UE Projet Altran 2020-2021

GRAFF Jonathan

January 28, 2021

Introduction - Contexte

Altran :

- conseil
- ingénierie
- recherche et développement

Introduction - Contexte

Altran :

- conseil
- ingénierie
- recherche et développement
- domaine : Automobile, Aéronautique, Spatial, Transport, Energie, Industrie, Communications, Electronique, Logiciel, Internet, Finance, Secteur Public ...

Introduction - Contexte

Altran :

- conseil
- ingénierie
- recherche et développement
- domaine : Automobile, Aéronautique, Spatial, Transport, Energie, Industrie, Communications, Electronique, Logiciel, Internet, Finance, Secteur Public ...
- 30 pays, 50000 employés

Le projet :
Altran Research

Le projet :
Altran Research $\rightarrow$ Future of Energy

Le projet :
Altran Research $\rightarrow$ Future of Energy $\rightarrow$ Sinbad

Le projet :
Altran Research → Future of Energy → Sinbad → Anagreen

Anagreen : collaboration avec ArcelorMittal, financé par l'ADEME (Agence de l'environnement et de la maîtrise de l'énergie)

Le but : développer un outil d'aide à la décision pour la récupération de la chaleur fatale dans l'industrie

La chaleur fatale : chaleur résiduelle issue d'un procédé et non utilisée par celui-ci.

La chaleur fatale : chaleur résiduelle issue d'un procédé et non utilisée par celui-ci.

- 16 % de la consommation énergétique nationale

La chaleur fatale : chaleur résiduelle issue d'un procédé et non utilisée par celui-ci.

- 16 % de la consommation énergétique nationale
- 36% de la consommation industrielle

La chaleur fatale : chaleur résiduelle issue d'un procédé et non utilisée par celui-ci.

- 16 % de la consommation énergétique nationale
- 36% de la consommation industrielle
- probablement sous-évalué

La chaleur fatale : chaleur résiduelle issue d'un procédé et non utilisée par celui-ci.

- 16 % de la consommation énergétique nationale
- 36% de la consommation industrielle
- probablement sous-évalué

Historiquement :

- peu d'intérêt jusqu'aux années 1970 (énergie peu chère, peu de conscience écologique)

La chaleur fatale : chaleur résiduelle issue d'un procédé et non utilisée par celui-ci.

- 16 % de la consommation énergétique nationale
- 36% de la consommation industrielle
- probablement sous-évalué

Historiquement :

- peu d'intérêt jusqu'aux années 1970 (énergie peu chère, peu de conscience écologique)
- prise de conscience aujourd'hui (énergie plus chère, taxes sur les rejets polluants...) d'où l'intérêt des entreprises

Un échangeur thermique est un dispositif permettant de transférer de la chaleur d'un flux chaud vers un flux plus froid, sans les mélanger
Trois types :

- à co-courant

Un échangeur thermique est un dispositif permettant de transférer de la chaleur d'un flux chaud vers un flux plus froid, sans les mélanger
Trois types :

- à co-courant
- à contre-courant

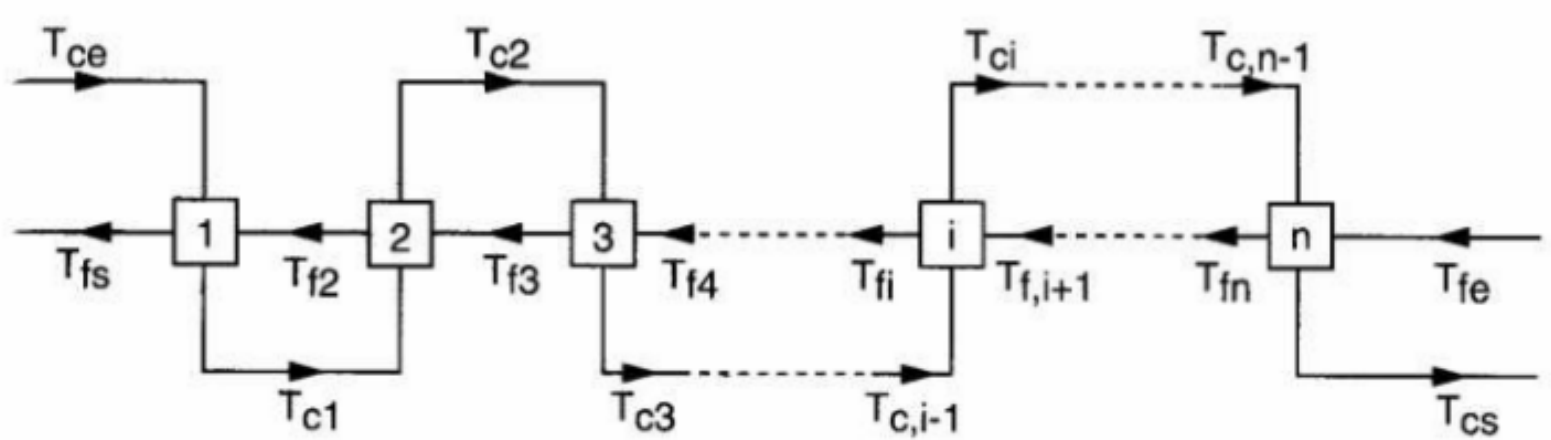
Un échangeur thermique est un dispositif permettant de transférer de la chaleur d'un flux chaud vers un flux plus froid, sans les mélanger

Trois types :

