République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Supérieure d'Informatique

FASI - Travail pratique

Apprentissage par renforcement en utilisant la méthode Q-Learning avec des applications pratiques en python et la librairie gym

Par : **SMAIL Sabrina**

Du groupe : **SIT2**

Année universitaire : 2020/2021

Tables des matières

1. Introduction

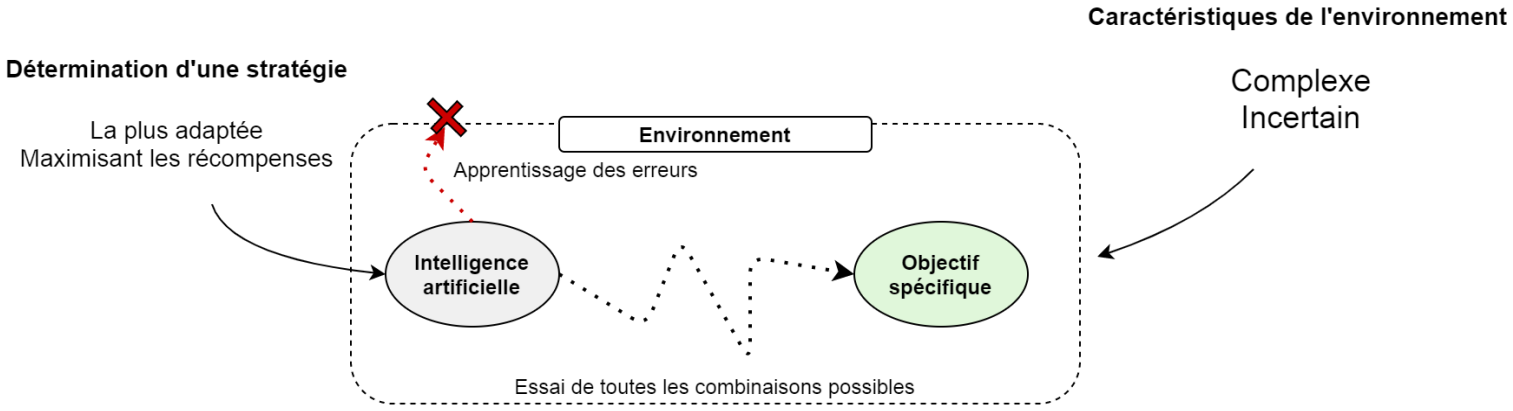
Ce TP, réalisé par Sabrina SMAIL du groupe SIT2, a pour but de présenter la méthode du Q-Learning dans le contexte de l'apprentissage par renforcement.

Durant cet exposé, je présenterais brièvement l'apprentissage par renforcement, suivi par la méthode du Q-Learning de façon théorique, ensuite je passerai aux cas pratiques en utilisant la librairie Open Gym à travers l'un de ses environnements offerts en utilisant le Q-Learning.

2. L'apprentissage par renforcement

2.1. Présentation

L'apprentissage par renforcement (RL) est l'une des méthodes du machine learning avec l'apprentissage supervisé et l'apprentissage non supervisé. C'est une technique qui permet à une intelligence artificielle d'atteindre un objectif spécifique dans un environnement incertain et potentiellement complexe sans aide ou indications au préalable. Pour y parvenir, L'agent essaye toutes les actions possibles et apprend de ses erreurs selon les observations fournies par son environnement, lui permettant ainsi de déterminer la stratégie d'action la plus adaptée pour résoudre le problème tout en maximisant ses récompenses.



