Reinforcement Learning TP3
theophile.stourbe

QLearning Agent

Les hyper-paramètres ont été changé pour améliorer l'entraînement de l'agent. Le learning rate est resté à 0.5. Si il est trop bas l'agent mets beaucoup trop de temps à apprendre. J'ai choisi epsilon = 0.1 car en le gardant trop élevé on perd de la précision une fois que l'agent a passé la phase d'exploration. Pour ce qui est de la valeur de gamma, j'ai choisi la valeur 0.6.

QLearning Epsilon scheduler

Pour cet agent, j'ai choisi d'implémenter un scheduler exponentiel, défini par la fonction:

$$(x-2)^{-2k} - 2^{-2k} \times (1-x)$$

avec k un entier >= 1. Dans mon cas j'ai choisi $k = 3$ et $0 <= x <= 1$, pour avoir une courbe qui descend rapidement aux abords de 1, entre 0 et 1.

Les mêmes hyper-paramètres que le premier agent ont été choisi.

Sarsa

Sarsa est similaire au QLearning agent sauf qu'il n'y a pas de notion d'aléatoire dans sa décision, il prend toujours la direction qu'il pense être la meilleure.

Il utilise aussi les mêmes hyper-paramètres que les deux autres agents, sauf qu'il n'a pas de paramètres epsilon.

Training

Pour le training j'avais d'abord opter pour une implémentation ou l'agent recommencait au sein de la même boucle l'entraînement une fois qu'il avait finit (ou dépassé les 200 actions). J'ai changé d'avis car j'avais de problèmes avec le QLearning agent avec epsilon scheduler étant donné qu'il commencait avec une valeur d'epsilon à 1 et qu'il effectuait le cycle en entier dans la fonction *play_and_train*. De ce fait, je sors de la fonction lorsque l'agent réussi l'exercice ou dépasse le nombre max d'action. Cela accelère grandement l'entraînement.