- à co-courant
- à contre-courant
- à courant croisés

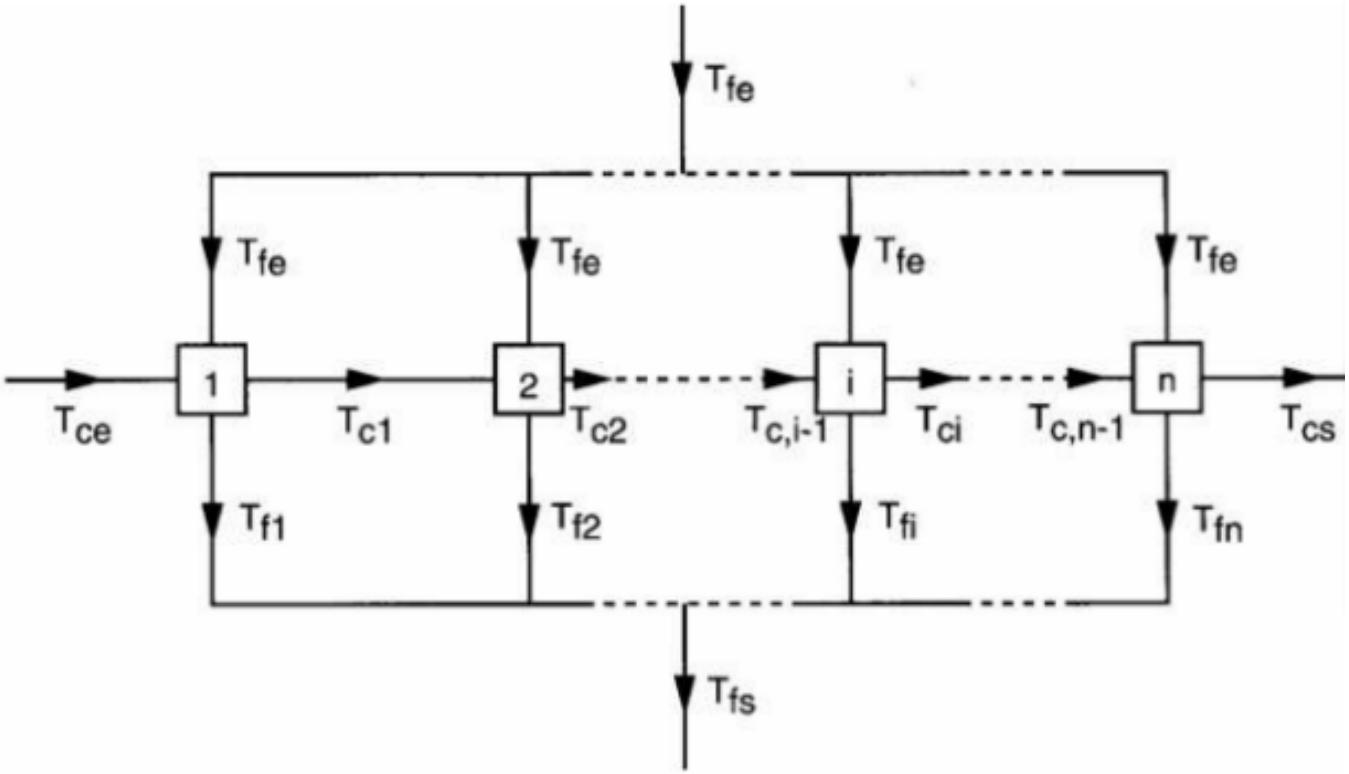
Plusieurs montages différents :

- en série :

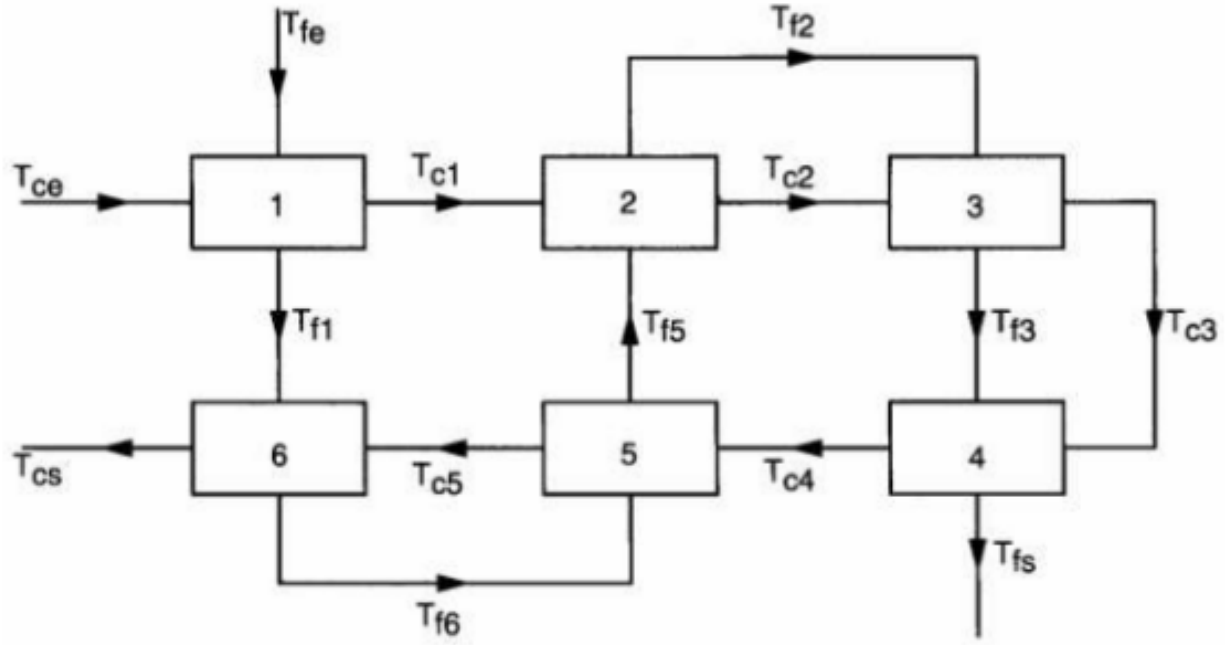


Plusieurs montages différents :

- un en parallèle et l'autre en série :



Plusieurs montages différents : en maillage :



Plusieurs montages différents :

Les deux en parallèle. Cette méthode est moins utilisée que les autres pour des raisons de performances moindres.

