

Chap 1 - Introduction à l'Apprentissage par renforcement

Jordy Palafox

Intelligence artificielle et Applications

ING 2

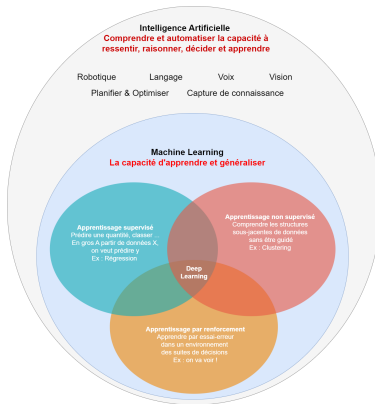
2023-2024



Introduction

Objectif du cours

Comprendre les rudiments de l'**apprentissage par renforcement** (AR).
Mais c'est quoi l'apprentissage par renforcement ?



Objectif de l'AR

Apprendre à un agent (un algorithme) à faire, à réaliser des tâches par des interactions avec son environnement grâce à des rétro-actions.

Exemple

On veut construire un algorithme qui permet la conduite autonome d'une voiture.

Une actu de cet été

Exemple

On a un dé et le but du jeu est de le lancer tant que la somme des faces obtenues ne dépasse pas un seuil, 21 par exemple. Avant chaque lancer, on regarde notre score, c'est le **score final** si on s'arrête. Si on choisit de lancer et que l'on dépasse le seuil, on prend une **pénalité** (-10 par exemple).

Ici plusieurs stratégies sont possibles :

- Lancer le dé un nombre fixe de fois,
- Lancer le dé tant que l'on ne dépasse pas 15,
- Lancer le dé tant que l'on ne dépasse pas 15 et choisir de lancer ou non à pile ou face
- etc ...

Exemple introductif

On voit apparaître un certain nombre d'éléments permettant de formaliser plus ou moins rigoureusement le problème :

- On a un ensemble d'actions possibles $\mathcal{A} = \{\text{lancer}, \text{arrêter}\}$,
- A chaque instant t de la partie, on prend la décision sur le score actuel. Le score est l'état du jeu en cours, or les états possibles sont des entiers entre 0 et ... l'infini. On notera \mathcal{S} l'ensemble des états, qui est $\mathcal{S} = \{0, 1, 2, \dots, 21, \text{plus que } 21\}$,
- En fonction du score donc de l'état actuel s_t , par exemple 13, on peut savoir l'état suivant (de façon probabiliste) :
 - on s'arrête, le score est bloqué
 - on continue et on a 1 chance sur 6 d'obtenir 14,15,16,17,18,19

On a donc une fonction $(s_{t+1}|s_t, a_t)$ où a_t est l'action réalisée une fois en l'état s_t .

Exemple introductif

En résumé, on a :

- un ensemble d'instants $t \in \mathcal{T}$,
- un ensemble d'états du jeu $s_t \in \mathcal{S}$,
- des actions possibles $a_t \in \mathcal{A}$,
- une fonction \mathcal{P} permettant de savoir comment évolue l'état du système pour chaque couple d'action / état,
- une fonction de score, d'évaluation,
- un objectif (à maximiser)

Le couple $(\mathcal{S}, \mathcal{P})$ définit un système dynamique, c'est-à-dire un système qui évolue au cours du temps avec la particularité d'évoluer en fonction des actions de \mathcal{A} .

Exemple introductif

On notera que :

- Certains états sont particuliers : 0 est l'état initial, "plus de 21" est un état terminal (le jeu s'arrête),
- la fonction \mathcal{P} est la loi d'évolution du système,
- l'état du système ne dépend que de l'état présent, on parle de système **markovien**,
- la dynamique est **probabiliste** puisqu'elle repose sur un lancer de dés (par opposition aux systèmes **déterministes**, on connaît explicitement la conséquence d'une action),
- on connaît toutes les informations pour prendre sa décision (**information complète**), alors l'inverse un algorithme qui apprend à jouer au poker doit décider sans connaître le jeu de son adversaire (**information cachée**).

Tout ceci définit un **problème de décision markovien** qui permet de généraliser le problème d'introduction.

Processus de décision markovien (ou de Markov)

Définition

On appelle **processus de décision de Markov** un quadruplet $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$ où :

- \mathcal{S} ensemble des états d'un processus (fini),
- \mathcal{A} ensemble des actions possibles,
- \mathcal{P} fonction de transition définie par :

$$\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \longrightarrow [0, 1]$$

$$(s, a, s') \mapsto \mathcal{P}(s, a, s') = \mathbb{P}[s_{t+1} = s' | s_t = s, a_t = a]$$

- \mathcal{R} fonction de retour définie par :

$$\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \longrightarrow \mathbb{R}$$

$$(s, a, s') \mapsto \mathcal{R}(s, a, s') = \mathbb{E}[r_t | s_{t+1} = s', s_t = s, a_t = a]$$

c'est donc le retour moyen ou espéré quand on exécute a en s et qu'on arrive en s' et où r_t est la récompense perçue.

