Université Sorbonne

Master Androide 1ère Année

P-Androide

Cahier des Charges

Auteurs :

Coline LACOUX
Ryan OHOUENS
Maël FRANCESCHETTI

Encadrant :

Olivier SIGAUD

Mercredi 19 Février 2020

Table des Matières

1 Présentation du Projet

1.1 Contexte et définition du problème

L'évaluation d'algorithmes d'apprentissage par renforcement profond à actions continues repose sur l'utilisation de benchmarks standards qui consistent souvent en la simulation d'un système physique plus ou moins complexe qu'il faut contrôler. Parmi ces nombreux benchmarks (Half-Cheetah, Ant, Humanoid, etc.), Swimmer est un cas particulier. La comparaison de nombreux algorithmes d'apprentissage par renforcement profond et méthodes évolutionnaires, voire des combinaisons des deux ont montré que, plus un algorithme fait appel à des méthodes d'apprentissage par renforcement profond sophistiquées, moins il est performant sur Swimmer. A l'inverse, les méthodes évolutionnaires se comportent bien dans cet environnement.

1.2 Environnement

Pour ce projet, nous utiliserons l'environnement Swimmer-v2 distribué par OpenAI : https://gym.openai.com/envs/Swimmer-v2/.

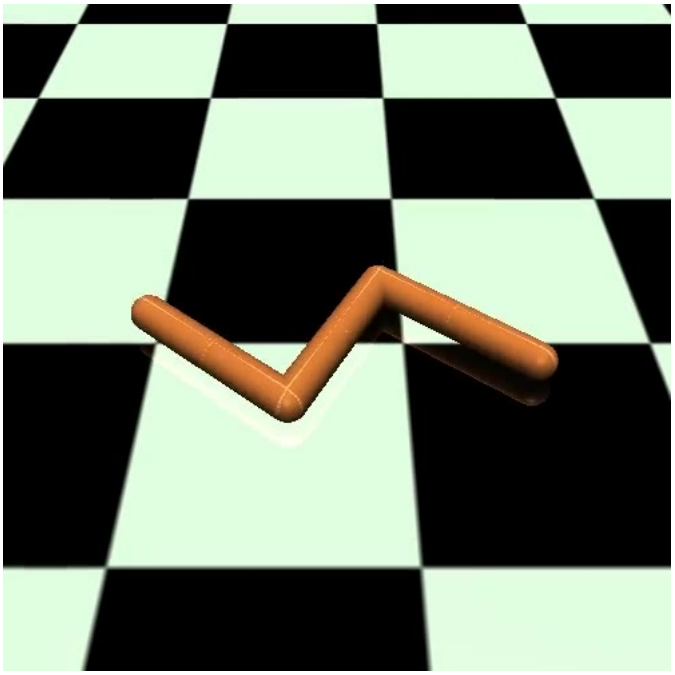

FIGURE 1 – Environnement Swimmer

Cet environnement est un benchmark utilisant le simulateur physique 3D Mujoco : http://www.mujoco.org/.

Le Swimmer est composé de trois flotteurs reliés par deux articulations motorisées. L'espace autour du swimmer est plan et visqueux. Le Swimmer peut se déplacer en effectuant une action, définie par une valeur d'impulsion réelle comprise dans $[-1, 1]$ pour chaque moteur. Un état de l'environnement est défini par huit éléments : la position sur l'axe X de l'élément central du Swimmeur (milieu du flotteur central), sa position Y, la vélocité sur l'axe X, la vélocité sur l'axe Y, l'angle de rotation de chaque articulation et la vélocité en rotation sur chaque articulation. Le score (récompense) obtenu est calculé à chaque déplacement, selon la formule suivante :

$$score = \frac{oldX - newX}{nbSteps} - costCoeff \times \sum_{i=0}^{nbActions} action[i]^2$$

avec $costCoeff$ fixé à $0,0001$. Autrement dit, la récompense est la vitesse en X moins le coût du déplacement :

$$score = v - cost \text{ , avec } cost = costCoeff \times \sum_{i=0}^{nbActions} action[i]^2$$

et avec v la vélocité de déplacement sur l'axe X.
Le score maximum atteignable en un épisode sur cet environnement est d'environ 360.

1.3 Littérature

Les algorithmes d'apprentissage par renforcement permettent d'obtenir de meilleurs résultats que les algorithmes évolutionnaires sur la majorité des environnements mujoco. Il a été envisagé de combiner les algorithmes évolutionnaires et de RA, avec l'algorithme CEM-TD3 le résultat est encore meilleur que les algorithmes de RA sur ces environnements. L'environnement Swimmer réagis cependant très différemment, comme on peut l'observer sur la Figure 2 qui montre les résultats obtenus sur quelques environnements mujoco avec ces différents algorithmes.