- flux chauds - flux froids

- flux chauds - flux froids
- Quels échangeurs placer ?

- flux chauds - flux froids
- Quels échangeurs placer ?
- Où les placer ?

- flux chauds - flux froids
- Quels échangeurs placer ?
- Où les placer ?
- Quand les placer ?

- flux chauds - flux froids
- Quels échangeurs placer ?
- Où les placer ?
- Quand les placer ?

Le problème est combinatoire : problème NP-difficile !

- flux chauds - flux froids
- Quels échangeurs placer ?
- Où les placer ?
- Quand les placer ?

Le problème est combinatoire : problème NP-difficile !
Pour 10 flux chauds et 10 flux froids, si on se limite à 15 échangeurs, il y a 3×10^{24} possibilités différentes, juste pour le placement.

Plusieurs solutions apportées pour ces problèmes :

- méthode empirique : repose sur des avis d'experts

Plusieurs solutions apportées pour ces problèmes :

- méthode empirique : repose sur des avis d'experts
- méthode du pincement : consiste à diviser le problème en deux sous-problèmes plus simples. Repose sur des principes thermodynamiques

Plusieurs solutions apportées pour ces problèmes :

- méthode empirique : repose sur des avis d'experts
- méthode du pincement : consiste à diviser le problème en deux sous-problèmes plus simples. Repose sur des principes thermodynamiques
- algorithmes mathématiques : gloutons, programmation linéaire ou non linéaire mixte entière, génétiques, stochastiques, recuit simulé...

Plusieurs solutions apportées pour ces problèmes :

- méthode empirique : repose sur des avis d'experts
- méthode du pincement : consiste à diviser le problème en deux sous-problèmes plus simples. Repose sur des principes thermodynamiques
- algorithmes mathématiques : gloutons, programmation linéaire ou non linéaire mixte entière, génétiques, stochastiques, recuit simulé...

Toutes ces solutions ont des avantages et des inconvénients, mais aucune n'est assurée de donner a solution optimale

Nouvelle solution : l'apprentissage par renforcement profond

Le but du projet ici est de vérifier si une méthode par apprentissage par réseaux de neurones à renforcement profond, pourrait résoudre ce problème d'optimisation combinatoire.

Nouvelle solution : l'apprentissage par renforcement profond

Le but du projet ici est de vérifier si une méthode par apprentissage par réseaux de neurones à renforcement profond, pourrait résoudre ce problème d'optimisation combinatoire.

L'idée vient de l'équipe de Google Brain qui a obtenu des résultats encourageants sur le problème du voyageur de commerce.

Nouvelle solution : l'apprentissage par renforcement profond

Le but du projet ici est de vérifier si une méthode par apprentissage par réseaux de neurones à renforcement profond, pourrait résoudre ce problème d'optimisation combinatoire.

L'idée vient de l'équipe de Google Brain qui a obtenu des résultats encourageants sur le problème du voyageur de commerce.

Le but du projet Anagreen est de se demander si ces mêmes résultats pourraient être transposables sur ce problème de réseaux d'échangeurs de chaleur.

Le principe

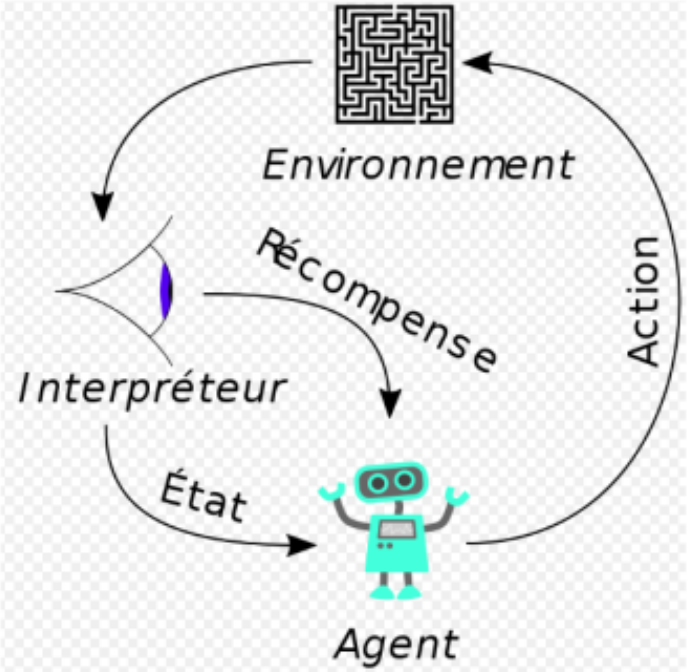
L'apprentissage par renforcement :

Un agent (robot) apprend par lui-même le comportement à adopter face à un problème à résoudre. L'agent est placé dans un environnement et peut avoir plusieurs états. Par rapport à chaque état, il peut effectuer un ensemble d'actions.