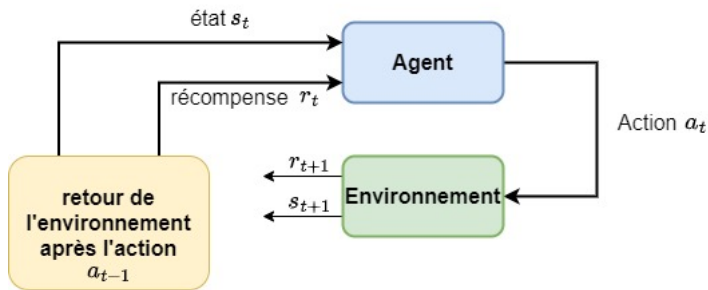
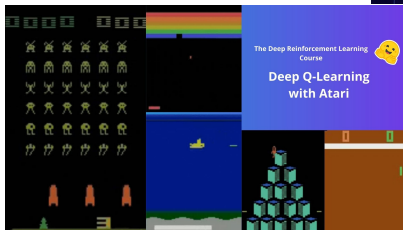
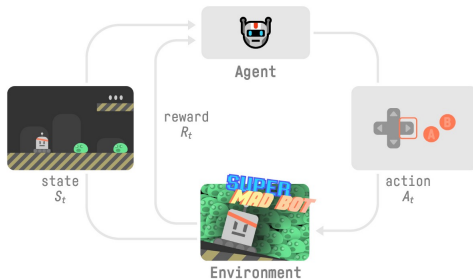
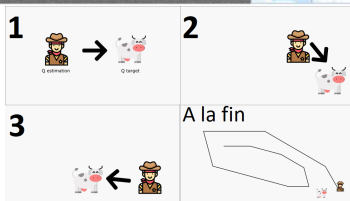


Figure: AR en bref

Quelques exemples



Quelques exemples



Ici on parle de Processus de Décision Markovien qui repose sur les notions de probabilité suivante :

Définition

Un **processus stochastique** est la donnée :

- d'un espace probabilisé $(\Omega, \mathcal{F}, \mathbb{P})$,
- d'un espace mesurable (E, \mathcal{G}) ,
- d'une famille de variables aléatoires $(Y_t)_{t \in T}$ définies sur $(\Omega, \mathcal{F}, \mathbb{P})$ à valeurs dans (E, \mathcal{G}) .

où Ω , E sont des ensembles, \mathcal{F} , \mathcal{G} sont des tribus et \mathbb{P} une probabilité.

Définition

Une **processus stochastique** est une suite de variables aléatoires $(Y_t)_{t \in T}$ où t est le temps.

Définition

Soit X_t , $t \in \mathbb{N}$ un processus stochastique à valeurs dans un ensemble **fini ou dénombrable** $\mathcal{S} = \{s_1, \dots, s_t, \dots\}$ vérifiant la **propriété de Markov** i.e. si étant donné l'état présent, la prédiction de l'état futur ne dépend que de l'état actuel et pas des états précédents. Alors (X_t) est appelée **Chaîne de Markov** et on a :

$$\mathbb{P}(X_{t+1} | X_t, X_{t-1}, \dots, X_1, X_0) = \mathbb{P}(X_{t+1} | X_t)$$

Complément sur les processus stochastiques et markovien

En continu, c'est un peu plus difficile. On le laisse de côté !

Une chaîne de Markov est entièrement définie par sa **matrice de transition** P dont les coefficients sont donnés par :

$$P_{ij} = \mathbb{P}(X_{n+1} = s_j | X_n = s_i).$$

C'est une **matrice stochastique** : la somme des éléments de chaque ligne est égale à 1.