	CEM			TD3		
Environment	Mean	Var.	Median	Mean	Var.	Median
HALF-CHEETAH-V2	2940	12%	3045	9630	2.1%	9606
HOPPER-V2	1055	1.3%	1040	3355	5.1%	3626
WALKER2D-V2	928	5.4%	934	3808	8.9%	3882
SWIMMER-V2	**351**	**2.7%**	**361**	63	14%	47
ANT-V2	487	6.7%	506	4027	10%	4587

	TD3 Multi-Actor			CEM-TD3		
Environment	Mean	Var.	Median	Mean	Var.	Median
HALF-CHEETAH-V2	9662	2.8%	9710	**10725**	**3.7%**	**11539**
HOPPER-V2	2056	20%	2376	**3613**	**2.9%**	**3722**
WALKER2D-V2	3934	4.1%	3954	**4711**	**3.3%**	**4637**
SWIMMER-V2	76	14%	60	75	15%	62
ANT-V2	3567	22%	3911	**4251**	**5.9%**	**4310**

FIGURE 2 – Source : https://arxiv.org/pdf/1810.01222.pdf

1.4 Objectifs

L'objet de ce projet est de comprendre en détail les mécanismes qui expliquent les résultats obtenus sur l'environnement Swimmer. On soupçonne un phénomène de "deceptive gradient" : suivre le gradient de la performance comme le font les algorithmes d'apprentissage par renforcement profond conduit à détériorer le contrôleur plutôt qu'à l'améliorer.

Le projet supposera dans un premier temps de mettre en oeuvre des algorithmes d'apprentissage par renforcement profond et évolutionnaires dans l'environnement Swimmer sous l'interface standard OpenAI gym, puis de produire des outils permettant d'analyser en profondeur les résultats pour les interpréter de façon indiscutable.

2 Solutions étudiées

2.1 Etude des minima locaux

L'hypothèse de la présence de minimas locaux bloquants pour les algorithmes d'apprentissage par renforcement est la première piste étudiée. En effet, on observe que sur l'environnement Swimmer l'algorithme d'apprentissage par renforcement TD3 dépasse difficilement un cap de score situé aux alentours de 60, alors que l'algorithme évolutionnaire CEM atteint sans difficulté des scores supérieures à 300. Si cette piste est avérée, il sera de vigueur de comprendre pourquoi ces algorithmes n'arrivent pas à sortir de ces minimas contrairement aux algorithmes évolutionnaires.

On travaillera donc à mettre en place des outils d'étude des minimas locaux, notamment l'étude de la géométrie de la fonction objectif par perturbation aléatoire (https://arxiv.org/pdf/1811.11214.pdf) et la visualisation de donnée en grande dimension avec T-SNE (http://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf).

2.2 Comportement des algorithmes

On s'intéressera également à entraîner des acteurs tout d'abord avec un algorithme d'apprentissage par renforcement TD3 et à ensuite les entraîner sur un algorithme évolutionnaire CEM pour "débloquer" l'apprentissage, et essayer de comprendre l'origine du blocage. Pour celà, on aura besoin de pouvoir entraîner des acteurs sur les différents algorithmes et de sauvegarder/charger leur modèle à postériori.

De façon similaire, on pourra avec CEM entraîner un acteur et ajouter l'entraînement d'un critique, et ensuite continuer l'entraînement avec TD3 : on verra alors si, même avec un critique entraîné, TD3 fait chuter la fitness des acteurs ou non, et s'il l'améliore ou non.

Nous étudierons aussi les causes possibles de cette défaillance des algorithmes d'apprentissage par renforcement en essayant de modifier l'environnement et en étudiant l'impact de ces changements sur les résultats des algorithmes, entre autre : retirer la viscosité du milieu, retirer la position du Swimmer de l'état de l'environnement, retirer le coût de déplacement, modifier la structure du Swimmer.

2.3 Evolution des comportements du Swimmer

Afin de mieux comparer les étapes majeures de progression du comportement du Swimmer lors de l'apprentissage, on réalisera des "zoos" présentant tous les types de comportements observés lors de nombreux apprentissages sur les différents algorithmes. On effectuera un zoo pour les algorithmes évolutionnaires et un second zoo pour les algorithmes d'apprentissage par renforcement, afin de pouvoir les confronter. On permettra de récupérer les paramètres de l'acteur correspondant à chaque comportement présenté : celà nous sera utile pour vérifier la présence de minimas/maximas locaux.

On réalisera également des frises d'évolution des comportement lors de l'apprentissage sur ces deux types d'algorithmes, afin d'éventuellement mettre en avant une bifurquation de direction dans l'évolution des comportements lors de l'apprentissage. Si jamais nous observons des retours en arrière ou des inter-croisements entre différents comportements, on pourra alors représenter l'évolution sous forme de chaîne de Markov plutôt que sur une frise. Faire des statistiques sur la fréquence de chaque comportement observé serait également un bon outil.

2.4 Gradient déceptif

S'il advient qu'il ne s'agit pas d'un problème de minimas locaux blocants pour les algorithmes d'apprentissage par renforcement, il faudra étudier la piste du gradient déceptif et en comprendre les mécanismes.

3 Ressources

4 Planning prévisionnel