Le principe

L'apprentissage par renforcement :

Un agent (robot) apprend par lui-même le comportement à adopter face à un problème à résoudre. L'agent est placé dans un environnement et peut avoir plusieurs états. Par rapport à chaque état, il peut effectuer un ensemble d'actions.



Le principe :

- L'agent choisit son action (ou reçoit les probabilités d'action) selon une fonction appelée "politique".

Le principe :

- L'agent choisit son action (ou reçoit les probabilités d'action) selon une fonction appelée "politique".
- A chaque action effectuée, l'agent va changer d'état, et recevoir une récompense.

Le principe :

- L'agent choisit son action (ou reçoit les probabilités d'action) selon une fonction appelée "politique".
- A chaque action effectuée, l'agent va changer d'état, et recevoir une récompense.

Pendant la phase d'apprentissage, l'agent va alterner entre deux comportements :

Le principe :

- L'agent choisit son action (ou reçoit les probabilités d'action) selon une fonction appelée "politique".
- A chaque action effectuée, l'agent va changer d'état, et recevoir une récompense.

Pendant la phase d'apprentissage, l'agent va alterner entre deux comportements :

- une phase d'exploration où l'agent va tester aléatoirement de nouvelles actions

Le principe :

- L'agent choisit son action (ou reçoit les probabilités d'action) selon une fonction appelée "politique".
- A chaque action effectuée, l'agent va changer d'état, et recevoir une récompense.

Pendant la phase d'apprentissage, l'agent va alterner entre deux comportements :

- une phase d'exploration où l'agent va tester aléatoirement de nouvelles actions
- une phase d'exploitation, où la politique actuelle est éprouvée.

Le principe :

- L'agent choisit son action (ou reçoit les probabilités d'action) selon une fonction appelée "politique".
- A chaque action effectuée, l'agent va changer d'état, et recevoir une récompense.

Pendant la phase d'apprentissage, l'agent va alterner entre deux comportements :

- une phase d'exploration où l'agent va tester aléatoirement de nouvelles actions
- une phase d'exploitation, où la politique actuelle est éprouvée.

Les états terminaux : le jeu s'arrête (gagné ou perdu)

L'environnement renvoie à l'agent, en fonction de son état s et de son action a, un autre état s' et une récompense r, qui est un nombre réel. Mathématiquement, env est une fonction

$$\begin{array}{rccc} env : & S \times A & \rightarrow & S \times \mathbb{R} \\ & (s, a) & \mapsto & (s', r) \end{array}$$

L'environnement renvoie à l'agent, en fonction de son état s et de son action a, un autre état s' et une récompense r, qui est un nombre réel. Mathématiquement, env est une fonction

$$\begin{array}{rccc} env : & S \times A & \rightarrow & S \times \mathbb{R} \\ & (s, a) & \mapsto & (s', r) \end{array}$$

Le but est de trouver une suite de paires état-action (s, a) qui maximise la récompense G, où G vérifie :

$$G = \sum_t \gamma^t r_t$$

L'environnement renvoie à l'agent, en fonction de son état s et de son action a, un autre état s' et une récompense r, qui est un nombre réel. Mathématiquement, env est une fonction

$$\begin{array}{rccc} env : & S \times A & \rightarrow & S \times \mathbb{R} \\ & (s, a) & \mapsto & (s', r) \end{array}$$

Le but est de trouver une suite de paires état-action (s, a) qui maximise la récompense G, où G vérifie :

$$G = \sum_t \gamma^t r_t$$

γ est le facteur de réduction : sert à faire converger la somme et à rendre les récompenses immédiates plus importantes (en général ≈ 0.9).

La Q-fonction optimale

$$Q^*(s,a) := \max_{(s_t,a_t)_{t\geq 0}} \left(\sum_{t\geq 0} \gamma^t r_t \ \middle| s_0 = s, a_0 = a \right)$$

La Q-fonction optimale

$$Q^*(s,a) := \max_{(s_t,a_t)_{t\geq 0}} \left(\sum_{t\geq 0} \gamma^t r_t \ \middle| s_0 = s, a_0 = a \right)$$

La fonction Q^* est appelée la Q-fonction optimale. Il s'agit de la fonction qui renvoie la récompense maximale pouvant être obtenue à partir d'un état et d'une action de départ.

La Q-fonction optimale

$$Q^*(s,a) := \max_{(s_t,a_t)_{t\geq 0}} \left(\sum_{t\geq 0} \gamma^t r_t \ \middle| s_0 = s, a_0 = a \right)$$

La fonction Q^* est appelée la Q-fonction optimale. Il s'agit de la fonction qui renvoie la récompense maximale pouvant être obtenue à partir d'un état et d'une action de départ.
La fonction n'est pas connue, on va chercher à l'approcher.
Une fois cette fonction connue, on connaitra aussi la trajectoire à suivre pour obtenir ce maximum :

$$a_1 = \text{argmax}_{a\in A_{s_1}} Q^*(s_1,a), \ a_2 = \text{argmax}_{a\in A_{s_2}} Q^*(s_2,a) \text{ etc.}$$

Théorème de Bellmann

Soient (s, a) un couple état-action, et posons $(s', r) = env(s, a)$.
La fonction Q^* vérifie:

$$Q^*(s, a) = r + \gamma \max_{a' \in A_{s'}} Q^*(s', a')$$

Théorème de Bellmann

Soient (s, a) un couple état-action, et posons $(s', r) = env(s, a)$.
La fonction Q^* vérifie:

$$Q^*(s, a) = r + \gamma \max_{a' \in A_{s'}} Q^*(s', a')$$

Approcher la fonction Q^* par un algorithme itératif.
Si Q vérifie "à peu près" la relation précédente, alors Q sera proche de Q^*.