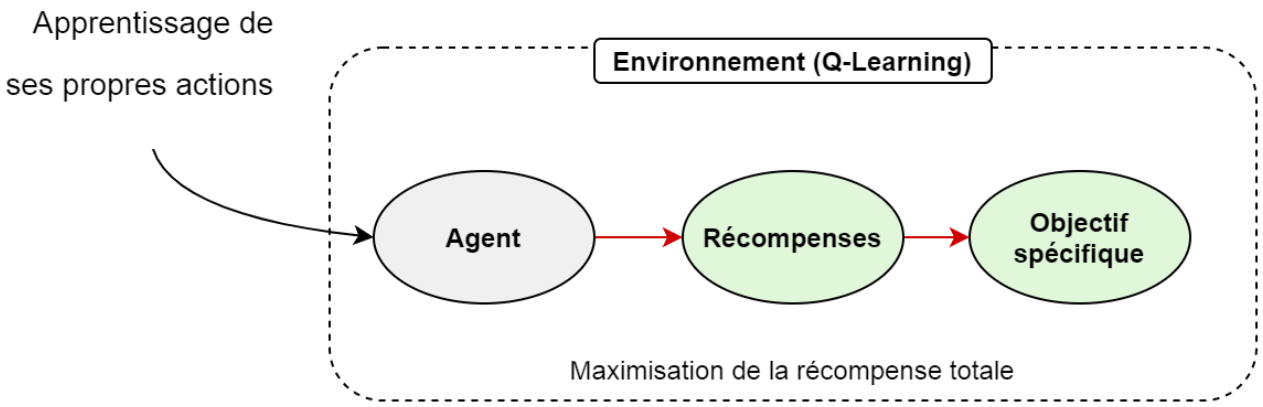
2.2. Historique

Les algorithmes d'apprentissage par renforcement sont variés, on cite le TD-learning proposé en 1988, et le Q-learning issu de la thèse de Chris Watkins en 1989 et publié réellement en 1992.

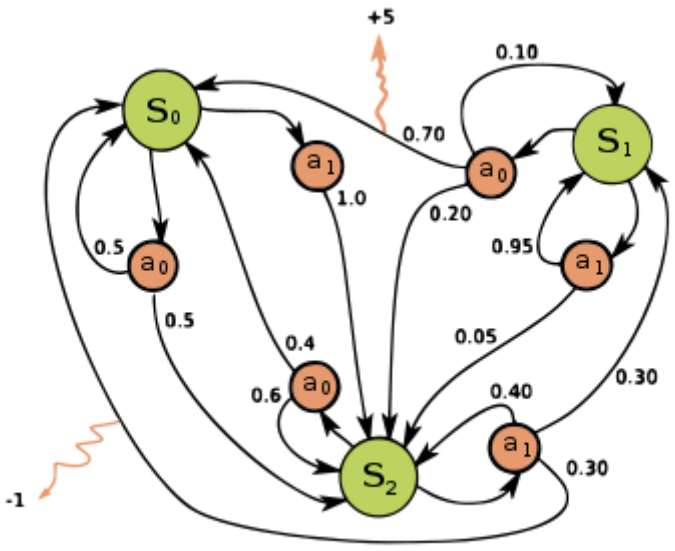
3. Le Q-Learning

3.1. Présentation

Le Q-Learning est un algorithme hors politique de l'apprentissage par renforcement, basé sur l'équation de Bellman. Il est considéré hors politique pour la simple raison qu'il apprend de ses propres actions hors politique actuelle. Le Q-learning cherche à apprendre une politique optimale de sélection d'actions qui maximise la récompense totale.



Une des caractéristiques majeures du Q-learning est sa capacité à comparer les récompenses probables résultant de la prise d'actions sans connaissance initiale de l'environnement. On peut modéliser un tel système avec un processus de décision markovien inconnu par l'agent qui évolue, car cet agent découvre son environnement au fur et à mesure. La figure ci-dessous représente un tel environnement markovien où les récompenses sont les nœuds colorés, et les arcs représentant les espérances des récompenses.



3.2. Historique

L'algorithme du Q-learning a été introduit par Chris Watkins en 1989, et a été publié en 1992. Il a donné naissance à de nombreux variants comme le Deep Q-learning, le Double Q-learning, le Delayed Q-learning ainsi que l'algorithme Greedy GQ.

Le Delayed Q-Learning et le double Q-learning sont deux extensions du Q-learning qui sont utilisées dans l'apprentissage par renforcement. Le Delayed Q-Learning retarde simplement toute estimation jusqu'à ce qu'il y ait un échantillon significatif d'observations. Le Double Q-learning comprend deux tables Q pour réduire le biais.

3.3. Concepts clés

3.3.1. L'environnement

Pour le Q-learning, il s'agit d'un environnement inconnu et à explorer par un agent. On peut imaginer un labyrinthe truffé de récompenses et de pièges sans aucune connaissance préalable de leurs dispositions.

