ÉCOLE NATIONALE D'INGÉNIEURS DE TUNIS

Département informatique

Projet Bibliographique

Apprentissage par Renforcement et Domaines d'application

Elaboré par :

Boujemaa Lina

Masmoudi Sami

Encadré par :

Abir Gallas

1ère Année informatique

Année universitaire : 2022/2023

Remerciements

On voudrais d'abord exprimer notre gratitude envers notre professeur, tuteur et superviseur académique, Mme Abir Gallas, pour les conseils qu'elle nous a prodigués, pour les efforts qu'elle a déployés pour nous aider à saisir ce domaine d'expertise aussi rapidement que possible. Nous tenons également à la remercier de partager son savoir avec nous et de nous offrir de grandes opportunités tout au long du chemin. C'était vraiment un plaisir de travailler avec vous.

Enfin, nos remerciements vont à tous nos professeurs pour leur contribution infatigable à nous fournir des connaissances essentielles pour devenir des ingénieurs. Et ils vont également à toute l'administration et au personnel de l'école pour fournir de bonnes conditions d'apprentissage et essayer d'être aussi flexibles que possible.

Abstract

The end-of-year internship project is a comprehensive research work on reinforcement learning (RL). The project is divided into three chapters. The first chapter is an introduction to machine learning (ML) and the types of learning in particular. This chapter explores in detail reinforcement learning, including its history, objective, principle, as well as its advantages and disadvantages. The second chapter examines the methods and algorithms of reinforcement learning. It presents Markov decision processes, Q-learning, SARSA, and other important algorithms. This chapter also provides examples of practical applications of these methods. The third and final chapter explores the application domains of reinforcement learning and presents the major players in this field. Application examples include robotics, games, finance, online advertising, and medicine. Overall, this PFA project provides a comprehensive and practical introduction to reinforcement learning, as well as an overview of methods and applications in different domains.

Résumé

Notre projet PFA est un travail de recherche sur l'apprentissage par renforcement (RL). Le projet est divisé en trois chapitres. Le premier chapitre est une introduction à l'apprentissage automatique (ML) et aux types d'apprentissage en particulier. Ce chapitre explore en détail l'apprentissage par renforcement, y compris son historique, son objectif, son principe, ainsi que ses avantages et inconvénients. Le deuxième chapitre examine les méthodes et les algorithmes de l'apprentissage par renforcement. Il présente les processus de décision de Markov, Q-learning, SARSA et d'autres algorithmes importants. Ce chapitre fournit également des exemples d'applications pratiques de ces méthodes. Le troisième et dernier chapitre explore les domaines d'application de l'apprentissage par renforcement et présente les grands intervenants dans ce domaine. Les exemples d'applications comprennent la robotique, les jeux, la finance, la publicité en ligne et la médecine. Dans l'ensemble, ce projet de PFA fournit une introduction complète et pratique à l'apprentissage par renforcement, ainsi qu'un aperçu des méthodes et des applications dans différents domaines.

Table des matières

0.1 Liste des acronymes

RL Reinforcement Learning

ML Machine Learning

MDP Markov Decision Processes

IA Intelligence Artificielle

DQN Deep Q-Network

CNN Convolutional Neural Network

IRL Inverse Reinforcement Learning

SARSA State-Action-Reward-State-Action

Table des figures

Liste des tableaux

Introduction

L'Intelligence Artificielle (IA) et le Machine Learning (ML) sont des domaines en pleine expansion, avec des applications dans de nombreux secteurs tels que la santé, les transports, la finance et l'industrie. Les avancées en matière d'IA ont été alimentées par de nouveaux modèles de machine learning, tels que l'apprentissage supervisé et non supervisé, ainsi que l'apprentissage par renforcement.

L'apprentissage par renforcement est une approche de machine learning qui permet à une entité (un agent) d'apprendre à interagir avec un environnement en prenant des décisions qui maximisent une récompense. Ce type d'apprentissage est inspiré de la psychologie behavioriste et de la théorie de la prise de décision, et a des applications dans des domaines tels que la robotique, les jeux, la finance, la publicité en ligne et la santé.

Dans ce projet, nous nous concentrons sur l'apprentissage par renforcement et examinons comment il peut être utilisé pour améliorer les performances d'un agent dans un environnement donné. Nous discuterons des principaux concepts de l'apprentissage par renforcement, tels que les états, les actions, les récompenses, les politiques et les fonctions de valeur, ainsi que des algorithmes populaires utilisés dans ce domaine, tels que Q-learning, SARSA et Deep Q-Networks (DQN). Enfin, nous explorerons les applications récentes de l'apprentissage par renforcement dans différents domaines, ainsi que les défis et les perspectives d'avenir de cette approche de machine learning.

Chapitre 1

Apprentissage par renforcement

Dans ce chapitre, on va introduire les différents types d'apprentissage automatique et à l'histoire de l'apprentissage par renforcement.

L'apprentissage automatique est une branche de l'IA qui permet aux ordinateurs d'apprendre à partir de données. Les types d'apprentissage incluent l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement, qui remonte aux années 1950 et a été initialement développé pour les jeux. Au cours de ce chapitre, nous explorerons l'évolution de chaque type d'apprentissage, ainsi que les avancées clés qui ont conduit à la reconnaissance de l'apprentissage automatique comme une technologie essentielle pour l'avenir de l'IA.

1.1 L'apprentissage automatique

L'apprentissage automatique [1] est un domaine l'étude de l'intelligence artificielle,qui permet à des ordinateurs d'apprendre des modèles à partir de données. C'est le processus par lequel un système informatique apprend à collecter des données, plutôt que d'être programmé pour effectuer une tâche spécifique.
L'apprentissage automatique consiste à utiliser des techniques statistiques pour permettre à un ordinateur d'identifier des modèles et des relations dans les données. Les algorithmes d'apprentissage automatique sont conçus pour apprendre de ces modèles et améliorer leurs performances sur une tâche donnée au fil du temps en fonction des données d'apprentissage et de validation. Cela implique l'utilisation d'algorithmes et de modèles mathématiques pour découvrir des relations cachées dans les données, les tendances et les modèles qui peuvent être utilisés pour résoudre des problèmes dans divers domaines.

1.1.1 Avantages et Inconvénients

L'apprentissage automatique a vu des cas d'utilisation allant de la prédiction du comportement des clients à la création du système d'exploitation pour les voitures autonomes.

En ce qui concerne les avantages [22], le machine learning peut aider les entreprises à comprendre leurs clients à un niveau plus profond. En collectant les données des clients et en les corrélant avec leurs comportements au fil du temps, les algorithmes de machine learning peuvent apprendre des associations et aider les équipes à adapter le développement de produits et les initiatives marketing à la demande des clients.

Certaines entreprises utilisent le machine learning comme moteur principal de leur modèle d'entreprise. Par exemple, Uber utilise des algorithmes pour associer les conducteurs aux passagers. Google utilise le machine learning pour afficher des publicités lors de recherches.

Mais le machine learning présente des inconvénients [22]. Tout d'abord, cela peut être coûteux. Les projets de machine learning sont généralement pilotés par des data scientists, qui ont des salaires élevés. Ces projets nécessitent également une infrastructure logicielle qui peut être coûteuse.

Il y a aussi le problème de la partialité du machine learning. Les algorithmes formés sur des ensembles de données qui excluent certaines populations ou qui contiennent des erreurs peuvent conduire à des modèles inexactes du monde qui, au mieux, échouent et, au pire, sont discriminatoires. Lorsqu'une entreprise base ses processus d'affaires sur des modèles biaisés, elle peut rencontrer des problèmes réglementaires et de réputation.

1.1.2 Les types d'apprentissage

Il existe trois types d'apprentissage automatique : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

Apprentissage supervisé

L'apprentissage supervisé [1] est un type d'apprentissage automatique où un algorithme est entraîné à partir de données étiquetées. L'objectif est de trouver une relation entre les données d'entrée et les étiquettes associées, afin de pouvoir prédire l'étiquette pour de nouvelles données d'entrée. L'étape d'apprentissage se fait en utilisant un ensemble de données d'entraînement pré-étiqueté.

Cela signifie que les données d'entrée sont associées à des étiquettes de sortie connues, et l'algorithme apprend à associer correctement les entrées aux sorties en se basant sur les exemples fournis dans le jeu de données d'entraînement. L'objectif

est de créer un modèle qui peut prédire avec précision les sorties pour de nouvelles entrées qui n'ont pas encore été vues. Dans l'apprentissage supervisé, le jeu de données est divisé en deux parties : un ensemble de données d'entraînement et un ensemble de données de test. Le modèle est entraîné sur l'ensemble de données d'entraînement, qui contient les exemples étiquetés, afin d'apprendre à prédire les sorties pour de nouvelles entrées. Ensuite, le modèle est évalué sur l'ensemble de données de test pour évaluer sa capacité à généraliser les connaissances acquises lors de l'apprentissage.

