Travail effectué sur le sujet: Prioritized memory access explains planning and hippocampal replay

Enzo DURAND, Francesco SABATINO
11 janvier 2023

Abstract

Notre projet porte sur l'étude du papier de Mattar et Daw, proposant une modélisation du fonctionnement des cellules place dans le cerveau des mammifères pour améliorer l'apprentissage d'une tâche de navigation spatiale en intégrant la notion d'expérience, de planification et de prédiction. Après avoir compris le sujet, nous avons dans un premier temps reproduit les résultats du papier pour les comprendre, puis mis en évidence certaines limites des modèles proposés.

1 Compréhension du sujet

Une partie importante a été la compréhension du sujet qui est complexe. On se place dans le cas d'une tâche de navigation spatiale. On cherche à optimiser l'apprentissage de l'environnement pour planifier un chemin en se basant sur le fonctionnement des cellules places de l'hippocampe, dans le cerveau d'un mammifère. Le principe est de construire une théorie pour prédire, au cours de la navigation, s'il est préférable d'exploiter une expérience passée, ou s'il convient de tenter de prédire les conséquences d'une action dans le futur.

Nous avons étudié le phénomène d' **hippocampal replay**, qui correspond à la simulation de la navigation par le cerveau, ainsi que la notion de **need**, représentation de l'utilité de prévoir des états futurs, ainsi que de **gain**, représentant l'utilité de ré-exploiter les informations récupérées dans des états précédents, dans le cas notamment d'une évolution qui ne s'est pas déroulée comme prévu (notion de surprise, cad. erreur de prédiction des conséquences d'une action). Nous avons pu remarquer la pertinence d'essayer de construire des expériences pour comprendre en quoi l'expression du gain par rapport au need permettait de mettre en évidence des phases d'apprentissage de l'environnement, de planification de chemin ou encore de consolidation des connaissances. Nous avons également pu remarquer l'intérêt de mettre en évidence des cas où ces étapes ne se comportent pas comme prévu et engendrent des phénomènes de stagnation qui ralentissent l'apprentissage de l'environnement.

2 Validation expérimentale

Nous avons utilisé l'implémentation Python des algorithmes de Mattar et Daw proposée par Thomas MISIEK. Ici, une souris étant placée dans un labyrinthe, elle doit apprendre l'environnement et trouver le fromage afin de trouver un chemin optimisé au cours du temps pour s'y rendre plus rapidement. Au cours de l'apprentissage, nous avons déplacé le fromage pour vérifier que la procédure s'adapte au changement. Pour évaluer la performance des procédures, nous les avons comparées entre elles sur des labyrinthes similaires, en y ajoutant une politique sans replay pour vérifier l'amélioration de la performance de la prise de décision avec les méthodes proposées dans l'article. Pour s'assurer de la consistance des résultats, nous répétons chaque expérience 10 fois.
Il est possible de reproduire les expériences en suivant les instructions du fichier README.md. Les expériences que nous avons effectuées se situent dans le dossier checkpoints/res, pour éviter qu'une nouvelle expérience efface les données déjà recueillies.
Nous avons pu remarquer avec le labyrinthe de base **mattar** reprenant l'exemple du papier, que les performances des algorithmes sont bien celles attendues: les approches basées sur le modèle des cellules places s'adapte très rapidement à leur environnement, alors que l'approche sans replay met plus de temps. On retrouve cette différence de performances lorsqu'on change la fonction de reward, les approches optimisées sont moins impactées par le changement et s'adaptent plus rapidement. On remarque toutefois que l'approche qui se base sur le gain uniquement est assez lente, probablement car cela correspond à ne se baser que sur l'expérience d'exploration déjà effectuée sans aucune prédiction dans le futur, ce qui ralentit les possibilités d'évolution. Nous pensons que l'absence de need implique donc une replanification systématique à l'aide du gain, d'où le temps plus élevé par épisode. On retrouve cela dans une moindre mesure avec l'approche DYNA, qui n'active cette replanification que de manière aléatoire donc moins souvent.

On constate, au niveau de la planification, que celle-ci intervient de façon importante dans un environnement déjà connu ayant évolué. En effet notre expérience montre l'augmentation du terme de planification dans l'épisode suivant le changement du reward: l'agent connaît déjà son environnement, c'est donc avantageux pour lui de replanifier pour aller chercher la nouvelle position du reward.
Nous avons également testé l'amélioration de la performance sur une instance différente **spec_maze**: un labyrinthe basique avec un seul chemin possible reliant deux endroits de ce labyrinthe. Nous plaçons la souris à un des deux endroits, et le reward dans le chemin reliant les deux endroits; puis nous changeons le reward pour le placer au deuxième endroit. Cela a largement dégradé la performance de la méthode sans replay qui a pris beaucoup de temps pour s'adapter. Nous avons pu confirmer l'utilité d'utiliser l'expérience précédence en remarquant que ce phénomène s'est activé au moment du changement de reward, pour les méthodes concernées.

3 Limites de l'approche

Nous avons pu valider l'amélioration engendrée par le replay, mais on peut trouver des cas où cela portera préjudice à la décision. Nous avons pour cela imaginé une instance **custom** construite de façon à proposer plusieurs chemins différents et longs pour atteindre un premier reward, et un seul chemin très court pour atteindre le deuxième, mais dans le sens opposé.
Cela a dégradé la performance des algorithmes basés sur le retour d'expérience qui ont probablement tenté de se rabattre sur des chemins secondaires prometteurs, sans considérer un chemin totalement différent mais bien plus court. L'approche basée sur le need uniquement a été fortement impactée par cela, la faisant quasiment régresser au niveau de l'approche sans replay. Les autres se sont adaptées plus rapidement, mais au terme de nombreuses étapes de replanification. Le nombre de chemins envisageables (notamment ceux qui n'occasionnent pas de cycle) fait que les phases de planification ont souvent été longues sur la plupart des méthodes.
Nous avons également pu tester les différences de certaines approches sur un labyrinthe composée d'une unique ligne droite avec une position initiale et un reward invariants. On a vu que les méthodes basées sur l'exploitation de l'expérience passée ont sous-performé, étant plus longues que la méthode sans replay, alors que les autres ont gardé une performance meilleure. Ces temps supérieurs s'expliquent par la planification continue qui a eu lieu, contrairement aux autres approches qui n'ont pas recalculé leurs fonctions d'utilité.