3.3.2. L'agent

Il évolue dans l'environnement en le tâtonnant dans la phase exploratoire afin de mettre à jour les espérances d'atteinte des récompenses selon les actions prises. Tandis que dans la phase d'exploitation, l'agent se fie à ces espérances calculées afin de déterminer les meilleures actions à prendre.

3.3.3. La fonction Q(état, action)

La fonction Q prend deux entrées : l'état et l'action et retourne la future récompense estimée pour l'action à cet état.

$$Q(s_t, a_t)^{\pi} = \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots | s_t, a_t]$$

Il s'agit donc d'une estimation de la récompense à obtenir parmi toutes les récompenses R prévues :

- Lorsque l'agent se trouve à l'état **st**.
- Lorsque l'agent prend l'action **at** : L'action représente la transition d'état.

- Lorsque l'agent suit une même politique **π** durant toute son évolution dans l'environnement. Dans la plupart des cas, la politique est le choix de la meilleure récompense.
- Lorsque les récompenses décroissent à chaque action effectuée, avec un taux **γ** appartenant à l'intervalle]0,1]

3.3.4. La table du Q-Learning

La table du Q-learning est une matrice (n,m) où :

- Les indices des lignes [[1, n]] représentent les états possibles dans le processus de décision markovien.
- Les indices des colonnes [[1, m]] représentent les actions pouvant être prises à partir d'un état.
- La cellule (i j) contient le score que procure l'action j pour l'état i.

Cette table est initialisée avec des zéros. Elle est remplie par l'agent au cours de la phase d'exploration, tandis qu'elle est exploitée lors de la phase d'exploitation.

Actions

Etats	A1	A2		Am
S0	Q(S0, A1)	Q(S0, A2)	...	Q(S0, Am)
S1	Q(S1, A1)	Q(S1, A2)	...	Q(S1, Am)
	...	...	...	...
Sn	Q(Sn, A1)	Q(Sn, A2)	...	Q(Sn, Am)

3.3.5. La mise à jour de la table Q

$$Q(s_t, a_t)_{new} = Q(s_t, a_t)_{old} + \alpha[r + \gamma max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)_{old}]$$

La mise à jour de la table Q se fait lors de la phase d'exploration de l'environnement par l'agent. Lorsque l'agent prend une action aléatoire, alors selon s'il a atteint une récompense

ou non, et selon les espérances visibles au prochain état et précédemment renseignées, il peut mettre à jour les espérances de son état actuel.

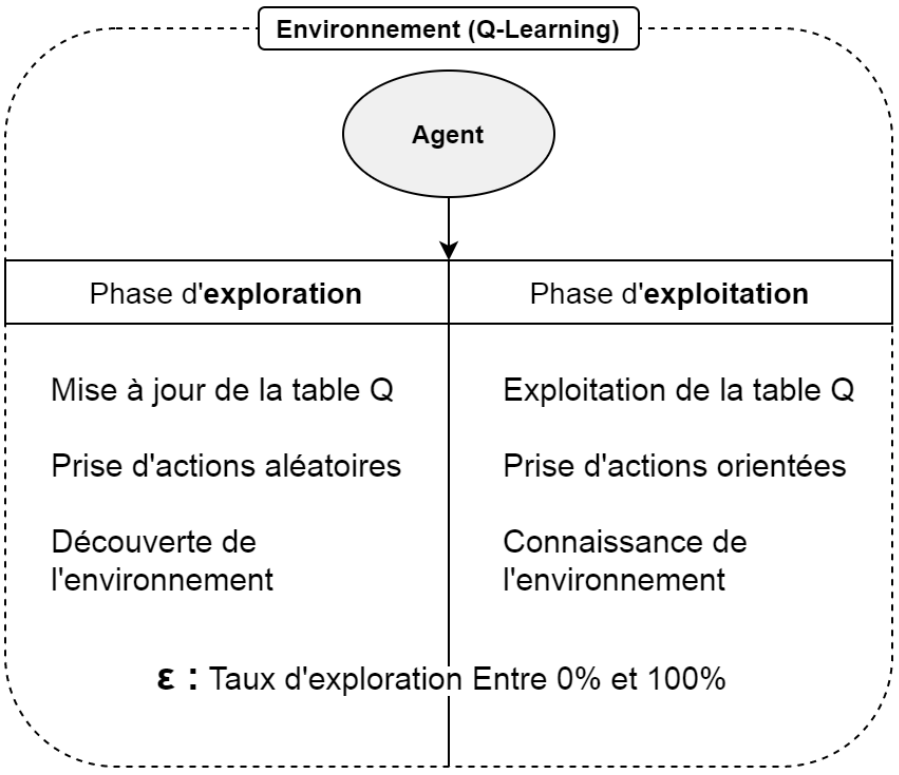
Q(st, at)new désigne la nouvelle espérance de récompense pour l'agent, en prenant l'action at, lorsqu'il se trouve à l'état st, et elle est obtenue par la formule précédente, et elle est en fonction de :