Apprentissage non supervisé

L'apprentissage non supervisé [1] est une méthode où un modèle est entraîné pour identifier des modèles, des structures et des relations dans un ensemble de données sans être supervisé par des étiquettes ou des réponses pré-établies. Contrairement à l'apprentissage supervisé, où le modèle est entraîné à prédire une sortie en utilisant des exemples étiquetés, l'apprentissage non supervisé ne reçoit pas de rétroaction directe pour ajuster ses prédictions. Au lieu de cela, il doit trouver des modèles dans les données en explorant les relations et les différences entre les exemples d'entrée. Les techniques courantes d'apprentissage non supervisé comprennent la classification non supervisée, la réduction de la dimensionnalité, la détection d'anomalies et le clustering. La classification non supervisée implique la catégorisation des exemples d'entrée en groupes homogènes en fonction de leurs caractéristiques, tandis que la réduction de la dimensionnalité vise à réduire la complexité des données en extrayant les caractéristiques les plus importantes. La détection d'anomalies consiste à identifier des exemples qui diffèrent considérablement de la majorité des autres exemples, tandis que le clustering regroupe les exemples en fonction de leur similarité. L'apprentissage non supervisé est souvent utilisé dans des domaines tels que l'analyse de données, l'exploration de données, la reconnaissance de motifs et la segmentation de données. Il est également souvent utilisé pour pré-traiter les données avant l'apprentissage supervisé.

Apprentissage semi-supervisé

L'apprentissage semi-supervisé [1] est un type d'apprentissage automatique qui combine des éléments de l'apprentissage supervisé et de l'apprentissage non supervisé. Contrairement à l'apprentissage supervisé, où l'algorithme apprend à partir de données étiquetées, et à l'apprentissage non supervisé, où l'algorithme apprend à partir de données non étiquetées, l'apprentissage semi-supervisé utilise à la fois des données étiquetées et non étiquetées.

L'objectif de l'apprentissage semi-supervisé est de maximiser la performance de l'algorithme en utilisant un ensemble de données plus large et plus diversifié.

En effet, les données étiquetées sont souvent rares et coûteuses à collecter, tandis que les données non étiquetées sont plus faciles à obtenir. En utilisant les deux types de données, l'apprentissage semi-supervisé permet d'améliorer la précision et la généralisation de l'algorithme, tout en réduisant les coûts de collecte et de marquage des données.

L'apprentissage semi-supervisé est couramment utilisé dans des domaines tels que la reconnaissance d'image, la classification de texte, l'analyse de réseau et la bioinformatique.

Cependant, l'apprentissage semi-supervisé peut également présenter des défis et des limites, tels que la nécessité de traiter des données manquantes ou bruyantes, la sélection appropriée des données étiquetées et non étiquetées, et la difficulté de quantifier la contribution relative de chaque type de données à la performance de l'algorithme. En fin de compte, l'utilisation de l'apprentissage semi-supervisé dépendra des spécificités du problème de l'apprentissage automatique et des ressources.

Apprentissage par renforcement

L'apprentissage par renforcement [1] est une méthode où un agent apprend à prendre des décisions en interagissant avec un environnement. L'agent prend des actions dans l'environnement et reçoit des récompenses ou des pénalités en fonction des résultats de ces actions. L'objectif de l'agent est de maximiser la somme cumulée des récompenses qu'il reçoit au fil du temps. L'apprentissage par renforcement utilise une boucle de rétroaction continue pour permettre à l'agent d'apprendre de ses actions. L'agent utilise une stratégie pour décider quelle action prendre en fonction de l'état actuel de l'environnement. Une fois l'action prise, l'environnement passe à un nouvel état et l'agent reçoit une récompense ou une pénalité en fonction de cet état. L'agent ajuste alors sa stratégie pour mieux correspondre aux résultats obtenus et prend une nouvelle action. Les techniques courantes d'apprentissage par renforcement comprennent les méthodes basées sur la valeur et les méthodes basées sur la politique. Les méthodes basées sur la valeur impliquent d'estimer la valeur de chaque état de l'environnement et d'utiliser ces estimations pour guider la prise de décision. Les méthodes basées sur la politique impliquent de trouver directement une stratégie optimale pour maximiser la somme cumulée des récompenses. L'apprentissage par renforcement est souvent utilisé dans des domaines tels que la robotique, les jeux, la planification et le contrôle des processus. Il est également souvent utilisé pour apprendre des comportements complexes qui seraient difficiles à programmer à la main.

On trouve dans le Tableau 1.1 une étude comparatifs des différents types d'apprentissage .

TABLE 1.1 – Les types d'apprentissage

Type d'apprentissage	**Description**	**Exemples d'algorithme**
Apprentissage supervisé	L'algorithme apprend à partir de données étiquetées, c'est-à-dire des exemples d'entrée/sortie. L'objectif est de prédire la sortie pour de nouvelles entrées.	Régression linéaire, Régression logistique, Forêts aléatoires, Réseaux de neurones
Apprentissage non supervisé	L'algorithme apprend à partir de données non étiquetées, c'est-à-dire des données sans étiquettes ou catégories prédéfinies. L'objectif est de trouver des modèles ou des structures cachées dans les données.	Clustering, Analyse en composantes principales (ACP), Réduction de la dimensionnalité
Apprentissage semi-supervisé	Une combinaison de l'apprentissage supervisé et non supervisé, où l'algorithme utilise à la fois des données étiquetées et non étiquetées.	Réseaux de neurones, Méthodes de propagation de graphes
Apprentissage par renforcement	L'algorithme apprend à partir d'une expérience itérative en interagissant avec un environnement dynamique. L'objectif est de maximiser une récompense donnée pour une action donnée.	Q-learning, Policy Gradient, Actor-Critic

Type d'apprentissage	Complexité	Temps de résolution	Domaine d'application	Nombre de techniques
Apprentissage supervisé	Élevée	Élevé	Classification, Régression, Traitement du langage naturel, Vision par ordinateur, etc.	Nombreuses techniques comme les arbres de décision, les réseaux de neurones, les SVM, etc.
Apprentissage non supervisé	Moyenne à élevée	Moyen à élevé	Clustering, Détection d'anomalies, Réduction de dimensionnalité, etc.	Techniques incluant le clustering hiérarchique, le clustering par densité, les algorithmes de factorisation matricielle, etc.
Apprentissage semi-supervisé	Moyenne	Moyen à élevé	Classification, Régression, Traitement du langage naturel, Vision par ordinateur, etc.	Techniques incluant les méthodes de propagation de l'étiquette, les modèles génératifs, les graphes de Markov, etc.
Apprentissage par renforcement	Élevée	Élevé	Jeux, Robotique, Gestion de ressources, etc.	Techniques incluant les Q-Learning, SARSA, les politiques basées sur des valeurs, etc.

TABLE 1.2 – Tableau comparatif des types d'apprentissage de l'apprentissage automatique.

1.2 Apprentissage par renforcement

1.2.1 Objectif de l'Apprentissage par renforcement

L'objectif principal de l'Apprentissage par Renforcement (RL) [24] est de former des agents autonomes capables de prendre des décisions efficaces dans des environnements incertains et potentiellement complexes. Cela peut inclure des tâches telles que la navigation dans des espaces en 3D, la résolution de problèmes, la prise de décision en matière de jeux et de négociations, etc.

Le RL utilise une approche par essais et erreurs pour former ces agents [23]. L'agent interagit avec son environnement et reçoit des récompenses ou des pénalités en fonction de ses actions. Il utilise ces rétroactions pour améliorer continuellement ses décisions.
Le RL s'appuie sur des algorithmes de traitement de l'information pour apprendre de ces interactions et déterminer les meilleures actions à prendre pour maximiser les récompenses [24] à long terme. Le RL est différent de l'apprentissage supervisé, où l'agent reçoit des instructions précises sur ce qu'il doit faire pour atteindre une tâche donnée. Au lieu de cela, dans le RL, l'agent doit découvrir par lui-même comment atteindre les récompenses en explorant son environnement et en testant différentes stratégies.
En fin de compte, le RL vise à développer des agents intelligents capables de prendre des décisions en temps réel pour atteindre leurs objectifs de manière efficace dans des environnements complexes et incertains.

1.2.2 Historique

L'apprentissage par renforcement est un domaine en constante évolution qui cherche à développer des algorithmes permettant à un agent d'apprendre à prendre des décisions dans un environnement complexe en maximisant une récompense. L'agent apprend à travers des interactions répétées avec l'environnement, en essayant différentes actions et en observant les récompenses résultantes.

L'histoire de l'apprentissage par renforcement remonte aux années 1950 [25], lorsque les psychologues ont commencé à étudier comment les animaux apprennent à partir de leurs interactions avec l'environnement.Cette approche a été formalisée par le modèle de Markov de la décision (MDP), qui décrit les étapes du processus de prise de décision dans un environnement incertain. En 1951, Richard Bellman a introduit la programmation dynamique, une méthode qui permet de résoudre des problèmes de contrôle optimaux en utilisant des méthodes itératives.
Dans les années 1980, le RL a commencé à être utilisé pour résoudre des problèmes de contrôle de processus industriels [25] , tels que la régulation de la température ou la commande des niveaux de liquides. Dans les années 1990, le RL a été appliqué avec succès à des problèmes de jeux, notamment le backgammon et le damier, où les ordinateurs ont battu les champions humains. Dans les années 2000, le RL est devenu de plus en plus populaire dans les domaines de la robotique, de l'apprentissage par renforcement profond [25], de l'optimisation de la publicité en ligne et de la prise de décision en ecomment elle avait utilisé le RL pour créer un système qui pouvait apprendre à jouer à des jeux Atari à partir de zéro. Ce système, appelé DQN (Deep Q-Network), était capable de battre les meilleurs joueurs humains dans plusieurs jeux Atari.

