Projet deep reinforcement learning : évolution des agents par renforcement, vers des espaces d'action hybrides

Mots-clés: DRL, IA, PPO, H-PPO

Proposé par :

- Romain Dulout Université Gustave Eiffel (COSYS-ERENA) (romain.dulout@univ-eiffel.fr)

1. Contexte

L'intégration croissante de l'intelligence artificielle (IA) dans de nombreuses branches technologiques a non seulement permis la résolution de problèmes complexes, mais a également profondément métamorphosé ces domaines, ouvrant les portes à des avancées révolutionnaires. Par exemple, l'IA est aujourd'hui utilisée dans le domaine médical pour analyser de nombreuses données complexes de sorte à détecter de manière précoce les cancers [1]. Dans le secteur automobile, elle propulse l'avènement des véhicules autonomes et des systèmes de transports intelligents coopératifs (C-ITS), transformant ainsi la mobilité urbaine vers des standards axés sur la sécurité de l'usager et l'écologie [2]. Elle a également contribué à lever de nombreuses menaces dans le domaine de la cybersécurité, telles que les attaques par déni de service distribuées (DDOS) en accélérant l'identification des menaces potentielles, et en renforçant la détection des comportements suspects [3]. Enfin, l'emploi massif de l'IA dans le domaine de la finance a rendu possible l'analyse rapide de vastes ensembles de données financières, offrant des perspectives précieuses pour prendre des décisions éclairées et anticiper les mouvements du marché [4].

L'IA adopte diverses approches pour répondre aux besoins spécifiques de chaque domaine technologique. Au cœur de ces paradigmes se trouve le machine learning (ML), offrant aux machines la capacité d'apprendre à partir de données et de générer des prédictions ou des actions sans programmation explicite [5]. Lorsque les données sont étiquetées, l'apprentissage supervisé offre des modèles précis pour les prédictions ou classifications (comme la régression linéaire/logistique, Naive Bayes, Random Forest, SVM, K-NN). À l'inverse, l'apprentissage non supervisé, devient un outil puissant pour explorer, organiser et comprendre les données sans nécessiter d'étiquetage manuel (tels que k-moyennes, regroupement hiérarchique, estimation par noyau). Enfin, le deep learning, une sous-catégorie du ML, exploite des réseaux de neurones profonds pour traiter des données complexes, dégageant automatiquement des caractéristiques pertinentes et produisant des modèles de haute précision pour la prédiction et la classification [6].

Cependant, l'usage exclusif de ces algorithmes montre ses limites pour résoudre l'ensemble des défis propres aux différents domaines scientifiques. En effet, disposer préalablement d'un jeu de données pour entraîner les modèles s'avère souvent irréalisable. C'est ainsi que s'est développée la catégorie des algorithmes d'apprentissage par renforcement, autorisant une IA (agent) à résoudre des problèmes séquentiels sans nécessiter de modèle préétabli [7]. Ainsi, un agent interagit activement avec son environnement en prenant des actions pour atteindre des objectifs définis, et se voit attribuer des récompenses ou des sanctions selon la qualité de ses actions. L'apprentissage se fait de manière progressive : l'agent explore diverses actions, ajuste sa stratégie (policy) en fonction des retours de l'environnement (states) et des récompenses reçues, afin d'améliorer ses performances au fil de ses interactions.

Aujourd'hui, l'intégration du deep learning aux méthodes d'apprentissage par renforcement a ouvert des perspectives inédites et hautement performantes dans ce domaine : il s'agit du deep

reinforcement learning (DRL). Pour des choix d'actions discrets, tels que la direction à prendre pour un véhicule autonome (par exemple, tourner à gauche ou à droite), des algorithmes tels que le DQN (Deep Q Network) offrent stabilité et performances. En revanche, pour des actions continues, comme déterminer un angle de rotation pour un véhicule (par exemple, entre 0° et 180°), des méthodes comme le DDPG (Deep Deterministic Policy Gradient), le TD3 (Twin Delayed DDPG), ou le SAC (Soft Actor-Critic) s'avèrent plus adaptées [8]. Ces dernières reposent sur des architectures actor-critic, où un réseau de neurone"actor" définit les actions continues à prendre, pendant qu'un réseau de neurone "critic" les évalue, optimisant ainsi la prise de décision dans des environnements complexes.

Ces derniers temps, l'algorithme PPO (Proximal Policy Optimization) [9] s'est démarqué parmi l'ensemble des solutions de DRL pour la gestion des espaces d'actions continus. Il est une évolution du TRPO, héritant de tous ses avantages avec une mise en place simplifiée et une moindre complexité algorithmique, tout en surpassant les autres solutions en termes de performances dans de nombreuses configurations. Ainsi, nous nous concentrerons dans la suite de ce projet sur la compréhension et la mise en place de cet algorithme. Cependant, un problème persiste (comme dans toutes ces solutions) : la gestion des espaces d'actions hybrides qui mêlent espaces d'actions continus et discrets. Par exemple, dans le cas du véhicule autonome, il serait intéressant que l'agent puisse prendre une action discrète (choisir la direction) et simultanément une action continue (déterminer l'angle de rotation dans cette direction). Toutefois, les structures des algorithmes existants ne conviennent pas à résoudre ces problèmes spécifiques. Pour relever ce défi, certains chercheurs suggèrent une amélioration des architectures en introduisant des espaces d'actions paramétrés, où un groupe d'actions discrètes est défini par un ensemble de paramètres continus [10].

