
INTELLIGENCE BIO-INSPIRÉE

TRAVAUX PRATIQUES

COMPTE-RENDU DES TRAVAUX PRATIQUES SUR L'APPRENTISSAGE PROFOND
PAR RENFORCEMENT

RÉALISÉ PAR

SUBLET GARY – 11506450
VILLERMET QUENTIN – 11507338

Polytech Lyon
Université Claude Bernard Lyon 1

2019

Table des matières

1	Deep Q-network sur CartPole	2
1.1	Début	2
1.1.1	Question 1	2
1.1.2	Question 2	2
1.2	Experience replay	5
1.2.1	Question 3	5
1.2.2	Question 4	5
1.3	Deep Q-learning	5
1.3.1	Question 5	5
1.3.2	Question 6	5
1.3.3	Question 7	5
1.3.4	Question 8	5

1 Deep Q-network sur CartPole

1.1 Début

1.1.1 Question 1

Le code de l'agent aléatoire est entièrement contenu dans le fichier `cartpole.py`, trouvable dans les sources du projet.

Il est presque entièrement basé sur l'exemple des créateurs de `gym` présent sur le GitHub du module.

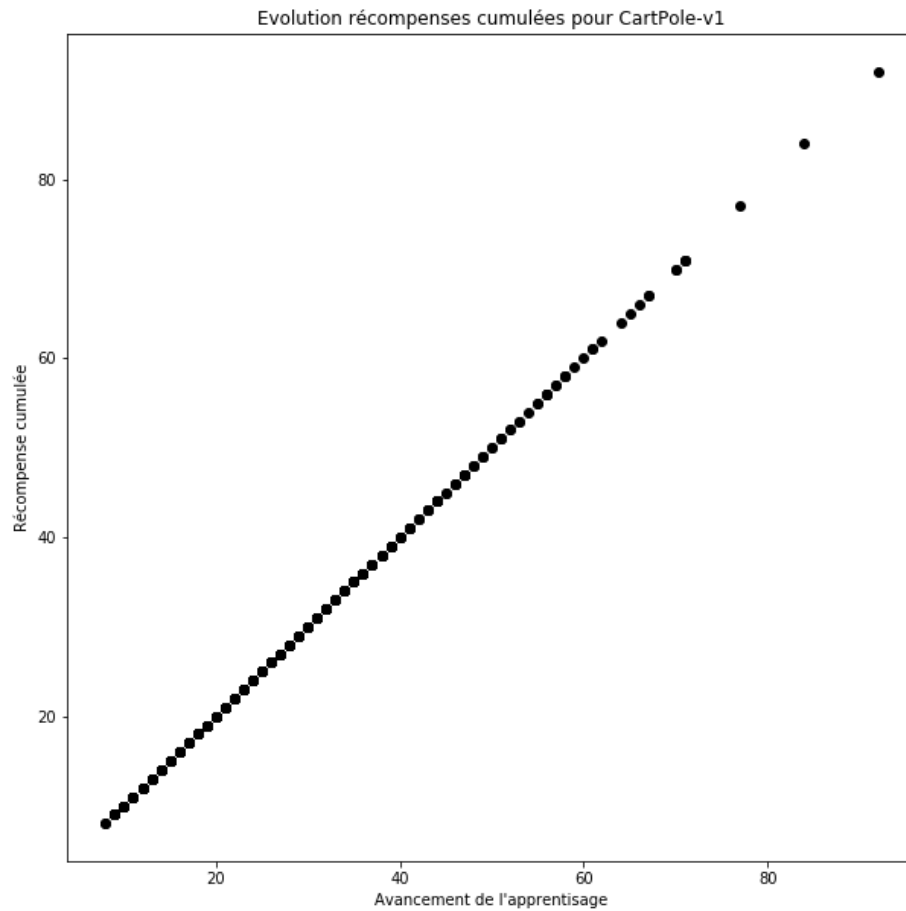
1.1.2 Question 2

NB : En plus de l'affichage demandé, nous avons pris l'initiative de créer une visualisation de la récompense cumulée au fil des épisodes (dénotant en quelque sorte le « succès » de la session d'apprentissage).

Nous avons opté pour un affichage `matplotlib`, avec *mean-pooling* pour alléger le graphe en cas de grand nombre d'épisodes lorsque nécessaire.