Depuis lors, le RL a continué à progresser, avec des algorithmes plus sophistiqués et des applications dans de nouveaux domaines tels que la conduite autonome, la robotique de service et l'optimisation de la production industrielle.

1.2.3 Principe

L'apprentissage par renforcement repose sur une boucle d'apprentissage continue[4]. L'agent interagit avec son environnement, en prenant des actions et en recevant des récompenses ou des pénalités. L'objectif de l'agent est de trouver la meilleure stratégie pour maximiser la somme cumulée des récompenses au fil du temps.
Le processus d'apprentissage se déroule en plusieurs étapes :

1-L'agent observe l'état actuel de l'environnement.

2-L'agent utilise une politique pour sélectionner une action en fonction de cet état.

3-L'environnement se met dans un nouvel état en réponse à l'action de l'agent.

4-L'agent reçoit une récompense ou une pénalité en fonction de l'état de l'environnement.

5-L'agent utilise la récompense ou la pénalité pour mettre à jour sa politique pour améliorer ses décisions futures.

La figure 1.1 explique ce principe.

La stratégie utilisée par l'agent pour sélectionner ses actions peut être basée sur la valeur ou sur la politique. Les méthodes basées sur la valeur impliquent l'estimation de la valeur de chaque état de l'environnement. La valeur d'un état est la somme cumulée des récompenses que l'agent peut s'attendre à recevoir à partir de cet état. Les méthodes basées sur la politique consistent à trouver directement une stratégie optimale pour maximiser les récompenses.

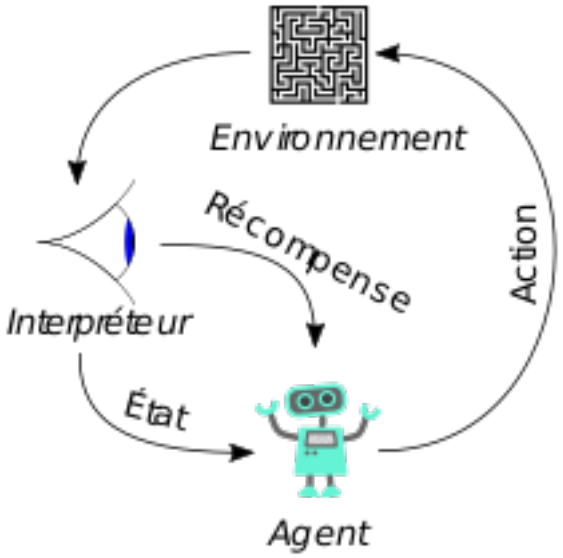

FIGURE 1.1 – Principe de l'apprentissage par renforcement[5]

1.2.4 Les avantages

Le Reinforcement Learning est une méthode efficace pour déterminer les situations qui nécessitent une action. Elle aide également à identifier les actions qui produisent les plus grandes récompenses sur le long terme. De plus, cette approche permet à l'agent IA de comprendre les actions qui conduisent à des récompenses importantes. En somme, le Reinforcement Learning permet à l'agent de prendre des décisions éclairées pour maximiser les récompenses obtenues. »L'apprentissage par renforcement présente plusieurs avantages [2] : Auto-amélioration : les agents peuvent apprendre de leurs erreurs et ajuster leurs comportements pour maximiser les récompenses. Flexibilité : le Reinforcement Learning peut être utilisé pour résoudre une grande variété de problèmes, allant des jeux en passant par les systèmes de contrôle industriels. Simulation : le Reinforcement Learning peut être utilisé pour simuler des situations complexes et évaluer les comportements futurs sans prendre de risques réels. Prise de décision : le Reinforcement Learning permet aux agents de prendre des décisions en temps réel en fonction des informations environnantes. Économie de temps et d'efforts : l'apprentissage par renforcement peut automatiser certaines tâches qui sont habituellement effectuées manuellement, ce qui peut économiser du temps et des efforts.
En général, l'apprentissage par renforcement est une technique puissante pour les agents autonomes pour apprendre à prendre des décisions dans des environnements complexes et incertains.

1.2.5 Les inconvénients

L'apprentissage par renforcement n'est pas toujours la meilleure option pour résoudre tous les problèmes. Quand il y a suffisamment de données disponibles, l'apprentissage supervisé peut être une approche plus appropriée. De plus, l'apprentissage par renforcement nécessite des ressources considérables [2] et peut prendre beaucoup de temps, ce qui peut rendre l'apprentissage supervisé plus pratique pour les cas d'utilisation plus complexes. Il est donc important de peser les avantages et les inconvénients de chaque approche pour choisir la plus appropriée pour résoudre un problème spécifique.

Voici quelques inconvénients de l'apprentissage par renforcement : Besoin de données : l'apprentissage par renforcement nécessite des données pour fonctionner, ce qui peut être un problème pour les systèmes qui n'ont pas suffisamment de données pour former un agent. Temps d'apprentissage : l'apprentissage par renforcement peut prendre beaucoup de temps pour produire des résultats optimaux, ce qui peut être un inconvénient pour les systèmes qui nécessitent des réponses rapides. Besoin de ressources : l'apprentissage par renforcement nécessite souvent des ressources importantes pour fonctionner, telles que de la puissance de calcul et de la mémoire, ce qui peut rendre la méthode inappropriée pour les systèmes qui sont limités en termes de ressources. Instabilité : les agents peuvent parfois apprendre des comportements suboptimaux ou même défavorables qui peuvent entraîner des conséquences néfastes. Difficulté à déterminer la récompense : il peut être difficile de déterminer une récompense qui représente correctement les objectifs de l'agent.

Conclusion

En conclusion, l'apprentissage automatique est une discipline en constante évolution qui offre de nombreuses possibilités pour résoudre des problèmes complexes. Nous avons vu qu'il existe plusieurs types d'apprentissage, chacun ayant ses avantages et ses inconvénients. L'apprentissage par renforcement est une méthode d'apprentissage très puissante qui permet à un agent d'interagir avec son environnement pour maximiser une récompense. Bien que cette méthode soit très efficace dans certains domaines, elle peut être difficile à mettre en œuvre et à ajuster dans d'autres

Chapitre 2

Méthodes et Algorithmes

Introduction

L'apprentissage par renforcement utilise des algorithmes et des modèles pour permettre aux agents d'apprendre en interagissant avec leur environnement. Les processus de décision markovien (MDP) sont utilisés pour modéliser ces problèmes, où les résultats d'une action dépendent de l'état actuel et des décisions futures. L'objectif est de trouver une politique qui maximise la récompense cumulative attendue.
La section suivante présente l'équation d'optimalité de Bellman,l'algorithme de Value Iteration, les fonctions de valeur et les politiques, ainsi que d'autres méthodes, telles que Q-Learning, SARSA et Gradient de la politique.

2.1 Processus de décision Markovien

L'apprentissage par renforcement repose généralement sur un cadre de processus de décision markovien (MDP), qui fournit une structure pour le problème d'apprendre à atteindre un objectif.

Dans ce processus, un agent apprend et prend des décisions en interagissant avec un environnement. Un problème peut être défini comme un processus de décision markovien lorsqu'il présente les propriétés suivantes :

— Un ensemble fini d'états S pour l'agent dans l'environnement, qui peut inclure la position de l'agent, sa vitesse et la position d'autres objets.
— Un ensemble fini d'actions A que l'agent peut prendre
— Un ensemble de récompenses scalaires R que l'agent peut recevoir, qui peuvent être données à chaque étape, telles que gagner de l'altitude pour un objet

volant ou marquer des points dans un jeu vidéo, ou données seulement à la fin d'une partie, généralement 1 pour une victoire et 0 pour une défaite.

À chaque étape de temps t=0,1,2 , l'agent reçoit une représentation de l'état de l'environnement St∈ S . Sur la base de cet état, l'agent sélectionne une action At ∈ A .. Cela nous donne la paire état-action (St, At).