- Q(st,at)old, l'ancienne espérance.
- max Q(st+1, at+1), qui est l'espérance maximale en prenant l'action la maximisant lorsque l'agent se trouve à l'état st+1.
- Certains coefficients :
    - **γ** pour décroitre les récompenses au fil du temps. Il est parfois fixé à 0.9 puis affiné par la suite.
    - **α** qui est le **taux d'apprentissage** (Learning rate). C'est une qui est fixée selon les observations faites.
    - **r** qui est la récompense de l'état actuel.

3.4. L'algorithme du Q-Learning

Il se déroule en deux phases, pouvant être combinées avec un taux **ε**.

La figure ci-dessous expose le déroulement de l'algorithme du Q-learning.



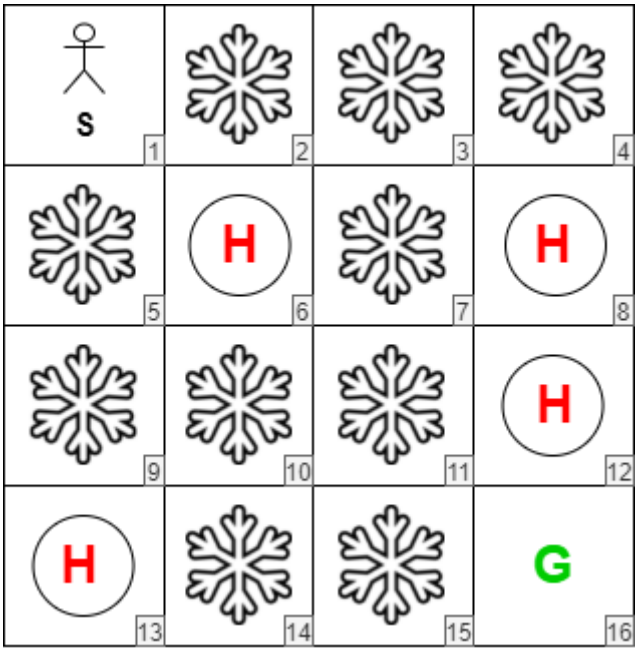
3.5. Exemples dans un environnement de programmation

3.5.1. La librairie gym

Gym est une bibliothèque Python qui offre une collection d'environnements pour développer et comparer les algorithmes d'apprentissage par renforcement.

3.5.2. Exemple 1 intégré à Gym : Le frisbee

Il s'agit d'un exemple intégré à Gym afin d'apprendre à manipuler l'environnement Q-learning.



Il y a une grille de 16 cases. Un agent S doit attendre son objectif (frisbee) en évitant les trous H. Cet agent n'a aucune connaissance de son environnement initial.

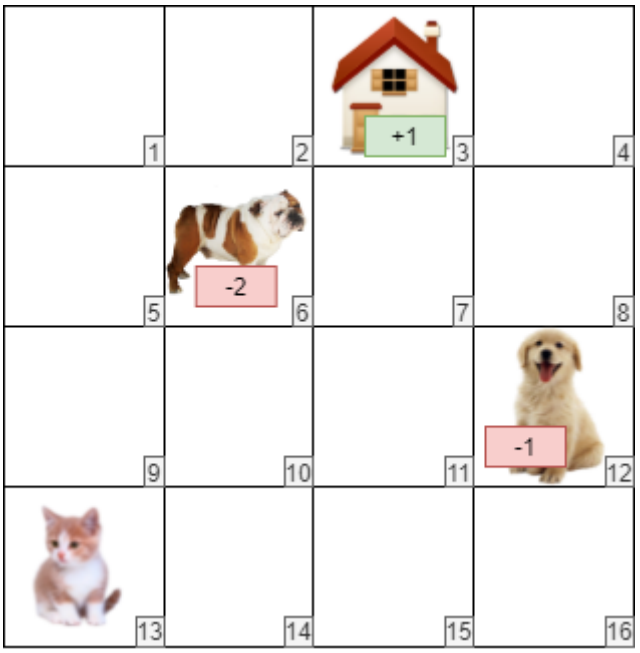
Le code de l'exemple est commenté et expliqué dans le fichier.
La grille dans l'environnement Gym :

```
SFFF
FHFH
FFFH
HFFG
```