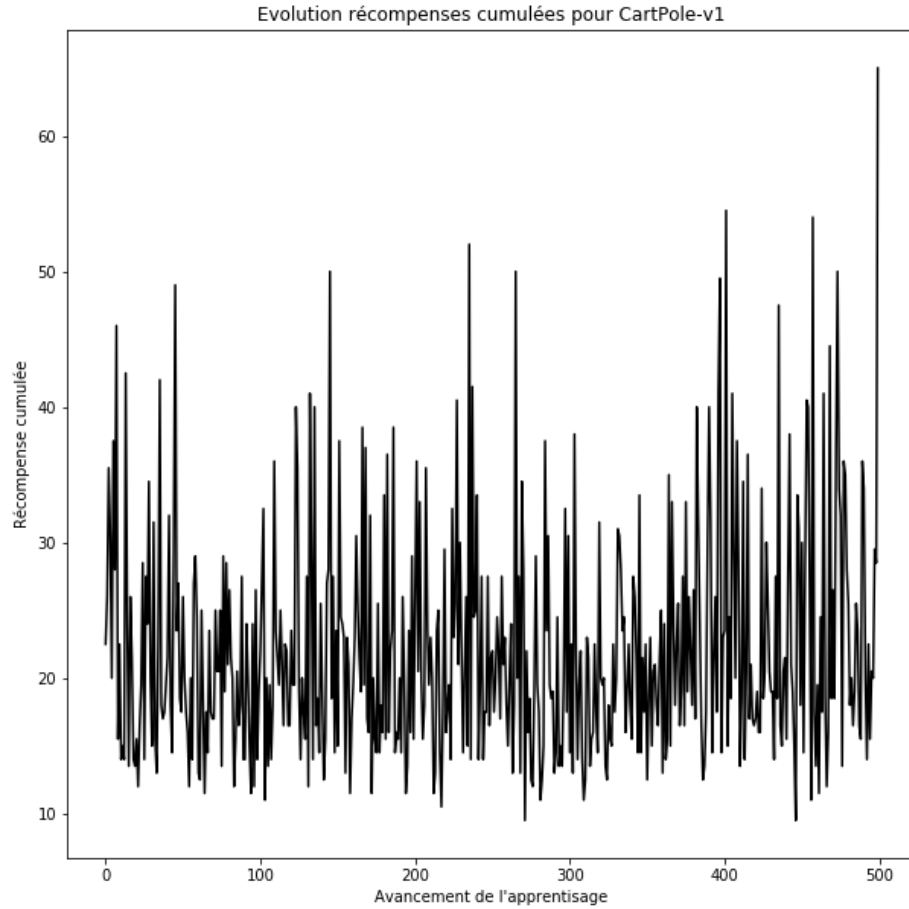
Dans notre premier cas, soit celui d'un agent aléatoire sur l'environnement *CartPole-V1*, le graphe des récompenses cumulées en fonction du nombre d'interactions aboutit forcément à un nuage de points suivant la fonction $f(x) = |x|$.

On constate peu d'*outliers* (qu'on considérera ici comme ayant $x > 60$), en raison du caractère aléatoire des décisions.



Ce résultat est sans surprise, puisque toute interaction entraîne une récompense de 1 dans cet environnement.

La deuxième visualisation n'est pour le moment pas très utile non plus, puisque notre agent n'apprend pas et agit de façon aléatoire.



Ces deux affichages prendront leur pertinence plus tard dans le développement, lorsque notre agent commencera à apprendre de son environnement. Il nous indiqueront respectivement la rapidité d'apprentissage des épisodes individuels, et l'efficacité générale de l'apprentissage en fonction du temps passé (*i.e.* des épisodes).

1.2 Experience replay

1.2.1 Question 3

Le code du buffer *experience replay* est dans le fichier `deeprl_agent.py`, plus particulièrement la fonction `experience` de la classe `Agent`.

1.2.2 Question 4

À partir de cette question, nous basculons sur un système de notebooks *Jupyter* pour une implémentation plus rapide des tests.

Le code de cette question est donc dans le fichier `experience_replay.ipynb`.

1.3 Deep Q-learning

L'implémentation du Q-learning se fait dans la continuation de la partie précédente, donc dans le fichier `experience_replay.ipynb`.

1.3.1 Question 5

Pour notre premier exemple, on construit donc un réseau avec 4 neurones d'entrée et 2 neurones de sortie – Dans l'environnement *CartPole-v1*, l'observation est de type `Box(4,)` et le domaine des actions est un `Discrete(2)` (en l'occurrence 0 et 1 pour bouger à gauche et à droite respectivement).

1.3.2 Question 6

1.3.3 Question 7

1.3.4 Question 8