Le temps est ensuite incrémenté à l'étape de temps suivante t+1 et l'environnement est transité vers un nouvel état St+1 ∈S À ce moment, l'agent reçoit une récompense numérique Rt∈ R pour l'action effectuée à partir de l'état At pris à l'état de l'environnement St. Nous pouvons considérer le processus de réception d'une récompense comme une fonction arbitraire qui mappe les paires état-action vers des récompenses. À chaque étape de temps t, nous avons :

$$f(St, At) = Rt + 1$$

La trajectoire représentant le processus séquentiel de sélection d'une action à partir d'un état, la transition vers un nouvel état et la réception d'une récompense peut être représentée comme :

$$S0, A0, R1, S1, A1, R2, S2, A2, R3...0.$$

Ce diagramme illustre parfaitement cette idée globale :

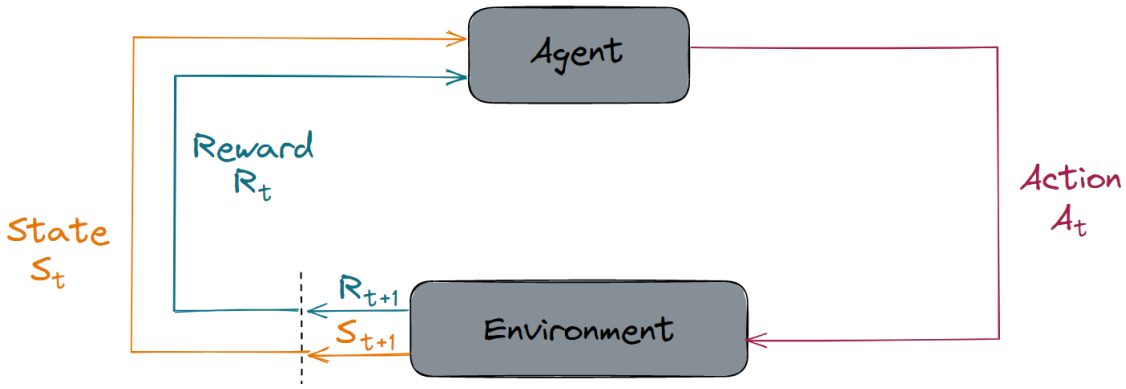

FIGURE 2.1 – Processus de décision Markovien

2.1.1 Retour attendu

L'objectif d'un agent dans un MDP est de maximiser ses récompenses cumulatives. Nous avons besoin d'une façon d'agréger et formaliser ces récompenses cumulatives. Pour cela, nous introduisons le concept de retour attendu des récompenses à un moment donné. Pour le moment, nous pouvons considérer le retour simplement comme la somme des récompenses futures. Mathématiquement, nous définissons le retour G au temps t comme :

$$Gt = Rt + 1 + Rt + 2 + Rt + 3 + ... + RT$$

où T est le temps final.

L'objectif de l'agent est de maximiser le retour attendu des récompenses. Ce concept de retour attendu est très important car c'est l'objectif de l'agent de maximiser le retour attendu. Le retour attendu est ce qui pousse l'agent à prendre les décisions qu'il prend.

2.1.2 Politiques

Une politique est une fonction qui associe à un état donné les probabilités de sélection de chaque action possible à partir de cet état. Nous utiliserons le symbole π pour représenter une politique.

Lorsque l'on parle de politiques, nous disons formellement qu'un agent "Suit une politique". Par exemple, si un agent suit la politique π au temps t, alors $\pi(a|s)$ est la probabilité que l'action A soit prise si l'agent est dans l'état S. Cela signifie qu'au temps t, sous la politique π, la probabilité de prendre l'action a dans l'état S est $\pi(a|s)$.

Notons que pour chaque état S, π est une distribution de probabilité sur les actions possibles. Par exemple, $\pi(droite)$et $\pi(gauche)$ sont des probabilités possibles pour l'état S.

2.1.3 Fonctions de valeur

Les fonctions de valeur sont des fonctions d'états ou de paires état-action qui estiment à quel point il est avantageux pour un agent d'être dans un état donné, ou à quel point il est avantageux pour l'agent d'effectuer une action donnée dans un état donné.

Cette notion de la qualité d'un état ou d'une paire état-action est donnée en termes de retour attendu. Rappelons que les récompenses qu'un agent s'attend à recevoir dépendent des actions qu'il prend dans des états donnés. Ainsi, les fonctions de valeur sont définies par rapport à des façons spécifiques d'agir. Comme la façon

dont un agent agit est influencée par la politique qu'il suit, on peut voir que les fonctions de valeur sont définies par rapport aux politiques.

2.1.4 Fonctions de valeur d'état

La fonction de valeur d'état pour la politique π, notée v_π, nous indique à quel point un état donné est bénéfique pour un agent suivant la politique π. En d'autres termes, elle nous donne la valeur d'un état selon π. Formellement, la valeur de l'état sous la politique est le retour attendu à partir de l'état au temps et en suivant la politique par la suite. Mathématiquement, nous définissons $v_\pi(s)$comme suit :

$$v_\pi(s) = E_\pi[G_t|S_t = s] = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s] \tag{2.1.1}$$

2.1.5 Fonction de valeur d'action

Fonction de valeur d'action De même, la fonction d'action-valeur pour une politique π, notée q_π, nous indique à quel point il est avantageux pour l'agent de prendre une action donnée à partir d'un état donné tout en suivant la politique q_π. En d'autres termes, cela nous donne la valeur d'une action sous π. Formellement, la valeur de l'action a dans l'état s sous la politique q_π est le retour attendu à partir de l'état s à l'instant t , en prenant l'action et en suivant la politique q_π par la suite.
Mathématiquement, nous définissons q_π (s,a) comme :

$$q_\pi(s, a) = E_\pi[G_t|S_t = s, A_t = a] = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a] \tag{2.1.2}$$

De manière conventionnelle, la fonction de valeur d'action $q_\$\pi$ est appelée la fonction Q, et la sortie de la fonction pour une paire état-action donnée est appelée une valeur Q. La lettre "Q" est utilisée pour représenter la qualité de la prise d'une action donnée dans un état donné.

2.1.6 Optimalité

L'objectif des algorithmes d'apprentissage par renforcement est de trouver une politique qui produira beaucoup de récompenses pour l'agent si l'agent suit effectivement cette politique.
Plus précisément, les algorithmes d'apprentissage par renforcement cherchent à trouver une politique qui rapportera plus de rendement à l'agent que toutes les autres politiques.

Politique optimale

En termes de rendement, une politique π est considérée meilleure ou égale à la politique $\pi^{'}$si le rendement attendu de π est supérieur ou égal au rendement attendu de $\pi^{'}$ pour tous les états. En d'autres termes,

$$\pi \geq \pi' \; ssi \; v_{\pi}(s) \geq v_{\pi'}(s) \; \forall s \in S. \tag{2.1.3}$$

Une politique π qui est meilleure ou au moins égale à toutes les autres politiques est appelée politique optimale.
On note la fonction de valeur optimale de l'état comme v_*, qui donne la plus grande récompense espérée réalisable par n'importe quelle politique π pour chaque état $s \in S$:

$$v_{\pi}(s) = \max_{\pi} v_{\pi}(s) \tag{2.1.4}$$

On note la fonction de valeur optimale de l'action q_*,qui donne la plus grande récompense espérée réalisable par n'importe quelle politique π pour chaque état $s \in S$:

$$q_{\pi}(s,a) = \max_{\pi} q_{\pi}(s,a) \tag{2.1.5}$$

2.1.7 Equation d'optimalité de Bellman

Une propriété fondamentale de la fonction de valeur optimale q_* est qu'elle doit satisfaire l'équation d'optimalité de Bellman suivante :

$$q_{\pi}(s,a) \;=\; E[R_{t+1} + \gamma \max_{a'} q_{\pi}(s',a')] \tag{2.1.6}$$

Ceci est appelé l'équation d'optimalité de Bellman. Elle indique que, pour tout couple état-action (s,a) à l'instant t, le rendement attendu à partir de l'état s, en choisissant l'action a et en suivant la politique optimale par la suite (c'est-à-dire la valeur Q de ce couple) sera le bénéfice attendu que l'on obtient en prenant l'action a dans l'état s, qui est $R_{(t+1)}$, plus le rendement attendu maximum actualisé que l'on peut atteindre à partir de tout autre couple état-action$(s^{'}, a^{'})$.
Puisque l'agent suit une politique optimale, l'état suivant $s^{'}$ sera l'état à partir duquel la meilleure action possible $a^{'}$ peut être prise à l'instant $t+1$.
Nous allons voir comment nous pouvons utiliser l'équation de Bellman pour trouver q_* . Une fois que nous avons q_*, nous pouvons déterminer la politique optimale car, avec q_* , pour tout état s, un algorithme d'apprentissage par renforcement peut trouver l'action qui maximise $q_*(s,a)$.

2.1.8 Value Iteration

L'algorithme Q-learning met à jour de manière itérative les Q-valeurs pour chaque paire état-action en utilisant l'équation de Bellman jusqu'à ce que la fonction Q converge vers la fonction Q optimale q_* . Cette approche est appelée itération de la valeur (value iteration)
On peut utiliser le Q-table pour stocker les Q-valeurs pour chaque paire état-action. L'axe horizontal de la table représente les actions, et l'axe vertical représente les états. La table Q est initialisée à des valeurs aléatoires ou à zéro.
A chaque étape, nous parcourons tous les états et pour chaque état, nous parcourons toutes les actions possibles. Pour chaque état-action, nous calculons la valeur Q en utilisant l'équation de Bellman et nous mettons à jour la table Q avec cette valeur. Cette mise à jour est effectuée de manière itérative jusqu'à ce que la table Q converge vers la table Q optimale.
La table Q peut être utilisée pour déterminer la politique optimale en sélectionnant pour chaque état l'action avec la valeur Q maximale. Cette méthode est simple et efficace pour les environnements à faible dimensionnalité, mais elle peut devenir impraticable pour les environnements avec un grand nombre d'états et d'actions.

