Cahier des Charges

P-ANDROIDE - Projet ANDROIDE

Étude de l'apprentissage par renforcement profond sur Pendulum

Master d'informatique spécialité androide

Première année

Année universitaire 2020 - 2021

Professeur : Olivier Sigaud
Étudiants : Vincent Fu
Yuhao Liu

1. Description générale

1.1 Contexte

Les algorithmes d'apprentissage par renforcement représentent des techniques importantes en Intelligence Artificielle. Ils sont notamment les raisons pour lesquelles il est possible pour un agent autonome (logiciel, robot, etc ...) d'apprendre par exemple à jouer un jeu vidéo, effectuer une reconnaissance faciale, ou encore ramasser des balles de tennis.

Pour évaluer les performances des algorithmes d'apprentissage par renforcement, il est courant de les appliquer en simulation à des problèmes de contrôle de mouvements complexes de grande dimension tels que les environnements HalfCheetah, Walker ou encore Humanoid en raison de la difficulté d'apprentissage. De ce fait, l'étude des simulations de problèmes de petite dimension comme les environnements CartPole, Pendulum ou MountainCar est parfois mise à l'écart lors d'importants travaux de recherche alors qu'il y a encore beaucoup à apprendre sur ces types de problèmes.

1.2 Objectif

L'objet de ce projet est donc d'en apprendre plus sur les problèmes de petite dimension et en particulier d'analyser les performances des algorithmes d'apprentissage par renforcement sur l'environnement Pendulum.

1.3 État de l'art et mise en pratique

Jusqu'à présent, des travaux ont montré que les méthodes de gradient sur les politiques (Policy Gradient) n'atteignent pas une performance suffisante sur l'environnement Pendulum, alors que les algorithmes d'apprentissage par renforcement classique comme *Deep Deterministic Policy Gradient* (DDPG), *Soft Actor Critic* (SAC) y parviennent. Il est donc intéressant d'expliquer quelles sont les raisons des performances des algorithmes d'apprentissage par renforcement sur Pendulum. Autrement dit, quels sont les composants, les outils utilisés dans ces algorithmes engendrant un gain de performance sur Pendulum et pour quelles raisons.

Pour ce faire, en pratique, il est nécessaire d'effectuer une étude par ablation (ablation study). L'étude par ablation consiste à étudier systématiquement chaque algorithme résultant des débranchements un à un des composants de l'algorithme initial.

2. Description fonctionnelle

2.1 Pré-requis et contraintes

1. Les notions abordées lors du projet n'étant pas à notre portée avec notre niveau actuel, il est nécessaire de suivre une formation initiale à l'apprentissage par renforcement profond. Regarder et revoir les cours de notre professeur est indispensable.
2. Le langage de programmation utilisé est Python.
   Cela implique donc d'installer les bibliothèques classiques de programmation pour l'apprentissage artificiel (Machine Learning) ainsi la bibliothèque gym regroupant tous les environnements classiques d'apprentissage par renforcement dont Pendulum.

3. Effectuer une étude sur les algorithmes d'apprentissage par renforcement nécessite d'avoir les programmes réalisant ces algorithmes.
   Il est donc impératif de trouver une bibliothèque regroupant cela et de l'installer. De plus, à titre indicatif, afin de faciliter le travail, sachant qu'on effectuera des modifications sur les programmes, il est intéressant de sélectionner une bibliothèque ayant une structure de programmation simple (pas de codes compliqués et de préférence n'entraînant pas de modification en chaîne importante à effectuer lors du modification d'une partie de code).
4. Afin d'observer les performances de manière précise, il est nécessaire d'avoir un programme permettant d'afficher les courbes de performances (en apprentissage par renforcement, ce sont les courbes de récompense moyenne en fonction du nombre de pas de temps d'apprentissage).
   Plusieurs manières d'obtenir le programme : soit en utilisant le programme intégré d'affichage de courbe du bibliothèque regroupant les algorithmes (en général, ces bibliothèques utilisent TensorBoard), soit en utilisant le programme de sauvegarde des politiques du bibliothèque et en programmant nous-même le programme d'affichage de courbe. Il faudra alors juste de charger les politiques apprises et les donner en argument à notre fonction d'affichage.

2.2 Instructions et mission

Voici la liste des étapes que nous devons suivre afin de mener à bien le projet :

Code	Descriptif
I01	Regarder les cours de notre professeur sur l'apprentissage par renforcement.
I02	Tester les méthodes de Policy Gradient sur Pendulum.
M01	Savoir évaluer les politiques en apprenant à charger les politiques apprises et à lancer la fonction d'évaluation.
I03	Choisir une bibliothèque parmi les bibliothèques regroupant les algorithmes d'apprentissage par renforcement.
M02	Identifier parmi les algorithmes classiques d'apprentissage par renforcement, l'algorithme qui est le plus susceptible d'obtenir des performances satisfaisantes sur Pendulum selon les indications du professeur.
I04	Évaluer les performances de l'algorithme issu du choix de M02 en suivant le raisonnement de l'instruction M01.
M03	Programmer une fonction d'affichage de courbe de récompense en vue d'observer de manière précise l'évolution des performances de l'algorithme issu du choix de M02.
I05	Afficher les courbes de récompense de l'algorithme de M02 et vérifier la cohérence des résultats (i.e. vérifier qu'on obtient une bonne performance sur Pendulum).
M04	Lister toutes les caractéristiques, les outils utilisés dans l'algorithme de M02.
I06	Supprimer une caractéristique parmi les caractéristiques listées dans M04 et effectuer l'instruction I05 en analysant les résultats.
I06 bis	Modifier une caractéristique parmi les caractéristiques listées dans M04 lorsque la caractéristique est non supprimable et effectuer l'instruction I05 en analysant les résultats.
I07	Poursuivre l'étude des performances en réglant les hyperparamètres dans le cas d'un mauvais résultat en I06, ou bien dans le cas d'un bon résultat en I06, continuer l'instruction I06 en supprimant cette fois-ci une caractéristique parmi les caractéristiques restantes.
MP	Effectuer une étude par ablation sur les caractéristiques listées dans M04 en suivant dans l'ordre les étapes : I06 (+ I05) ou I06 bis (+ I05), I07
MV	Lorsqu'une caractéristique critique au gain de performance est trouvée, vérifier la déduction en intégrant cette caractéristique dans un autre algorithme d'apprentissage par renforcement qui ne donne que des mauvaises performances sur Pendulum et en affichant les performances de l'algorithme résultant

2.3 Bilan attendu

À l'issue du projet, le bilan attendu est d'avoir trouver le ou les composants critiques amenant aux gains de performance pour l'environnement Pendulum et d'expliquer les raisons de ces performances par rapport à ces caractéristiques.

3. Livrables

3.1 Rendu attendu

À l'issue de l'UE, il est demandé de rendre un rapport du projet détaillant les travaux effectués pendant le projet avec en plus notre dossier incluant tous les programmes et ressources utiles ou utilisés pour le projet sous forme de git ainsi qu'une bibliographie (qui est intégrée au carnet de bord).

De plus, si possible, il est également demandé de rendre un fichier pdf expliquant l'utilisation des programmes utilisés comme par exemple un manuel d'utilisation.

3.2 Calendrier

— Les rendus du projet auront lieu le vendredi 21 mai.
— Les soutenances du projet auront lieu le jeudi 27 mai ou le vendredi 28 mai.