue d'une distribution de probabilité spécifique à cette machine, qui n'est pas connue a-priori. L'objectif du joueur est de maximiser la somme des récompenses obtenues par une séquence de tirages de levier. Le compromis crucial auquel le joueur doit faire face à chaque essai est entre "l'exploitation" de la machine qui a le gain attendu le plus élevé et "l'exploration" pour obtenir plus d'informations sur les gains attendus des autres machines. Le compromis entre l'exploration et l'exploitation se pose également dans l'apprentissage automatique.

En fonction de la façon dont nous effectuons l'exploration, il existe plusieurs façons de résoudre le bandit à bras multiples, comme l'algorithme ε-Greedy [7], les limites supérieures de confiance [8] et l'échantillonnage de Thompson [9].

## 2.9. Conclusion

L'apprentissage par renforcement est une approche informatique visant à comprendre et à automatiser l'apprentissage et la prise de décision orientés vers un but. Il se distingue des autres approches informatiques par l'accent qu'il met sur l'apprentissage par l'individu à partir d'une interaction directe avec son environnement, sans s'appuyer sur une supervision exemplaire ou des modèles complets de l'environnement. À notre avis, l'apprentissage par renforcement est le premier domaine à aborder sérieusement les questions informatiques qui se posent lors de l'apprentissage à partir de l'interaction avec un environnement afin d'atteindre des objectifs à long terme.

L'apprentissage par renforcement utilise un cadre formel définissant l'interaction entre un agent d'apprentissage et son environnement en termes d'états, d'actions et de récompenses. Ce cadre est conçu comme un moyen simple de représenter les caractéristiques essentielles du problème de l'intelligence artificielle. Ces caractéristiques comprennent un sens de la cause et de l'effet, un sens de l'incertitude et du non-déterminisme, et l'existence de buts explicites.