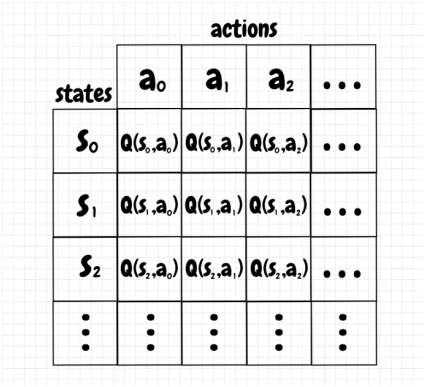

FIGURE 2.2 – Q-Valeurs en focnctions des états et des actions

2.2 Q-Learning

Q-Learning [11] est l'un des algorithmes les plus populaires pour les environnements à agent unique est l'apprentissage profond par renforcement Q, développé chez Google en 2016. Dans les premiers jours de l'apprentissage par renforcement, l'apprentissage Q a été appliqué au domaine du contrôle des processus , des processus chimiques, du contrôle automatique des processus industriels et dans le domaine du contrôle des avions. Actuellement, l'apprentissage Q est utilisé dans le domaine de la gestion de réseau, principalement pour l'optimisation de l'acheminement et du traitement des réceptions dans la communication réseau. Avec l'avènement d'AlphaGo, des recherches actives sont en cours dans le domaine de la théorie des jeux.

2.2.1 Principe

L'algorithme Q-learning est une méthode populaire pour prendre des décisions dans une situation donnée. Il utilise les valeurs Q pour déterminer la meilleure action à prendre dans un état donné, dans le but de maximiser la récompense attendue en suivant la politique optimale. Les valeurs Q sont mises à jour de manière itérative en utilisant l'équation de Bellman jusqu'à ce qu'elles convergent vers les valeurs optimales.

L'un des éléments clés du Q-learning [12] est sa méthode hors-politique, qui permet de séparer la politique d'interaction de la politique d'apprentissage. Cela signifie que même si l'action prise dans l'état suivant n'est pas optimale, l'information n'est pas immédiatement incorporée dans la mise à jour de la valeur Q de l'état actuel. Cela permet à l'algorithme d'apprendre de manière plus robuste et plus efficace. L'équation pour la valeur Q est la suivante :

$$Q(s,a) = Q(s,a)\alpha[R\gamma \max Q(s',a')Q(s,a)] \tag{2.2.1}$$

$\alpha(9)$ est le taux d'apprentissage et a une valeur comprise entre 0 et 1. R est une récompense et représente le taux de réduction de la récompense au fil du temps.

2.2.2 Algorithmes et modèles

2.2.2.1 Deep Q-Learning

La principale idée du deep Q-learning consiste à utiliser la relecture expérientielle pour combiner le Q-learning avec un réseau de neurones artificiels (CNN) [13]. Lorsque l'agent interagit avec l'environnement, il génère des échantillons (s, a,

r, s') qui peuvent être différents en fonction de l'environnement dans lequel il évolue. Cependant, si l'agent apprend uniquement à partir de ces échantillons, l'apprentissage peut être biaisé en raison de la corrélation entre eux.
Pour éviter cela, le deep Q-learning utilise une stratégie de collecte de nombreux échantillons. Lorsque le CNN apprend, les échantillons stockés en mémoire sont disposés aléatoirement et extraits le plus souvent possible [13]. La figure 2.3 illustre la différence entre Deep Q et Q)learning[26].

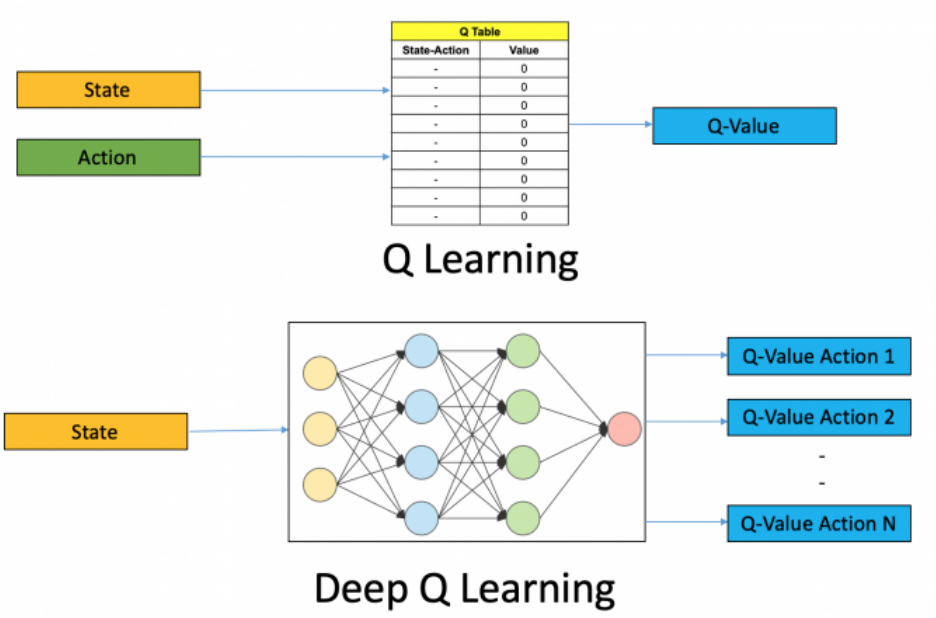

FIGURE 2.3 – Q-learning et Deep Q-learning [26]

2.2.2.2 Q-hiérarchique

L'apprentissage Q hiérarchique [14] améliore l'apprentissage Q de base en ajoutant un traitement hiérarchique au système existant. Cela divise les actions de l'agent en deux niveaux : le niveau supérieur contient le mouvement global vers le but, tandis que le niveau inférieur contient les mouvements plus fins comme monter, descendre, aller à gauche ou à droite. Cela permet une résolution plus rapide des problèmes complexes.

2.2.2.3 Double Q-Learning

Hasselt a développé le double Q-learning, qui résout le problème du Q-learning [16]. Le double Q-learning divise la fonction de valorisation du Q-learning qui

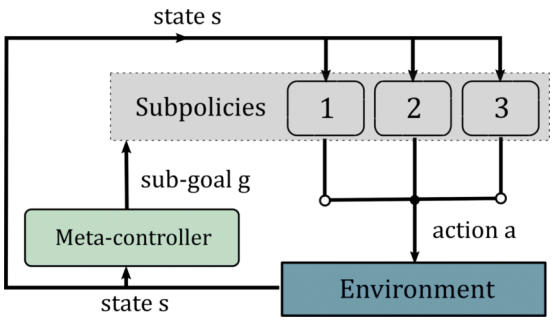

FIGURE 2.4 – Q-iérarchique [15]

détermine l'action pour éviter la déviation de la valeur dans l'algorithme du Q-learning. Le Q-learning conventionnel ne recherche aucune nouvelle valeur optimale après un certain temps, mais sélectionne à plusieurs reprises la valeur la plus élevée parmi les valeurs existantes. L'algorithme existant est identique au Q-learning. L'équation (9) est divisée en deux équations et la valeur est dérivée de manière sélective et aléatoire [15].

2.2.2.4 Autres

Il existe plusieurs algorithmes de Q-learning conçus pour les environnements à agent unique, notamment l'apprentissage incrémental à plusieurs étapes, l'apprentissage asynchrone d'approximation stochastique et l'apprentissage bayésien de Q. Ces algorithmes sont efficaces pour les tâches d'apprentissage par renforcement où le renforcement est retardé, et ils constituent une base pour développer des mécanismes d'apprentissage à plusieurs échelles de temps nécessaires aux applications réelles[17]. Une autre approche, appelée apprentissage modulaire de Q, a été introduite pour améliorer l'inefficacité de l'apprentissage de base de Q dans les systèmes multi-agents [18] . Enfin, il existe un algorithme de Q-learning basé sur un essaim qui peut être utilisé pour résoudre des problèmes complexes de manière distribuée.

2.3 Autres Methodes

2.3.1 SARSA

SARSA [19] est un algorithme sur la politique, ce qui le différencie de l'algorithme Q-Learning (algorithme hors politique). Sur la politique signifie qu'au cours de l'entraînement, nous utilisons la même politique pour que l'agent agisse (politique d'interaction) et mette à jour la fonction de valeur (politique de mise à jour).

Pendant ce temps, avec l'approche hors politique, nous utilisons des politiques différentes pour agir et mettre à jour.

Si l'on regarde l'équation utilisée par l'algorithme Q-Learning, on peut voir que la différence réside dans la manière dont il sélectionne la valeur de l'état suivant. C'est-à-dire que Q-Learning prend la valeur maximale pour l'état suivant en fonction des valeurs existantes dans la table Q. Pendant ce temps, SARSA prend la valeur de la paire état-action suivante.

En général, l'algorithme SARSA a une caractéristique de convergence plus rapide, tandis que Q-Learning donne une meilleure performance finale. Cependant, la plupart de la littérature étudie la politique de sélection d'action de l'algorithme Q-Learning ou SARSA pour équilibrer le dilemme exploitation-exploration[19] .

2.3.2 Gradient de la politique

Les applications importantes de l'apprentissage par renforcement (RL) nécessitent l'utilisation de fonction d'approximation généralisante telles que les réseaux de neurones, les arbres de décision ou les méthodes basées sur des instances. Au cours de la dernière décennie, l'approche dominante a été celle de la fonction de valeur.

