

Rendu TP deep RL

Adrien Guédet

17 Mars 2020

1 Introduction

L'objectif de ce tp est d'implémenter un algorithme d'apprentissage profond par renforcement, dans un environnement permettant une réplification facile des expériences, sur les jeux "cartpole" et "breakout".

2 Cartpole

2.1

Le fichier "RandomAgent.py" contient un agent simple qui, une fois créé à partir d'une liste d'actions possibles, renvoi une action aléatoire de cet ensemble à chaque fois. Pour chaque exécution, que ce soit en apprentissage ou en simple application, un graphe montrant l'évolution des récompenses au fil des épisodes s'affiche à la fin

2.2

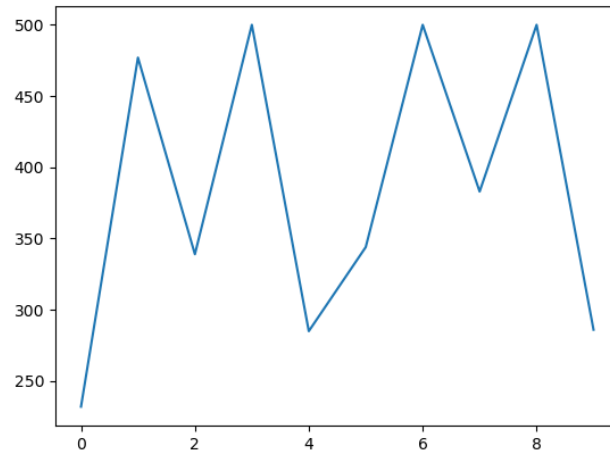
Le buffer implémenté, dans "Buffer.py", est un buffer générique capable de manipuler toute sortes d'états et d'insérer des éléments en temps constant ainsi que de récupérer des batch de données en environ $O(n)$ (n la taille du buffer).

2.3

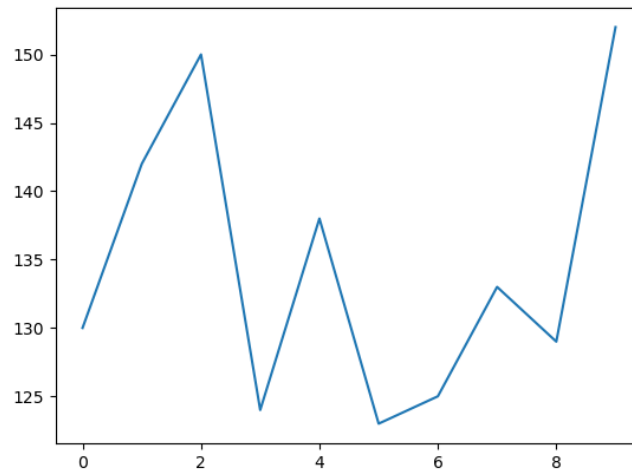
Le fichier "Network.py" contient un réseau de neurones simple composé de 2 couches cachées de tailles réduites L'agent explorateur, dans "Agent.py", applique une politique e-greedy avec in epsilon par défaut à 0,001.

2.4 résultats

Je n'ai malheureusement pas eu le temps de faire une recherche approfondie sur les hyperparamètres les plus adaptés, mais j'ai réussi à obtenir quelque résultats intéressants:



- le modèle "model_1.pt" est le plus performant obtenu et obtient généralement des scores supérieurs à 300 et atteint le score maximum de 500 environ 40% du temps.



- les modèles "model_2.pt" et "model_3.pt" montrent un comportement qui revient souvent quand l'apprentissage échoue: les parties commencent avec le baton partant d'un coté et l'agent n'arrive jamais à le redresser et se contente de retarder sa défaite.

3 Breakout

Je n'ai malheureusement pas réussi à faire fonctionner l'apprentissage sur le jeu "breakout" dans le temps imparti