2. Objectifs

Étape 1 :

Dans ce contexte, vous devrez, dans un premier temps, étudier le fonctionnement du PPO afin de l'implémenter. Pour cela, vous avez à disposition de nombreuses ressources en ligne, telles que des articles scientifiques, des cours en ligne ou encore ce tutoriel [11] – [14]. Pour cette première partie, vous êtes libre du choix de l'environnement avec lequel votre agent va interagir (vous devez simplement vous assurer que l'espace d'action soit continu). Si vous suivez le tutoriel [11] – [14], vous aurez à gérer le problème du pendule [15], dans lequel un pendule est attaché à une extrémité à un point fixe (l'autre extrémité étant libre), et dont l'objectif est d'appliquer un couple sur l'extrémité libre pour le faire basculer dans une position verticale (et la maintenir).

Étape 2 :

Une fois le PPO déployé et fonctionnel, en vous basant sur les travaux de [16], vous ferez évoluer l'architecture du PPO vers celle du H-PPO (hybrid proximal policy optimization) afin qu'il soit capable de résoudre la prise de décision séquentielle dans un espace d'actions hybrides (continues et discrètes). Pour cela, vous déploierez votre H-PPO dans l'environnement : « Robot Soccer Goal » [17] dans lequel l'agent doit apprendre à marquer un but en présence d'un gardien. Trois actions sont disponibles pour l'agent :

- Avancer avec la balle à (x,y)
- Marquer un but à gauche (h distance)
- Marquer un but à droite (h distance)

3. **Modalité d'évaluation**

- Soutenance et démonstration : 29 janvier 2024 (groupe B)
- Rapport + code
- Durée 20 minutes / groupe

Références

[1] T. Saba, 'Recent advancement in cancer detection using machine learning: Systematic survey of decades, comparisons and challenges', *Journal of Infection and Public Health*, vol. 13, no. 9, pp. 1274–1289, Sep. 2020, doi: 10.1016/j.jiph.2020.06.033.

[2] G. Bathla *et al.*, 'Autonomous Vehicles and Intelligent Automation: Applications, Challenges, and Opportunities', *Mobile Information Systems*, vol. 2022, p. e7632892, Jun. 2022, doi: 10.1155/2022/7632892.

[3] X. Yuan, C. Li, and X. Li, 'DeepDefense: Identifying DDoS Attack via Deep Learning', in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, May 2017, pp. 1–8. doi: 10.1109/SMARTCOMP.2017.7946998.

[4] E. A. Gerlein, M. McGinnity, A. Belatreche, and S. Coleman, 'Evaluating machine learning classification for financial trading: An empirical approach', *Expert Systems with Applications*, vol. 54, pp. 193–207, Jul. 2016, doi: 10.1016/j.eswa.2016.01.018.

[5] N. Burkart and M. F. Huber, 'A Survey on the Explainability of Supervised Machine Learning', *Journal of Artificial Intelligence Research*, vol. 70, pp. 245–317, Jan. 2021, doi: 10.1613/jair.1.12228.

[6] S. Pouyanfar *et al.*, 'A Survey on Deep Learning: Algorithms, Techniques, and Applications', *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–36, Sep. 2019, doi: 10.1145/3234150.

[7] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, 'A Brief Survey of Deep Reinforcement Learning', *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017, doi: 10.1109/MSP.2017.2743240.

[8] T. Haarnoja *et al.*, 'Soft Actor-Critic Algorithms and Applications'. arXiv, Jan. 29, 2019. Accessed: Dec. 06, 2023. [Online]. Available: http://arxiv.org/abs/1812.05905

[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, 'Proximal Policy Optimization Algorithms'. arXiv, Aug. 28, 2017. Accessed: Dec. 06, 2023. [Online]. Available: http://arxiv.org/abs/1707.06347

[10] W. Masson, P. Ranchod, and G. Konidaris, 'Reinforcement Learning with Parameterized Actions'. arXiv, Nov. 26, 2015. Accessed: Dec. 06, 2023. [Online]. Available: http://arxiv.org/abs/1509.01644

[11] E. Y. Yu, 'Coding PPO from Scratch with PyTorch (Part 1/4)', Analytics Vidhya. Accessed: Dec. 05, 2023. [Online]. Available:

https://medium.com/analytics-vidhya/coding-ppo-from-scratch-with-pytorch-part-1-4-613dfc1b14c8

[12] E. Y. Yu, 'Coding PPO From Scratch With PyTorch (Part 2/4)', Medium. Accessed: Dec. 05, 2023. [Online]. Available: https://medium.com/@eyyu/coding-ppo-from-scratch-with-pytorch-part-2-4-f9d8b8aa938a

[13] E. Y. Yu, 'Coding PPO from Scratch with PyTorch (Part 3/4)', Analytics Vidhya. Accessed: Dec. 05, 2023. [Online]. Available: https://medium.com/analytics-vidhya/coding-ppo-from-scratch-with-pytorch-part-3-4-82081ea58146

[14] Z. Xia, 'Coding PPO from Scratch with PyTorch (Part 4/4)', Medium. Accessed: Dec. 06, 2023. [Online]. Available: https://medium.com/@z4xia/coding-ppo-from-scratch-with-pytorch-part-4-4-4e21f4a63e5c

[15] 'Pendulum - Gym Documentation'. Accessed: Dec. 06, 2023. [Online]. Available: https://www.gymlibrary.dev/environments/classic_control/pendulum/

[16] Z. Fan, R. Su, W. Zhang, and Y. Yu, 'Hybrid Actor-Critic Reinforcement Learning in Parameterized Action Space'. arXiv, May 30, 2019. Accessed: Dec. 06, 2023. [Online]. Available: http://arxiv.org/abs/1903.01344

[17] 'GitHub - cycraig/gym-goal: OpenAI Gym environment for Robot Soccer Goal'. Accessed: Dec. 06, 2023. [Online]. Available: https://github.com/cycraig/gym-goal