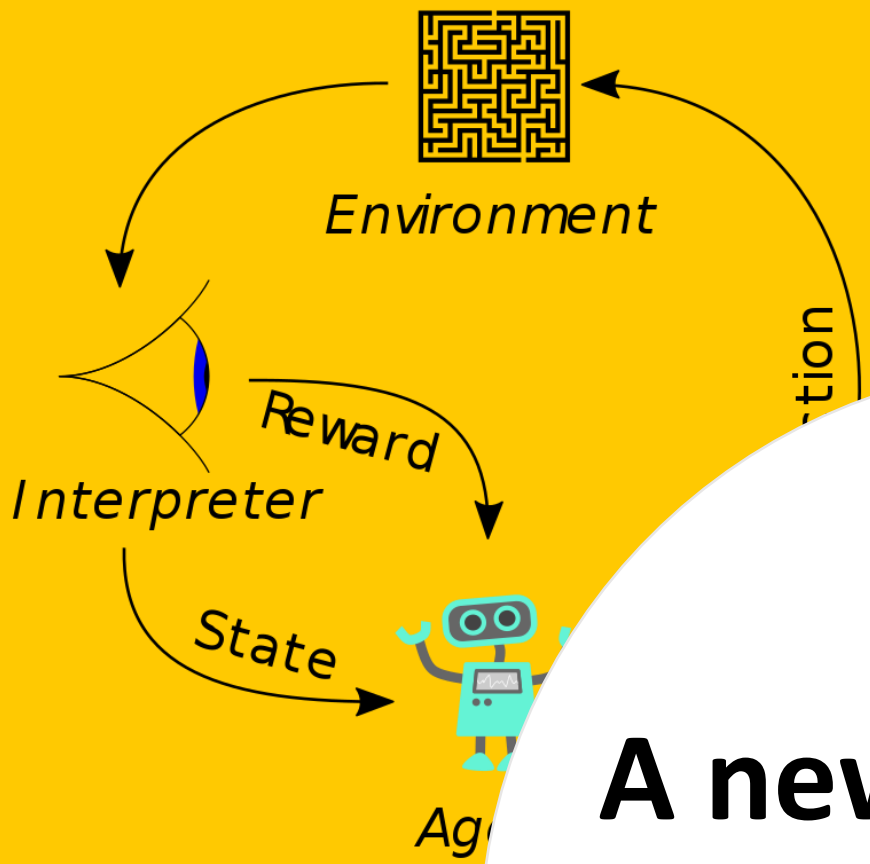
BeeBryte

Energy Intelligence & Automation

7 min



A new way to go!
Le reinforcement learning expliqué

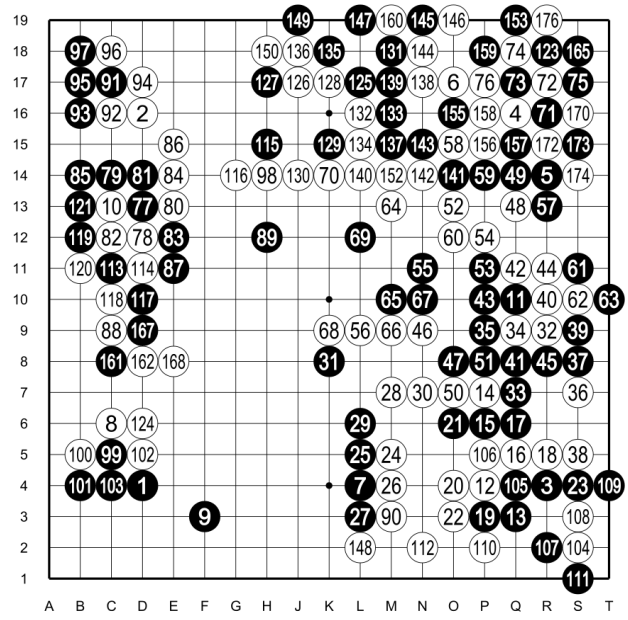
Simon CLOÎTRE, Yu Jia CHEONG, Patrick LEGUILLETTE

24 Juillet 2019

Introduction au Reinforcement Learning

En 1997, un an après un premier duel remporté par l'humain, Deep Blue, le superordinateur développé par IBM, bat le champion du monde d'échec Garry Kasparov. C'est la première fois qu'une intelligence artificielle remporte un tel défi et marque une étape importante dans le développement de ce domaine.

Les chercheurs se sont depuis attaqués à des problèmes de complexité grandissante, et tout particulièrement à un autre jeu de plateau réputé pour ses combinaisons quasi infinies : le jeu de go. A titre de comparaison, quand le jeu d'échec présente 10^{40} configurations légales, le jeu de go en dénombre 10^{170}.