L'algorithme

1. on initialise Q à 0, ϵ à 1

L'algorithme

1. on initialise Q à 0, ϵ à 1
2. on choisit un état s aléatoirement

L'algorithme

1. on initialise Q à 0, ϵ à 1
2. on choisit un état s aléatoirement
3. on choisit aléatoirement une action avec une probabilité de ϵ : c'est la phase d'**exploration**.
   Et avec une probabilité de $1 - \epsilon$, on choisit l'action qui maximise $Q(s, a)$ (avec le Q courant) : c'est la phase d'**exploitation**.

L'algorithme

1. on initialise Q à 0, ϵ à 1
2. on choisit un état s aléatoirement
3. on choisit aléatoirement une action avec une probabilité de ϵ : c'est la phase d'**exploration**.
   Et avec une probabilité de $1 - \epsilon$, on choisit l'action qui maximise $Q(s, a)$ (avec le Q courant) : c'est la phase d'**exploitation**.
4. on calcule $s', r = env(s, a)$ et on met à jour :
$$Q(s, a) = r + \gamma \max_{a'} Q(s', a')$$
5. on diminue ϵ.

L'algorithme

1. on initialise Q à 0, ϵ à 1
2. on choisit un état s aléatoirement
3. on choisit aléatoirement une action avec une probabilité de ϵ : c'est la phase d'**exploration**.
   Et avec une probabilité de $1 - \epsilon$, on choisit l'action qui maximise $Q(s, a)$ (avec le Q courant) : c'est la phase d'**exploitation**.
4. on calcule $s', r = env(s, a)$ et on met à jour :
$$Q(s, a) = r + \gamma \max_{a'} Q(s', a')$$
5. on diminue ϵ.
6. si l'état s n'est pas terminal, on recommence à l'étape 3 avec $s = s'$. S'il l'est, on recommence à l'étape 2.

L'algorithme

1. on initialise Q à 0, ϵ à 1
2. on choisit un état s aléatoirement
3. on choisit aléatoirement une action avec une probabilité de ϵ : c'est la phase d'**exploration**.
   Et avec une probabilité de $1 - \epsilon$, on choisit l'action qui maximise $Q(s, a)$ (avec le Q courant) : c'est la phase d'**exploitation**.
4. on calcule $s', r = env(s, a)$ et on met à jour :
$$Q(s, a) = r + \gamma \max_{a'} Q(s', a')$$
5. on diminue ϵ.
6. si l'état s n'est pas terminal, on recommence à l'étape 3 avec $s = s'$. S'il l'est, on recommence à l'étape 2.
7. On arrête l'algorithme selon un critère d'arrêt, par exemple au bout d'un nombre fixé d'itérations

Apprentissage par renforcement profond

Le nombre d'états est en général immense, souvent même infini $\Rightarrow$ impossible de construire la fonction Q

Apprentissage par renforcement profond

Le nombre d'états est en général immense, souvent même infini $\Rightarrow$ impossible de construire la fonction Q

On utilise alors à la place un réseaux de neurones pour pouvoir prédire la valeur de la fonction Q, en général un réseau de neurones à plusieurs couches cachées, d'où le nom "profond".

Apprentissage par renforcement profond

Le nombre d'états est en général immense, souvent même infini $\Rightarrow$ impossible de construire la fonction Q

On utilise alors à la place un réseaux de neurones pour pouvoir prédire la valeur de la fonction Q, en général un réseau de neurones à plusieurs couches cachées, d'où le nom "profond".

En pratique, on part d'un couple état-action initial (s, a), et on construit un batch d'états-action en suivant la même stratégie que précédemment, selon ϵ.

Apprentissage par renforcement profond

Le nombre d'états est en général immense, souvent même infini $\Rightarrow$ impossible de construire la fonction Q

On utilise alors à la place un réseaux de neurones pour pouvoir prédire la valeur de la fonction Q, en général un réseau de neurones à plusieurs couches cachées, d'où le nom "profond".

En pratique, on part d'un couple état-action initial (s, a), et on construit un batch d'états-action en suivant la même stratégie que précédemment, selon ϵ.