La table Q de l'exemple exécuté avec Gym :

```
PS D:\TP\SMAIL_SABRINA_TEST> & D:/python/python.exe d:/TP/SMAIL_SABRINA_TEST/FrozenLake.py
[[0.14522031 0.14021782 0.15667303 0.13870695]
 [0.06121965 0.0842883  0.05555789 0.134005  ]
 [0.15334375 0.12969249 0.1324745  0.10886758]
 [0.05972634 0.04279943 0.08648425 0.10851028]
 [0.21486018 0.15435364 0.15291582 0.08863465]
 [0.         0.         0.         0.        ]
 [0.1027937  0.16350086 0.11870136 0.00119511]
 [0.         0.         0.         0.        ]
 [0.11444588 0.07100782 0.0441134  0.31724031]
 [0.18439945 0.36343811 0.26250688 0.19246717]
 [0.25484013 0.16969104 0.22601592 0.1372631 ]
 [0.         0.         0.         0.        ]
 [0.         0.         0.         0.        ]
 [0.1076838  0.35071997 0.56225232 0.27088709]
 [0.46235271 0.55730128 0.60122039 0.52365121]
 [0.         0.         0.         0.        ]]
```

3.5.3. Exemple 2 : Le chat



Un chat ne connaissant pas son chemin, veut atteindre la maison tout en évitant les chiens.

Le code de l'exemple est commenté et expliqué dans le fichier.

La table Q de l'exemple exécuté avec Gym :

```
PS D:\TP\SMAIL_SABRINA_TEST> & D:/python/python.exe d:/TP/SMAIL_SABRINA_TEST/codeChatSMAIL_SABRINA.py
----------la grille-----------
0       0       1       0
0       -2      0       0
0       0       0       -1
CHAT    0       0       0
-------------La qtable :--------------
[[ 0.81450625  0.77378094  0.77378094  0.88041085]
 [ 0.95       -1.0975      0.78853084  1.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.90053382  0.          0.53334661]
 [ 0.7857507   0.81450625  0.77884357 -1.0975    ]
 [ 0.95        0.85708934  0.77494915  0.94981842]
 [ 1.          0.9025     -1.0975      0.9025    ]
 [ 0.84056699 -0.183454    0.95        0.89963648]
 [ 0.77378094  0.77378094  0.81450625  0.857375  ]
 [-1.0975      0.81450625  0.81450625  0.9025    ]
 [ 0.95        0.857375    0.857375   -0.21904494]
 [ 0.81450625  0.77502278  0.77379467  0.81449094]
 [ 0.81450625  0.77378056  0.77378091  0.81450548]
 [ 0.857375    0.81449171  0.77376259  0.85689277]
 [ 0.9025      0.7462592   0.79170529  0.75841949]
 [-0.14437372  0.          0.85670312  0.47708994]]
```

4. Conclusion

À travers ce travail pratique, la méthode du Q-learning a pu être découverte dans son côté théorique et également testée sur deux exemples.

5. Références

https://gym.openai.com/docs/

https://gym.openai.com/envs/FrozenLake-v0/

http://www-igm.univ-mlv.fr/~dr/XPOSE2014/Machin_Learning/A_Q-Learning.html

http://www2.ift.ulaval.ca/~lamontagne/ift17587/modules/module6/renforcement.pdf

https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide/

https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html

q-learning :
https://datascientest.com/q-learning-le-machine-learning-avec-apprentissage-par-renforcement#:~:text=Le%20%C2%AB%20reinforcement%20learning%20%C2%BB%20ou%20apprentissage,t%C3%A2ches%20complexes%20de%20fa%C3%A7on%20autonome.

https://medium.com/free-code-camp/diving-deeper-into-reinforcement-learning-with-q-learning-c18d0db58efe