Lee Sedol (B) vs AlphaGo (W) - Game 3

(122) at 113 151 at 73 (154) at (72) 163 at 145 (164) at 73 (166) at (160)
169 at 145 171 at (160) 175 at 71

En mars 2016, les ingénieurs de Google DeepMind réalisent un exploit, comparable à leurs aînés d'IBM 20 ans plus tôt, en battant Lee Sedol, l'un des meilleurs joueurs au monde, avec leur programme AlphaGo. Un an plus tard, en octobre 2017, sort AlphaGo Zero, sorte d'évolution du précédent. La principale spécificité de cette dernière intelligence artificielle est que la seule chose que lui ont apprise ses créateurs sont les règles du jeu de go. Après seulement trois jours à se former de lui-même et comprendre quelles stratégies sont les plus favorables, AlphaGo Zero affronte son aîné AlphaGo. Le résultat est sans appel : 100 victoires à 0 pour le nouveau venu. Un tel résultat illustre le potentiel d'une branche du machine learning en plein essor : l'apprentissage par renforcement, ou Reinforcement Learning.

Au travers de cet article, nous vous présenterons les différentes composantes de base d'un algorithme de Reinforcement Learning ainsi que les principales problématiques qu'il soulève au travers d'un exemple d'intégration dans les solutions d'optimisation énergétique de BeeBryte.

LE REINFORCEMENT LEARNING DÉCORTIQUÉ

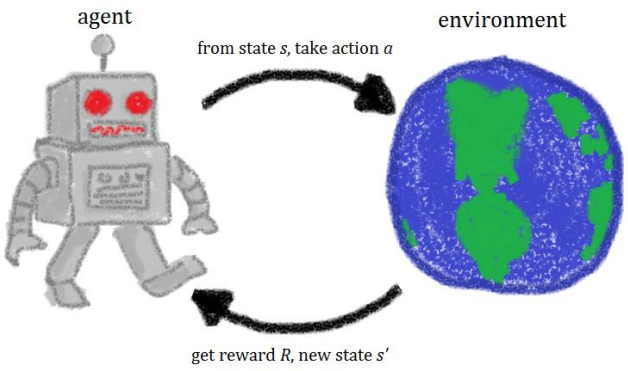
Pour illustrer le fonctionnement du Reinforcement Learning, prenons un cas d'usage que nous adressons chez BeeBryte : quand charger et décharger une batterie par rapport à un signal prix afin d'optimiser les transactions énergétiques résultantes.

Deux éléments constituent la base de tout algorithme de Reinforcement Learning. Tout d'abord, l'agent est l'entité intelligente qui va chercher à déterminer les meilleurs moments pour charger et décharger la batterie. Celui-ci évolue dans un environnement (constitué ici de la batterie et du marché de l'électricité) qui constitue la deuxième composante du problème. L'agent va prendre à chaque étape une nouvelle décision appelée action

(ici le choix de charger, décharger ou ne rien faire), selon les observations qu'il va faire de son environnement. Est appelé "état" (state) l'ensemble des informations que l'agent va percevoir de l'environnement et qui vont lui servir à choisir son action (ici le niveau de charge de la batterie et le prix actuel et à venir de l'électricité).

L'agent va partir à la découverte de son environnement en essayant d'abord aléatoirement diverses actions, et en mémorisant la relation entre état, action et conséquences, pour optimiser ses actions futures même face à des situations qu'il n'a jamais rencontrées.

Concrètement, son apprentissage passe par un système de récompense / pénalité lui permettant de quantifier la pertinence de ses actions, et adapter son comportement vers une performance maximale. De façon simplifiée, dans notre cas d'usage il va par exemple recevoir une récompense (reward) croissante avec le prix de l'électricité s'il décharge la batterie, ou à l'inverse une pénalité s'il choisit de la charger.



Mathématiquement, chaque couple (état, action) se voit attribuer un score, fonction des récompenses ou pénalités qui en découlent. Dans les faits, lorsque le problème se complexifie, une simple matrice (état, action) capitalisant le score de chaque couple ne suffit plus, et l'utilisation de réseaux de neurones permet, outre une gestion simplifiée des grandes dimensionnalités, de généraliser le comportement de l'agent à des situations inédites par rapport à son apprentissage.

Le Reinforcement Learning a pour avantage d'être moins coûteux à déployer qu'un calcul d'optimisation, dans le sens où il peut se contenter de connaissances a priori très limitées sur le système qu'il adresse, mais il aboutira généralement à un comportement sub-optimal seulement et peut se révéler long à entraîner.