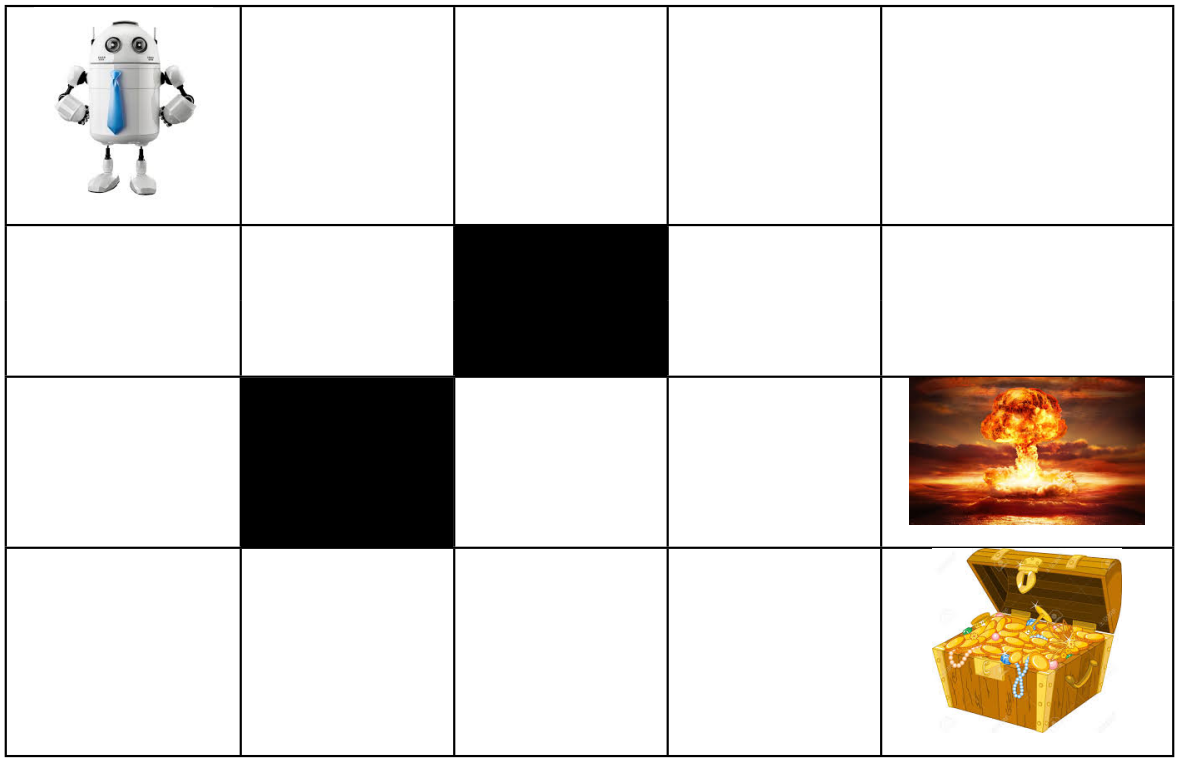
Une fois ce batch construit, on l'utilise pour modifier les poids ω du réseau de façon à ce que la distance entre les $Q(s, a)$ et les $r + \gamma \max_{a'} Q(s', a')$ soit minimale (où s' est l'état renvoyé après l'action a).

Implémentation d'un exemple : GridWorld

L'environnement sera une grille :

- l'agent en haut à gauche : $(0, 0)$
- en-bas à droite : un trésor
- au-dessus : une bombe
- des obstacles

Les états terminaux : la bombe et le trésor.



Les actions

Il y a quatre actions possibles

- haut
- gauche
- droite
- haut

Les actions

Il y a quatre actions possibles

- haut
- gauche
- droite
- haut

Si il y a un bord ou un obstacle, l'agent reste au même endroit.

Les récompenses

- +10 pour un déplacement sur le trésor
- -10 pour un déplacement sur la bombe
- -1 pour chaque déplacement