La méthode du gradient de la politique est une technique d'apprentissage par renforcement qui permet d'optimiser directement la politique d'un agent en utilisant des gradients de la fonction de récompense. Contrairement aux méthodes de la valeur de l'état ou de l'action, qui cherchent à estimer la valeur optimale des états ou des actions, la méthode du gradient de la politique vise à apprendre directement la politique qui maximise la récompense attendue.

Pour ce faire, la méthode du gradient de la politique utilise une fonction de score, appelée score de la politique, qui mesure l'efficacité d'une politique en un point donné de l'espace des états [20].

2.3.3 Apprentissage par imitation

L'apprentissage par renforcement inverse (IRL) est une méthode où un algorithme observe un "expert" résoudre un problème et tente d'apprendre à résoudre le problème de manière similaire, voire supérieure. L'expert peut être une personne ou un autre algorithme qui est capable de résoudre le problème de manière efficace et peut fournir de nombreux exemples pour l'apprentissage. Cette méthode est également appelée "apprentissage par imitation", car l'algorithme apprend en imitant l'expert plutôt qu'en définissant explicitement des récompenses, ce qui est souvent difficile dans l'apprentissage par renforcement.

Un exemple courant d'utilisation de l'IRL est la navigation autonome de véhicules[21], tels que les robots utilisés dans l'industrie pour la manutention et la

logistique. Il est relativement facile pour un être humain de piloter ces robots (en tant qu'experts), mais il est difficile de définir les récompenses ou les mesures de performance appropriées, telles que la vitesse ou la précision de la navigation, qui dépendent de nombreux facteurs. L'IRL peut être utilisé pour apprendre à naviguer de manière autonome en observant l'expert et en imitant son comportement.

Conclusion

Avec l'environnement innovant actuel, les différences entre les différents algorithmes d'apprentissage par renforcement permettent aux chercheurs de découvrir les limites et les possibilités de cette méthode. Les nouvelles tendances de recherche se concentrent sur le développement d'algorithmes capables d'apprendre à partir de moins d'échantillons et de généraliser à de nouveaux environnements. Par exemple, l'apprentissage par renforcement méta-apprentissage apprend à apprendre en optimisant le processus d'apprentissage lui-même. Un autre exemple est l'apprentissage par renforcement hiérarchique, qui apprend à résoudre un problème en le décomposant en sous-problèmes plus petits. Avec ces nouvelles avancées, l'apprentissage par renforcement Q continuera à jouer un rôle important dans les systèmes intelligents de l'avenir.

Chapitre 3

Domaines d'Application

Dans ce chapitre, nous nous intéressons aux grandes entreprises impliquées dans l'apprentissage par renforcement. Nous examinerons les réalisations de Google DeepMind, OpenAI, Tesla, IBM Watson et Facebook dans ce domaine. Ensuite, nous passerons en revue les domaines d'application de cette méthode, notamment les jeux, les robots et les systèmes de dialogue interactifs.

3.1 Grands intervenants sur le marché

3.1.1 Google DeepMind

Google DeepMind [6]est une entreprise britannique spécialisée dans l'intelligence artificielle qui a acquis une renommée mondiale grâce à son algorithme AlphaGo qui a battu le champion du monde de Go, Lee Sedol, en 2016. Depuis cette victoire historique, DeepMind a continué à développer des algorithmes d'apprentissage par renforcement pour résoudre des problèmes complexes. En 2018, ils ont créé AlphaZero, un algorithme qui peut apprendre à jouer à des jeux tels que le Go, le shogi et les échecs sans aucune connaissance préalable des règles.

3.1.2 OpenAI

OpenAI [3] est une organisation de recherche en intelligence artificielle fondée en 2015 par Elon Musk, Sam Altman et d'autres personnalités de l'industrie. Ils ont développé une série d'algorithmes d'apprentissage par renforcement pour résoudre des problèmes tels que la manipulation d'objets, la reconnaissance de la parole et la simulation de l'environnement. OpenAI a également créé des agents autonomes qui peuvent jouer à des jeux vidéo comme Dota 2 et Starcraft II.

3.1.3 Tesla

Tesla [7] utilise l'apprentissage par renforcement dans ses véhicules autonomes pour améliorer la sécurité et l'efficacité de la conduite. Les véhicules Tesla collectent en temps réel des données sur l'environnement de conduite, qui sont ensuite utilisées pour entraîner des algorithmes d'apprentissage par renforcement. Ces algorithmes sont capables d'apprendre à conduire de manière plus efficace en optimisant la consommation d'énergie et en évitant les accidents.

3.1.4 IBM Watson

IBM Watson est l'un des exemples les plus connus d'application de l'apprentissage par renforcement dans la création de systèmes de dialogue interactifs avancés [27]. Cette technique est particulièrement utile pour la création de chatbots et de systèmes de dialogue automatiques. L'entreprise a récemment développé un chatbot [28] pour la chaîne de télévision américaine NBC, qui a été conçu pour répondre aux questions des téléspectateurs sur les Jeux olympiques. Grâce à l'apprentissage par renforcement, ce chatbot peut comprendre les demandes des utilisateurs et fournir des réponses précises et adaptées en temps réel, améliorant ainsi l'expérience des utilisateurs de la chaîne NBC. Ce type d'application de l'apprentissage par renforcement est de plus en plus courant dans les domaines de la technologie et de l'innovation, offrant des solutions innovantes pour améliorer les interactions avec les clients et les utilisateurs

3.1.5 Facebook

Facebook [29] utilise l'apprentissage par renforcement pour améliorer ses algorithmes de recommandation, afin de proposer un contenu personnalisé aux utilisateurs en fonction de leurs centres d'intérêt et de leurs préférences. Les données de navigation des utilisateurs sont utilisées pour former les modèles de recommandation, qui sont mis à jour régulièrement pour tenir compte des nouveaux comportements des utilisateurs.
De plus, l'algorithme peut également évaluer si les recommandations fournies ont été bien accueillies par l'utilisateur, ajustant ainsi ses recommandations futures en fonction des réactions et des préférences de l'utilisateur. En utilisant l'apprentissage par renforcement pour améliorer ses algorithmes de recommandation, Facebook peut offrir une expérience utilisateur personnalisée et de haute qualité, renforçant ainsi l'engagement et la fidélité des utilisateurs envers la plateforme.

3.2 Les domaines d'Application

3.2.1 Robotique

Dans le domaine de la robotique [2] et de l'automatisation industrielle, l'apprentissage par renforcement est employé pour permettre au robot de développer un système de contrôle optimal qui s'adapte à l'aide de son expérience et de ses actions. De cette façon, le robot peut apprendre de manière autonome.
Dans le domaine de la robotique, l'apprentissage par renforcement est utilisé pour apprendre des politiques de contrôle de robots qui leur permettent d'accomplir des tâches complexes. Les robots peuvent interagir avec leur environnement pour collecter des données et améliorer leurs performances au fil du temps.
Par exemple, il existe une méthode [8] pour apprendre des politiques de contrôle de robots à partir de données brutes, telles que des images de caméras et des données de capteurs. Les résultats montrent que cette méthode permet d'obtenir des performances supérieures à celles des méthodes traditionnelles.

3.2.2 La fouille de texte

Le Reinforcement Learning est également utilisé dans le domaine du Text mining [9], comme en témoigne le travail des chercheurs de Salesforce, une entreprise leader dans le cloud computing. Ils ont intégré le Reinforcement Learning avec un modèle de génération de texte contextuel de pointe pour créer un système capable de réaliser des synthèses de textes longs.

3.2.3 La finance

Le renforcement apprentissage (RL) est un domaine en pleine croissance en finance [9], avec de nombreux travaux récents visant à utiliser cette technique pour améliorer les décisions d'investissement. Le RL permet de prendre des décisions en considérant les conséquences à long terme de ces décisions, ce qui est particulièrement important dans un environnement incertain comme celui des marchés financiers. Voici quelques applications courantes du RL dans le domaine de la finance :

1-Portefeuille d'investissement : Les algorithmes de RL peuvent être utilisés pour optimiser la composition d'un portefeuille d'investissement en fonction de différents critères, tels que la maximisation du rendement ou la minimisation du risque.

2-Trading algorithmique : Les algorithmes de RL peuvent être utilisés pour prendre des décisions de trading plus éclairées, telles que la détermination du

moment opportun pour acheter ou vendre un actif en fonction des mouvements du marché.

3-Gestion de risque : Les algorithmes de RL peuvent être utilisés pour aider à évaluer et à gérer les risques associés aux investissements en prenant en compte les conséquences potentielles des différentes stratégies d'investissement.

3.2.4 La sante

Dans le domaine de la santé [9], l'apprentissage par renforcement est utilisé pour apprendre des politiques de décision optimales pour la gestion des traitements et la prédiction de diagnostics. Les agents peuvent interagir avec les patients pour apprendre les meilleures pratiques médicales.

les auteurs proposent une méthode [10] pour apprendre des politiques de gestion des traitements à partir de données de santé électroniques. Les résultats montrent que cette méthode permet d'obtenir des performances supérieures à celles des méthodes traditionnelles.