Il suffit alors de connaître la distribution de X_0 pour calculer toutes les probabilités :

$$\begin{aligned} \mathbb{P}(X_n = x_n, \dots, X_1 = x_1, X_0 = x_0) &= \\ &= \mathbb{P}(X_n = x_n | X_{n-1} = x_{n-1}, \dots, X_0 = x_0) \mathbb{P}(X_{n-1} = x_{n-1}, \dots, X_0 = x_0) \\ &= \mathbb{P}(X_n = x_n | X_{n-1} = x_{n-1}) \mathbb{P}(X_{n-1} = x_{n-1}, \dots, X_0 = x_0) \\ &= P_{x_n x_{n-1}} \mathbb{P}(X_{n-1} = x_{n-1}, \dots, X_0 = x_0) \\ &= \dots \text{ (récurrence)} \\ &= P_{x_n x_{n-1}} \dots P_{x_1 x_0} \mathbb{P}(X_0 = x_0). \end{aligned}$$


Observations et Espace d'états

Distinguer un état d'une observation


- Un **état** s est une description complète de l'état du monde. Par exemple aux échecs, on a accès à l'ensemble des informations.
- Une **observation** o est une description partielle d'un état.

Observation Space

State: complete description of the state of the world (no hidden information).



Observation: partial description of the state of the world.



Espace d'actions

Les actions peuvent être soit **discrètes** soit **continues**.

- discrète signifie un nombre fini d'actions possibles,
- continue signifie qu'on a un nombre infini d'actions possibles.


Action Space

Discrete: finite number of possible actions



A screenshot from the video game Super Mario Bros. The game screen shows Mario on a brick floor, with various obstacles and power-ups. The text 'WORLD 1-1' and 'TIME 3:33' is visible at the top. This represents a discrete action space where the number of possible actions is finite.

Continuous: infinite number of possible actions

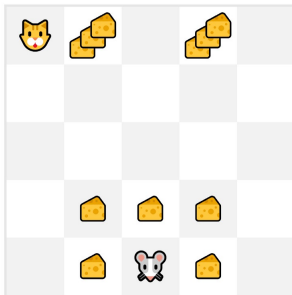
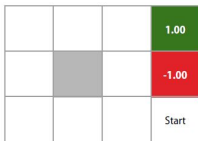


A screenshot from a self-driving car interface, showing the steering wheel and dashboard. The text 'Deep RL Course' is visible at the bottom. This represents a continuous action space where the number of possible actions is infinite.

Les récompenses

Les récompenses peuvent être positives ou négatives. L'idée principale est de voir comment l'agent les prend en compte pour améliorer ses performances dans le futur.

Grid world



Définition

Un **problème de décision de Markov** (PDM) est un problème modélisé par un processus de décision de Markov **et** où l'on cherche à optimiser une certaine **fonction objectif**.

PDM et modèle d'optimalité

Quelques formes classiques de fonctions objectif :

- Somme des retours le long d'une trajectoire (**modèle à horizon fini**):

$$\sum_{t \geq 0} r_t$$

- Moyenne des retours le long d'une trajectoire (**modèle à récompense moyenne**):

$$\frac{1}{T} \sum_{t=0}^{T-1} r_t$$

- Somme pondérée des retours avec facteur de dépréciation (**modèle à horizon infini** pondéré avec facteur d'actualisation) :

$$\sum_{t=0}^{+\infty} \gamma^t r_t$$

avec $\gamma \in [0, 1[$. C'est le modèle le plus fréquent et que l'on utilisera dans la suite.

Attention

On rencontrera deux cas :

- \mathcal{P} et \mathcal{R} sont bien connus, on parle de **planification**,
- \mathcal{P} et \mathcal{R} sont inconnus, c'est vraiment ce que l'on appelle problème d'apprentissage par renforcement.

On vient d'expliquer que l'on allait chercher à optimiser une fonction objectif mais on ne va évidemment pas faire n'importe comment ... (on peut toujours cela dit et espérer que cela se passe bien).

On va donc mettre en place **une stratégie** pour optimiser notre fonction, une **politique** (policy).

Définition

Une politique π est une façon de choisir pour un agent son action.

Optimisation et politique

Les formes classiques de politique :

- **déterministe** : à un état, une action

$$\begin{aligned}\pi : \mathcal{S} &\longrightarrow \mathcal{A} \\ s &\mapsto a\end{aligned}$$

- **stochastique** : à un état, une distribution de probabilités sur les états

$$\begin{aligned}\pi : \mathcal{S} \times \mathcal{A} &\longrightarrow [0, 1] \\ (s, a) &\mapsto \pi(s, a) = \mathbb{P}[a_t = a | s_t = s]\end{aligned}$$

- **non stationnaire** : on dépend de l'instant de la décision

$$\begin{aligned}\pi : \mathcal{S} \times \mathcal{A} \times \mathcal{T} &\longrightarrow [0, 1] \\ (s, a, t) &\mapsto \pi(s, a) = \mathbb{P}[a_t = a | s_t = s, t]\end{aligned}$$

Pour le modèle à horizon infini pondéré, on peut montrer qu'il existe une politique optimale (dans un sens que l'on va définir) qui est déterministe **et** stationnaire.