LONG TERME VS COURT TERME

Un des problèmes pouvant limiter un apprentissage pas-à-pas est qu'une solution globalement optimale n'est pas forcément celle qui donne la meilleure récompense immédiate et il faut donc dans la phase d'apprentissage explorer des "mauvaises" décisions à court terme pour, dans certains cas, obtenir une grosse récompense à long terme, compensant les pénalités accumulées auparavant.

En effet, dans notre exemple, charger la batterie n'est jamais la meilleure option à court terme car elle va nous coûter plus cher que de la décharger ou de ne rien faire. Cependant, charger quand le prix est bas pour décharger quand le prix est haut est plus rentable que de ne rien faire.

Afin de traiter convenablement de telles contradictions, on introduit dans le calcul de notre score de chaque couple (état, action) non seulement la récompense qu'on devrait recevoir à l'étape suivante, mais également toutes les potentielles retombées futures, qu'elles soient bonnes ou mauvaises. L'un des principaux enjeux sera donc de bien définir quelle récompense attribuer à quel horizon temporel, et la fonction score qui en découle afin d'orienter l'agent vers la meilleure solution globale.

EXPLORATION VS EXPLOITATION

Comme expliqué précédemment, la meilleure solution n'est pas toujours celle qui apporte la meilleure récompense immédiatement, ce qui incite à explorer d'autres stratégies. Cependant, il est bien souvent difficile, voire impossible, d'essayer toutes les possibilités, et il faut donc chercher à ne pas perdre trop de temps avec celles qui n'ont que peu de

chances d'aboutir. C'est ce que l'on appelle le dilemme exploration/exploitation.

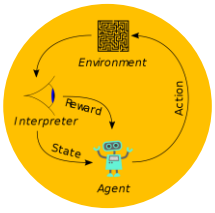
L'exploitation désigne le fait de ne se servir que des résultats qu'on a déjà pu obtenir. Ainsi, l'agent choisira toujours l'action qui a le score le plus élevé compte tenu de son niveau de connaissance de son environnement. A l'inverse, l'exploration lui permet d'approfondir sa connaissance de son environnement, mais au prix d'actions peu bénéfiques. Il est donc nécessaire de trouver le bon compromis entre ces deux dynamiques opposées, car une stratégie d'exploitation seule nous ferait stagner dans un comportement sub-optimal alors qu'une stratégie d'exploration seule se disperserait et serait globalement non optimale.

Une des missions du data scientist sera donc de déterminer le meilleur protocole de choix de l'action (appelé police) à son agent en fonction des performances (vitesse d'apprentissage vs bénéfice) visées.

CONCLUSION

Nous avons pu donner, au travers de cet article, un rapide aperçu de cette branche en plein développement du machine learning qu'est le Reinforcement Learning. Si nous avons mentionné certains écueils à éviter lorsqu'on veut utiliser cette approche, il ne faut pas oublier qu'il n'y a pas de solution universelle en IA et que chaque problème nécessite une application "sur mesure" de telle ou telle technologie.

Inspirée de l'apprentissage humain, cette discipline paraît simple à comprendre et redoutable, mais est encore en plein développement et souvent délicate à mettre en œuvre. Son potentiel n'est plus à démontrer (cf. AlphaGo Zero), et prometteuse de performances toujours accrues de nos solutions.



QUELQUES MOTS SUR BEEBRYTE

Fondée en 2015 par deux entrepreneurs chevronnés, Frédéric Crampé et Patrick Leguillette, BeeBryte est installée à Lyon et à Singapour. Notre équipe est aujourd'hui constituée de 25 professionnels.

Notre investisseur stratégique est la Compagnie Nationale du Rhône (CNR), le plus grand producteur d'énergies renouvelables en France. La société a aussi reçu le soutien de Bpifrance et de l'ADEME ainsi que des programmes d'accélération de TechFounders, INTEL, Greentech Verte et Novacité de la CCI Auvergne-Rhône-Alpes.

BeeBryte tire parti du formidable potentiel de l'intelligence artificielle pour rendre les bâtiments commerciaux et industriels plus économes en énergie et plus intelligents, afin de réduire leur facture électrique et leur empreinte carbone.

Notre logiciel-service contrôle automatiquement les ressources flexibles côté demande (e.g. équipements de chauffage-climatisation et systèmes de stockage / batteries) afin de modifier avantageusement le profil de consommation de nos clients et leur permettre d'économiser jusqu'à 40% sur leurs coûts énergétiques.

Notre technologie combine une méthodologie d'optimisation en temps réel brevetée en 2016, des modèles auto-apprenants et des analyses prédictives pour offrir des services énergétiques dynamiques totalement intégrés à l'Internet des Objets.

contact@beebryte.com

www.twitter.com/BeeBryteGroup
www.linkedin.com/company/beebryte
www.beebryte.com