Voici quelques applications courantes de RL dans le domaine de la santé :

1-Administration de médicaments : Les algorithmes de RL peuvent être utilisés pour optimiser la planification de la dose et la fréquence d'administration de médicaments en fonction de critères tels que l'efficacité, la tolérabilité et les effets secondaires.

2-Planification chirurgicale : Les algorithmes de RL peuvent être utilisés pour aider à planifier les interventions chirurgicales en considérant les conséquences potentielles sur la santé du patient à long terme.

3-Diagnostic : Les algorithmes de RL peuvent être utilisés pour aider les médecins à prendre des décisions plus informées en matière de diagnostic en considérant les antécédents médicaux, les résultats de tests et les autres facteurs pertinents.

4-Soins palliatifs : Les algorithmes de RL peuvent être utilisés pour optimiser la planification des soins palliatifs en fonction des objectifs du patient et des conséquences potentielles sur leur qualité de vie.

3.2.5 La Publicité en ligne

Dans le domaine de la publicité en ligne [2], l'apprentissage par renforcement est utilisé pour apprendre des politiques de décision optimales pour diffuser des publicités en ligne aux utilisateurs. Les agents peuvent interagir avec les utilisateurs pour apprendre les stratégies gagnantes. Par exemple, dans l'article "Real-time bidding by reinforcement learning in display advertising"[3] (Zhao et al., 2013), les auteurs proposent une méthode pour apprendre des politiques de diffusion de publicités en temps réel. Les résultats montrent que cette méthode permet d'obtenir des performances supérieures à celles des méthodes traditionnelles.

3.2.6 L'automatisation industrielle

L'apprentissage par renforcement est de plus en plus utilisé dans le domaine de l'automatisation industrielle [2] pour améliorer l'efficacité des processus de production. Ce type d'apprentissage permet à un système de prendre des décisions en temps réel en se basant sur les données d'entrée qu'il reçoit, telles que les données de capteurs, les spécifications de production et les conditions environnementales.

Il trouve une multitude d'applications dans l'automatisation industrielle telles que l'optimisation des trajectoires de robots, le contrôle de processus, la surveillance de la qualité, la maintenance prédictive, et l'optimisation des paramètres de production. Par exemple, il peut être utilisé pour minimiser les coûts tout en maintenant la qualité des produits ou pour optimiser les trajectoires des robots afin de réduire le temps de cycle de production.

Néanmoins, l'utilisation de l'apprentissage par renforcement dans l'automatisation industrielle implique des défis tels que la sécurité, la fiabilité et la scalabilité. Il est crucial que les systèmes de contrôle automatisés soient fiables et sécurisés pour éviter les accidents. De plus, les modèles d'apprentissage par renforcement doivent être conçus de manière à être efficaces à grande échelle et à tenir compte des contraintes de temps réel.

En conclusion, l'apprentissage par renforcement offre des avantages significatifs dans l'automatisation industrielle pour améliorer l'efficacité des processus de production. Toutefois, il est important de comprendre les défis et les considérations pour concevoir des systèmes fiables et efficaces

3.2.7 L'ingénierie

L'application de l'apprentissage par renforcement est en pleine expansion dans le domaine de l'ingénierie [2], qui permet aux systèmes de prendre des décisions en se basant sur les données d'entrée telles que les données de capteurs, les spécifications de conception et les contraintes du système. Les avantages de cette technique dans l'ingénierie sont multiples : optimisation de la conception de systèmes complexes, planification de la maintenance, détection de pannes et réduction des coûts. Par exemple, il peut être utilisé pour optimiser la conception de structures complexes tout en réduisant le coût de la production, ou pour planifier la maintenance des équipements en utilisant des prévisions de défaillance.Par exemple, Facebook a créé une plateforme open source appelée Horizon [2]. Il permet l'optimisation de grands systèmes de production en utilisant l'apprentissage par renforcement.

Cependant, l'utilisation de l'apprentissage par renforcement dans l'ingénierie pose également des défis tels que la complexité des modèles, la qualité des données et la sécurité. Pour assurer l'efficacité et la fiabilité des modèles d'apprentissage par renforcement, il est essentiel de concevoir des modèles spécifiquement adaptés

aux caractéristiques de l'ingénierie.

3.2.8 Les jeux

Dans le domaine des jeux, l'apprentissage par renforcement est utilisé pour apprendre des politiques de décision optimales pour jouer à des jeux tels que le Go, le poker, etc. Les agents peuvent interagir avec l'environnement du jeu pour apprendre les stratégies gagnantes. Par exemple [31], les auteurs proposent une méthode pour apprendre des politiques de décision optimales pour jouer à des jeux de type Atari. Les résultats montrent que cette méthode permet d'obtenir des performances supérieures à celles des méthodes traditionnelles. L'apprentissage par renforcement a été utilisé pour développer des algorithmes capables de jouer à de nombreux jeux différents, voici quelques exemples :

-Le jeu de Go : en 2016, l'algorithme AlphaGo développé par Google DeepMind a réussi à battre le champion du monde de Go, Lee Sedol, dans un match historique[2].

-Les échecs : des algorithmes d'apprentissage par renforcement ont également été utilisés pour jouer aux échecs, avec des résultats impressionnants. Par exemple, l'algorithme AlphaZero de Google DeepMind a réussi à battre les meilleurs programmes d'échecs existants en utilisant uniquement l'apprentissage par renforcement.

-Les jeux vidéo : l'apprentissage par renforcement est également utilisé pour entraîner des programmes capables de jouer à des jeux vidéo, tels que les jeux de combat et les jeux de course.

-Le poker : l'apprentissage par renforcement a également été utilisé pour développer des algorithmes de poker capables de battre les meilleurs joueurs humains. En 2019, l'algorithme Pluribus a réussi à battre des professionnels du poker dans une série de parties en tête-à-tête.

-Les jeux de plateau : l'apprentissage par renforcement peut également être utilisé pour entraîner des programmes capables de jouer à des jeux de plateau, tels que le Scrabble et le jeu de dames.

En résumé, l'apprentissage par renforcement a été utilisé pour développer des programmes capables de jouer à de nombreux jeux différents, notamment le Go, les échecs, les jeux vidéo, le poker et les jeux de plateau. Ces programmes ont souvent réussi à battre les meilleurs joueurs humains, ce qui témoigne de leur capacité à prendre des décisions complexes en interagissant avec leur environnement.

3.2.8.1 Exemple :Frozen lake game

Le jeu Frozen Lake [30] est un exemple classique d'environnement de grille simple dans lequel un agent doit naviguer depuis un point de départ jusqu'à un emplacement cible sur une surface de lac gelé. La surface est glissante, donc l'agent peut glisser dans différentes directions lorsqu'il essaie de se déplacer.
On représente les déplacements par (N,E,S,W).La figure 3.1 présente l'espace d'états sous forme d'une matrice avec des cases numérotés de 0 à 15.

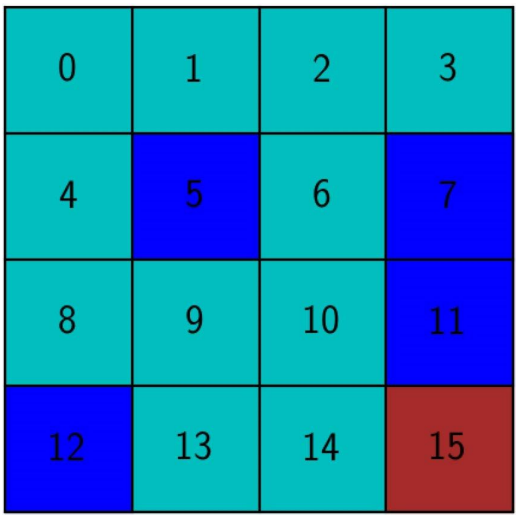

FIGURE 3.1 – Espace d'etats s $\in \{0, 1, 2, \ldots, 15\}$[30]

Le modèle d'apprentissage par renforcement recevrait une récompense positive lorsqu'il atteint le but, une pénalité négative lorsqu'il tombe dans l'eau et une récompense neutre de 0 dans tous les autres cas. Les valeurs des récompenses et des pénalités sont limitées à un ensemble (-1, 0, 1). La figure 3.2 Illustre l'espace de récompenses pour chaque état possible de l'agent.

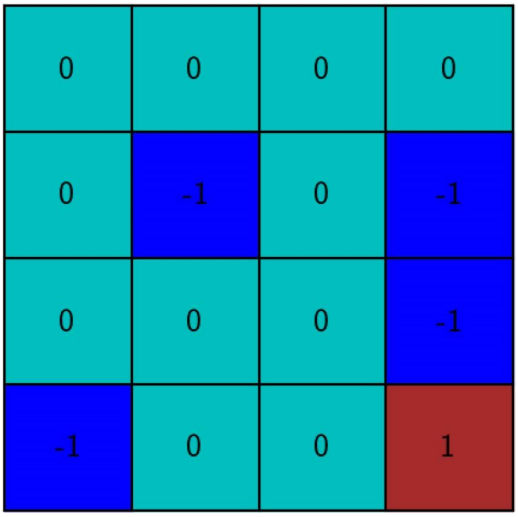

FIGURE 3.2 – Rewards r $\in \{-1, 0, 1\}$[30]

Il existe différents types d'algorithmes d'apprentissage par renforcement disponibles, tels que Q-learning, SARSA et Deep Reinforcement Learning. Le choix de l'algorithme dépendrait de la complexité du jeu et des ressources disponibles.