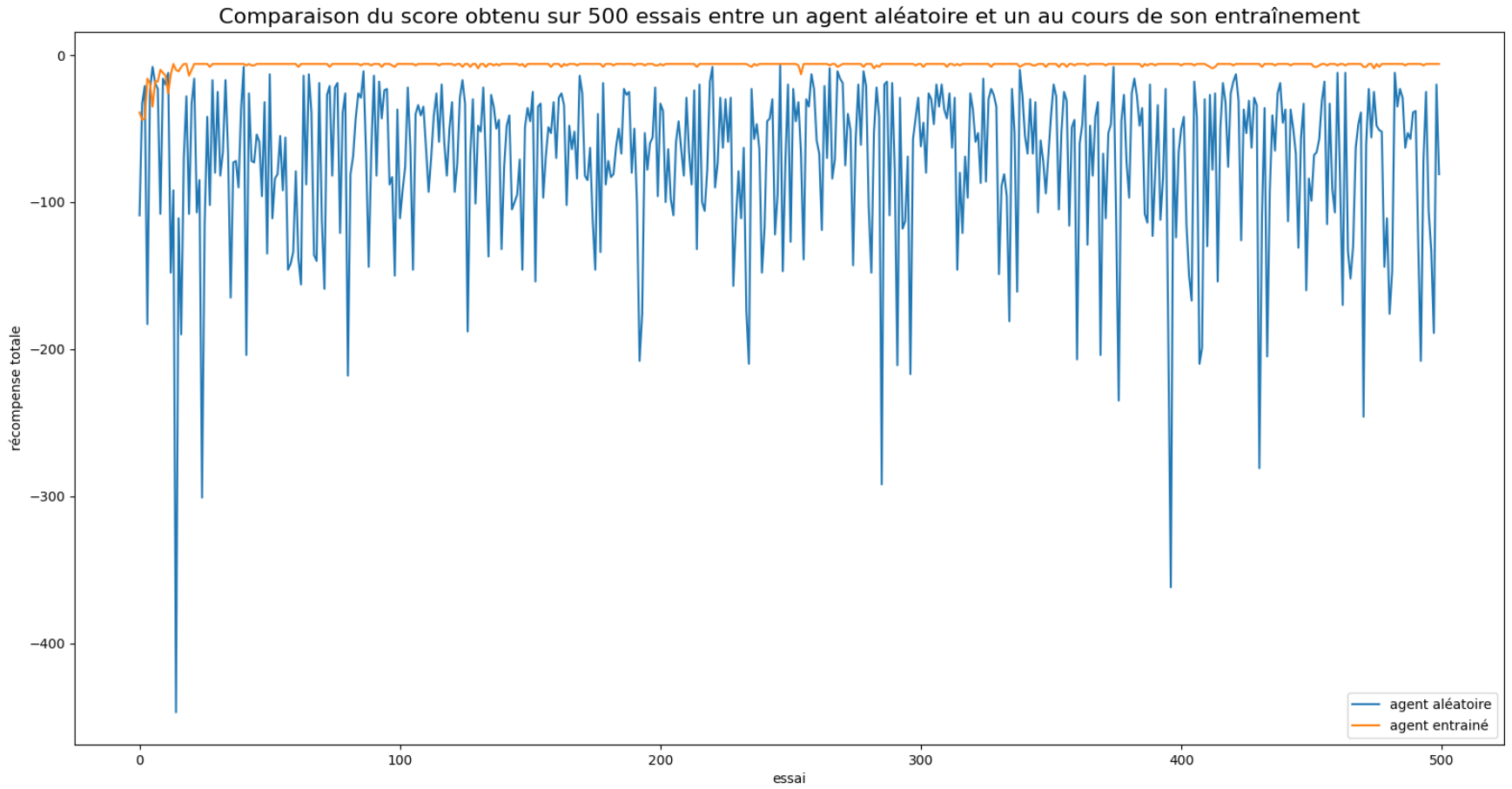
Les récompenses

- +10 pour un déplacement sur le trésor
- -10 pour un déplacement sur la bombe
- -1 pour chaque déplacement

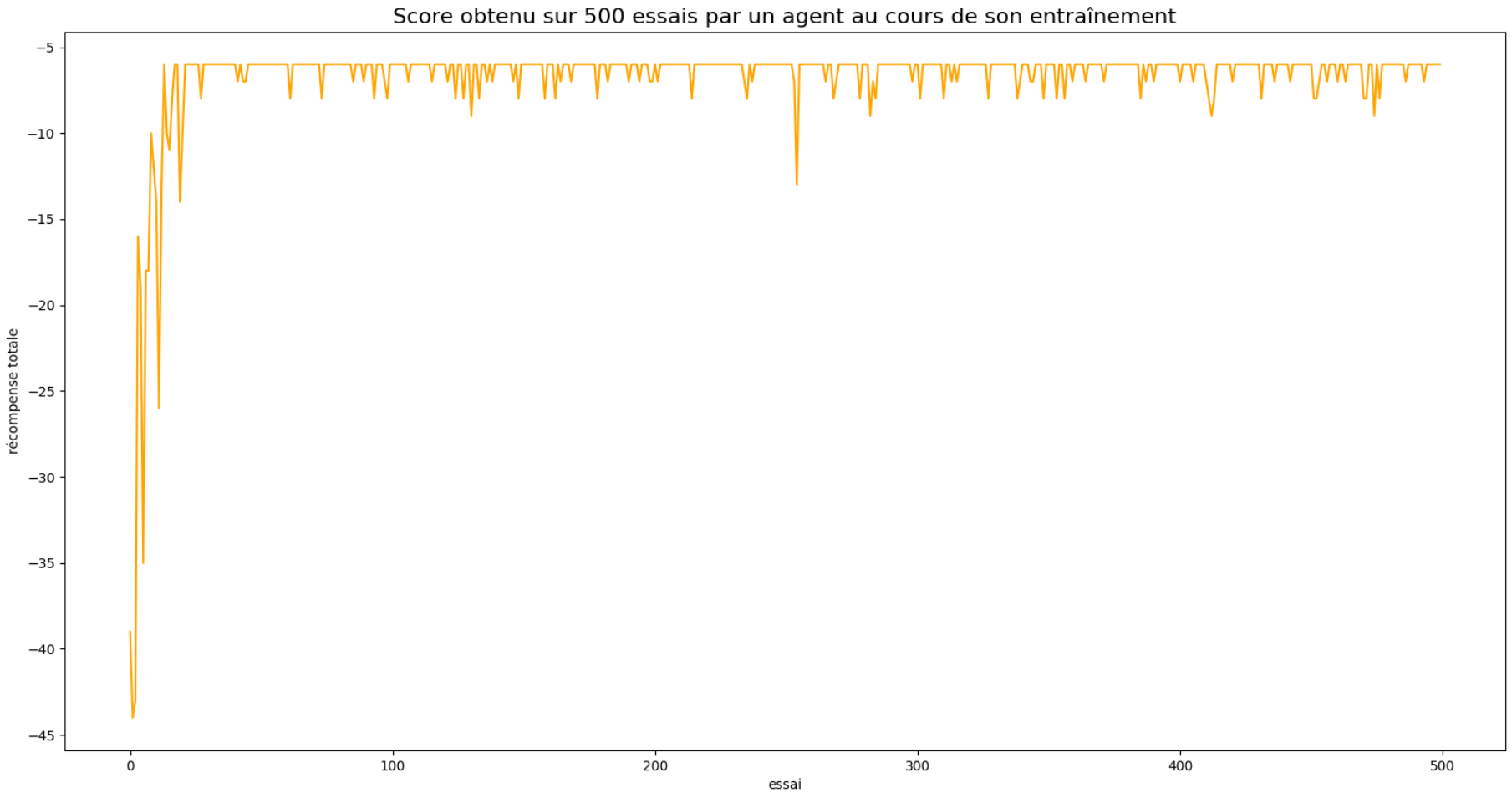
-1	-1	-1	-1	-1
-1	-1	0	-1	-1
-1	0	-1	-1	-10
-1	-1	-1	-1	10

Les résultats

Un agent aléatoire et un agent apprenant



Les résultats



La Q-table

La Q-table est composée, pour chaque case de la grille, d'un dictionnaire.