Problème : comment comparer des politiques ?

Valeur d'un état

Idée : la valeur d'un état indique s'il est intéressant pour l'objectif d'être dans cet état là. Une "bonne" politique va favoriser des états à forte valeur si on maximise et faible si on minimise.

Pour notre modèle

$$R(t) = \sum_{k \geq 0} \gamma^k r_{t+k}$$

on a l'observation suivante :

étant donné un état, un état dont la valeur est meilleure est dans son voisinage immédiat (sauf si état optimal déjà atteint !)

Problème : la présence d'optima locaux pour la fonction valeur, on va donc supposer qu'il n'y a qu'un seul optimum global et pas d'optimum locaux (c'est un problème fréquent en deep learning !)



Définition

La **valeur** de l'état s pour la politique π , notée $V^\pi(s)$, est définie par :

$$V^\pi(s) = \mathbb{E}_\pi[R(s)|s_0 = s]$$

C'est donc l'espérance de R quand l'environnement est initialement en l'état s_0 .

Définition

La **fonction valeur** est une application qui associe à tout état sa valeur, une politique π étant donnée.

Définition

On définit $Q^\pi(s, a)$ comme la valeur de l'état s quand on y effectue l'action a étant donnée la politique π :

$$Q^\pi(s, a) = \mathbb{E}_\pi[R(s) | s_0 = s, a_0 = a].$$

On parle de la **qualité** de la paire (s, a) .

En résumé : la valeur d'un état indique s'il est intéressant de passer par cet état pour notre problème d'optimisation et la qualité indique s'il est bien d'effectuer une action dans un état donné.

Propriété

Les fonctions V^π et Q^π sont solutions respectivement des équations suivante dites **de Bellman** :

$$V^\pi(s) = \sum_{a \in A} \pi(s, a) \sum_{s' \in S} \mathcal{P}(s, a, s') [\mathcal{R}(s, a, s') + \gamma V^\pi(s')]$$

$$Q^\pi(s, a) = \sum_{s' \in S} \mathcal{P}(s, a, s') [\mathcal{R}(s, a, s') + \gamma \sum_{a' \in A} \pi(s, a') Q^\pi(s, a')]$$

Définition

Un politique π est dite meilleure qu'une politique π' si $\forall s \in \mathcal{S}, V^\pi(s) \geq V^{\pi'}(s)$.

Comme on sait qu'il existe une politique optimale i.e meilleure que toutes les autres pour un modèle à horizon infini pondéré, on peut introduire la fonction valeur d'une politique optimale, notée V^* telle que :

$$V^*(s) = \max_{\pi} V^\pi(s) = Q^\pi(s, a) \forall s \in \mathcal{S}$$

d'où l'équation d'optimalité de Bellman pour V :

$$V^*(s) = \max_{a \in A} \sum_{s' \in \mathcal{S}} \mathcal{P}(s, a, s') [\mathcal{R}(s, a, s') + \gamma V^*(s')]$$

On a aussi la relation suivante :

$$V^*(s) = \max_{a \in \mathcal{A}} Q^*(s, a)$$

d'où l'équation d'optimalité de Bellman pour Q :

$$Q^*(s) = \sum_{s' \in \mathcal{S}} \mathcal{P}(s, a, s') [\mathcal{R}(s, a, s') + \gamma \max_{a' \in \mathcal{A}} Q^*(s', a')]$$

On peut maintenant s'attaquer à des algorithmes permettant de construire des politiques optimales ! Dans un premier temps dans des cas où toutes les informations sur le PDM sont disponibles puis le cas où \mathcal{P} et \mathcal{R} sont inconnus.

Problème de planification

Cadre du problème

On va supposer que l'on a un PDM à information complète. On veut donc calculer la valeur de la politique optimale et l'explicitier aussi.

- ① *Apprentissage par renforcement, Notes de cours*, Philippe Preux, GRAPPA
<https://philippe-preux.github.io/Documents/ndc-ar.pdf>
- ② <https://huggingface.co/learn/deep-rl-course>
- ③ *Reinforcement Learning : An introduction*, R.S. Sutton et A.G. Barto
- ④ *Introduction à l'apprentissage par renforcement*, A. Juton, V. Noël, R. Lila
- ⑤ Cours Houcine Senoussi
- ⑥ Cours Fidle <https://www.youtube.com/@CNRS-FIDLE>
- ⑦ Cours Stéphane Airiau
- ⑧ Cours Emmanuel Frank <https://irma.math.unistra.fr/~franck/talks/talk2021/RL.pdf>