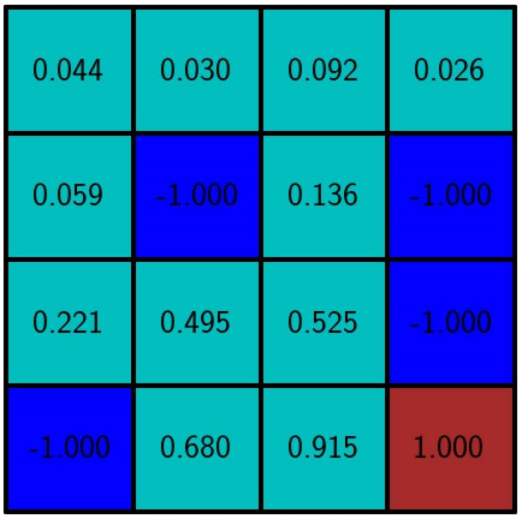

FIGURE 3.3 – Valeurs d'états après résolution State values v(s) = maxa q(s ; a) [30]

Pour choisir la politique optimale, le modèle sélectionne l'action qui a la plus grande valeur de Q (étant donné un état donné) et effectue cette action. Les valeurs Q pour chaque état-action sont stockées dans un tableau Q. Lorsque le modèle interagit avec l'environnement, il utilise le tableau Q pour sélectionner la prochaine

action à prendre. Pendant l'entraînement, le modèle met à jour la valeur de Q pour chaque état-action visité. L'objectif est de trouver la valeur Q optimale pour chaque état-action, qui représente la récompense cumulative attendu . Le tableau Q de la figure 3.3 est un exemple de valeurs d'états (Q-values) Après un nombre d'itérations.Il présente les récompenses attendues pour chaque action a prise à un état s.

s	S	E	N	W
0	0.044	0.027	0.034	0.036
1	-0.790	-0.030	0.030	-0.070
2	0.092	0.035	0.063	0.037
3	-0.791	-0.081	0.026	-0.039
4	0.046	-0.779	-0.067	0.059
5	0	0	0	0
6	0.136	-0.751	-0.141	-0.751
7	0	0	0	0
8	-0.743	0.221	0.095	0.046
9	0.495	0.291	-0.740	0.096
10	0.525	-0.716	0.027	0.401
11	0	0	0	0
12	0	0	0	0
13	0.408	0.680	0.290	-0.706
14	0.740	0.915	0.491	0.550
15	0	0	0	0

FIGURE 3.4 – Q-table [30]

Conclusion

En conclusion, les entreprises comme Google DeepMind, OpenAI, Tesla, IBM Watson et Facebook ont fait des progrès significatifs dans le domaine de l'apprentissage par renforcement. Leurs réalisations ont ouvert de nouvelles perspectives pour l'utilisation de cette méthode dans différents domaines tels que les jeux, les robots, les systèmes de dialogue interactifs et la conduite autonome. Les résultats obtenus montrent que l'apprentissage par renforcement a le potentiel de résoudre des problèmes complexes et d'améliorer considérablement la qualité de vie des gens. L'avenir de l'apprentissage par renforcement est prometteur, et il est susceptible de jouer un rôle crucial dans la résolution de problèmes difficiles dans divers domaines.

Conclusion Générale

En conclusion, nous avons exploré l'apprentissage par renforcement, une méthode d'apprentissage automatique qui peut être utilisée pour résoudre des problèmes complexes dans divers domaines. Nous avons discuté des concepts clés tels que les états, les actions, les récompenses, les politiques et les fonctions de valeur, ainsi que des algorithmes populaires tels que Q-learning, SARSA et DQN.

Nous avons également examiné les applications de l'apprentissage par renforcement dans différents domaines, notamment la robotique, les jeux, la finance, la publicité en ligne et la santé. Nous avons constaté que l'apprentissage par renforcement est une approche de machine learning puissante et prometteuse qui a le potentiel d'améliorer les performances de l'agent dans divers environnements.

Cependant, il y a encore des défis à surmonter, tels que l'exploration et l'exploitation, le problème de la malédiction de la dimensionnalité et l'interprétabilité des modèles de l'apprentissage par renforcement. Ces défis nécessitent une recherche continue pour améliorer et développer de nouvelles approches.

Enfin, nous sommes convaincus que l'apprentissage par renforcement continuera de jouer un rôle important dans le développement de l'IA et du machine learning. Les applications de l'apprentissage par renforcement dans des domaines tels que la robotique et la santé ont déjà montré leur impact positif sur la vie des gens, et nous nous attendons à ce que l'apprentissage par renforcement continue à offrir de nouvelles solutions à des problèmes complexes dans divers domaines à l'avenir.

Références

[1] machine learning. https://datascientest.com/machine-learning-tout-savoir, 15 avril 2023.

[2] reinforcement learning. https://www.lebigdata.fr/reinforcement-learning-definition, avril 2023.

[3] Openai. https://openai.com/, avril 2023.

[4] issam el alaoui. Apprentissage par renforcement – de la théorie à la pratique. https://blog.octo.com/apprentissage-par-renforcement-de-la-theorie-a-la-pratique/, avril 2023.

[5] Learning by doing. https://france.devoteam.com/paroles-dexperts/learning-by-doing-apprentissage-par-renforcement/, avril 2023.

[6] Google deep mind. https://www.deepmind.com, avril 2023.

[7] Tesla. https://www.tesla.com/autopilot, avril 2023.

[8] Robots. https://arxiv.org/abs/1504.00702, avril 2023.

[9] Reinforcement learning : Définition et application. https://datascientest.com/reinforcement-learning, avril 2023.

[10] Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. https://arxiv.org/pdf/1807.01473.pdf, avril 2023.

[11] Q-learning algorithms : A comprehensive classification and applications, jong wook kim,department of computer science, sangmyung university, seoul, south korea. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8836506, avril 2023.

[12] Q-learning. http://www.gatsby.ucl.ac.uk/~dayan/papers/cjch.pdf, avril 2023.

[13] Energy-efficient scheduling for real-time systems based on deep q-learning model. https://ieeexplore.ieee.org/document/8016380, avril 2023.

[14] A neural model of hierarchical reinforcement learningl. http://compneuro.uwaterloo.ca/publications/Rasmussen2017.html, avril 2023.

[15] Hierarchical learning from human preferences and curiosity. https://link.springer.com/article/10.1007/s10489-021-02726-3, avril 2023.

[16] Q-learning algorithms : A comprehensive classification and applications. https://www.researchgate.net/publication/335805245_Q-Learning_Algorithms_A_Comprehensive_Classification_and_Applications, avril 2023.

[17] Stochastic optimal relaxed automatic generation control in non-markov environment based on multi-step q() learning. https://ieeexplore.ieee.org/document/5706397, avril 2023.

[18] A modular approach to multi-agent reinforcement learning. https://link.springer.com/chapter/10.1007/3-540-62934-3_39, avril 2023.

[19] Backward q-learning : The combination of sarsa algorithm and q-learning. https://dl.acm.org/doi/10.1016/j.engappai.2013.06.016, avril 2023.

[20] Policy gradient methods for reinforcement learning with function approximation. https://homes.cs.washington.edu/~todorov/courses/amath579/reading/PolicyGradient.pdf, avril 2023.

[21] Learning driving styles for autonomous vehicles from demonstration. https://ieeexplore.ieee.org/document/7139555, avril 2023.

[22] machine learning. https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML, avril 2023.

[23] Reinforcement learning with neural network. https://www.baeldung.com/cs/reinforcement-learning-neural-network#:~:text=The%20objective%20of%20reinforcement%20learning,known%20as%20an%20optimal%20policy., avril 2023.

[24] concept objective in category reinforcement learning. https://livebook.manning.com/concept/reinforcement-learning/objective, avril 2023.

[25] Sebastian Thrun and Michael L Littman. Reinforcement learning : an introduction. *AI Magazine*, 21(1) :103–103, 2000.

[26] A hands-on introduction to deep q-learning using openai gym in python. https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/, avril 2023.

[27] watson. https://cbmm.mit.edu/sites/default/files/documents/watson.pdf, avril 2023.

[28] A deep reinforcement learning chatbot. https://arxiv.org/abs/1709.02349, avril 2023.

[29] Jason Gauci, Edoardo Conti, Yitao Liang, Kittipat Virochsiri, Yuchen He, Zachary Kaden, Vivek Narayanan, Xiaohui Ye, Zhengxing Chen, and Scott

Fujimoto. Horizon : Facebook's open source applied reinforcement learning platform. *arXiv preprint arXiv :1811.00260*, 2018.

[30] Deep reinforcement learning. `http://www.cs.otago.ac.nz/cosc470/09-deep-reinforcement-learning.pdf`, avril 2023.

[31] Proximal policy optimization algorithms. `https://arxiv.org/abs/1707.06347`, avril 2023.