- Ses quatre clés sont les quatre actions
- Ses valeurs sont celles de Q(état, action)

Voici une représentation dans une grille 3×3 :

0 0 *A* 0 0	0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0	0 0 *B* 0 0
0 0 0 0	0 0 0 0	0 0 *T* 0 0

Si l'agent va à droite, il recevra une récompense de -1. Donc :

$$Q((0,0),"droite") = -1 + \max_{a'} Q((0,1),a') = -1 + 0 = -1$$

$\begin{matrix} & 0 & \\ 0 & & -1 \\ & 0 & \end{matrix}$	$\begin{matrix} & 0 & \\ 0 & A & 0 \\ & 0 & \end{matrix}$	$\begin{matrix} & 0 & \\ 0 & & 0 \\ & 0 & \end{matrix}$
$\begin{matrix} & 0 & \\ 0 & & 0 \\ & 0 & \end{matrix}$	$\begin{matrix} & 0 & \\ 0 & & 0 \\ & 0 & \end{matrix}$	$\begin{matrix} & 0 & \\ 0 & B & 0 \\ & 0 & \end{matrix}$
$\begin{matrix} & 0 & \\ 0 & & 0 \\ & 0 & \end{matrix}$	$\begin{matrix} & 0 & \\ 0 & & 0 \\ & 0 & \end{matrix}$	$\begin{matrix} & 0 & \\ 0 & T & 0 \\ & 0 & \end{matrix}$

Si l'agent va ensuite à droite et en-bas, il recevra à chaque fois une récompense de -1. A la fin du premier épisode, on aura donc :

Si l'agent va ensuite à droite et en-bas, il recevra à chaque fois une récompense de -1. A la fin du premier épisode, on aura donc :

$\begin{matrix} 0 \\ 0 & -1 \\ 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 & -1 \\ 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 & 0 \\ -10 \end{matrix}$
$\begin{matrix} 0 \\ 0 & 0 \\ 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 & 0 \\ 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 & BA & 0 \\ 0 \end{matrix}$
$\begin{matrix} 0 \\ 0 & 0 \\ 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 & 0 \\ 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 & T & 0 \\ 0 \end{matrix}$

Imaginons un deuxième épisode où le chemin choisi sera

"droite - bas - bas - droite"

Alors :

Imaginons un deuxième épisode où le chemin choisi sera

"droite - bas - bas - droite"

Alors :

$\begin{matrix} 0 \\ 0 & -1 \\ 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 & -1 \\ -1 \end{matrix}$	$\begin{matrix} 0 \\ 0 & 0 \\ -10 \end{matrix}$
$\begin{matrix} 0 \\ 0 & 0 \\ 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 & 0 \\ -1 \end{matrix}$	$\begin{matrix} 0 \\ 0 & B & 0 \\ 0 \end{matrix}$
$\begin{matrix} 0 \\ 0 & 0 \\ 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 & 10 \\ 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 & TA & 0 \\ 0 \end{matrix}$

Si à un troisième épisode, le chemin choisi est

"bas - droite - bas - droite"

Si à un troisième épisode, le chemin choisi est

"bas - droite - bas - droite"

$\begin{matrix} & 0 & \\ 0 & & -1 \\ & -1 & \end{matrix}$	$\begin{matrix} & 0 & \\ 0 & & -1 \\ & -1 & \end{matrix}$	$\begin{matrix} & 0 & \\ 0 & & 0 \\ & -10 & \end{matrix}$
$\begin{matrix} & 0 & \\ 0 & & -1 \\ & 0 & \end{matrix}$	$\begin{matrix} & 0 & \\ 0 & & 0 \\ & 9 & \end{matrix}$	$\begin{matrix} & 0 & \\ 0 & B & 0 \\ & 0 & \end{matrix}$
$\begin{matrix} & 0 & \\ 0 & & 0 \\ & 0 & \end{matrix}$	$\begin{matrix} & 0 & \\ 0 & & 10 \\ & 0 & \end{matrix}$	$\begin{matrix} & 0 & \\ 0 & TA & 0 \\ & 0 & \end{matrix}$

Avec deux autres épisodes on obtiendra donc la Q-table suivante :

Avec deux autres épisodes on obtiendra donc la Q-table suivante :

$\begin{matrix} 0 \\ 0 & -1 \\ 7 \end{matrix}$	$\begin{matrix} 0 \\ 0 & -1 \\ -1 \end{matrix}$	$\begin{matrix} 0 \\ 0 & 0 \\ -10 \end{matrix}$
$\begin{matrix} 0 \\ 0 & 8 \\ 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 & 0 \\ 9 \end{matrix}$	$\begin{matrix} 0 \\ 0 & B & 0 \\ 0 \end{matrix}$
$\begin{matrix} 0 \\ 0 & 0 \\ 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 & 10 \\ 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 & TA & 0 \\ 0 \end{matrix}$

Les résultats pour la grille 5×3

−2.02 −1.07 4.00 −3.30	0.60 0.55 5.00 −2.59	2.11 −0.57 6.00 0.60	1.41 1.71 − 1.21 7.00	−0.8 −0.80 − 0.80 −0.10
−2.56 −2.60 − 2.54 −2.48	1.29 2.51 − 2.57 −2.58	0.00 0.00 0.00 0.00	4.07 6.46 − 0.40 8.00	−0.46 3.46 − 0.50 −0.47
−1.78 −1.80 − 1.80 −1.20	0.00 0.00 0.00 0.00	−0.10 0.04 5.14 −0.10	1.56 1.35 − 4.69 9.00	0.00 0.00 0.00 0.00
−1.20 −1.12 1.03 −1.10	−0.59 −0.56 4.31 −0.60	−0.10 −0.34 8.13 −0.10	6.28 2.25 10.00 4.70	0.00 0.00 0.00 0.00

Merci pour votre attention !