Étude Actor-Free critic Updates

Projet AI2D – encadré par M. Olivier Sigaud

15 mai 2025

Paul Chambaz & Frédéric Li Combeau

1 Apprentissage par renforcement

Présentation

On modélise le problème comme un problème de décision Markovien.

Problème de décision Markovien

Formalisation en tuple : $< S, A, T, R, \gamma >$

La Q-fonction représente l'espérance des récompenses cumulées pour un couple d'état action.

Q-fonction optimale

$Q^*(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s, a_0 = a, \pi^*\right]$

Différence temporelle

$L_Q(\psi) = \mathbb{E}_{(s,a,r,s')\sim B}\left[\left(Q_\psi(s,a) - r - \gamma V_\varphi(s')\right)^2\right]$

DDPG

Introduction architecture Acteur Critique.

Acteur

Politique proposant une action dans un état donné – $\pi : S \to A$

Critique

Q-network qui évalue la qualité d'un couple état action – $Q : S \times A \to \mathbb{R}$

SAC

Sortie gaussienne de l'acteur pour conserver l'action jugée optimale μ et le degré de certitude sur cette action σ^2. Maximisation de l'entropie.

Présentation de AFU

En apprentissage par renforcement:

- Problème central: estimer $\max_a Q(s, a)$ dans des espaces d'actions continus
- Méthodes traditionnelles (DDPG, SAC): utilisent l'acteur pour estimer indirectement ce maximum

Solution AFU:

- Décomposition: $Q(s, a) = V(s) + A(s, a)$ où:
- Mécanisme de pression adaptative vers le bas :
  - Indicateur $I(s, a) = 1$ si $V(s) + A(s, a) < Q(s, a)$, sinon 0
  - Paramètre $\rho \in [0, 1]$ contrôle l'intensité de la pression
  - Modifie le gradient: réduit partiellement lorsque $V(s)$ sous-estime
  - Crée une asymétrie qui force $V(s) \to \max_a Q(s, a)$

Architecture: 6 réseaux neuronaux (Q, $2 \times V$, $2 \times A$, π) avec paramètres ψ, ϕ, ξ, θ

Loss de AFU

$$L_Q(\psi) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{B}}\left[\left(Q_\psi(s,a) - r - \gamma \min_{i\in\{1,2\}} V_{\varphi_{i'}}(s')\right)^2\right]$$

$$L_{\mathrm{VA}}(\varphi_i,\xi_i) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{B}}\left[Z\left(\Upsilon_i^a(s) - r - \gamma \min_{i\in\{1,2\}} V_{\varphi_{i'}}(s'), A_{\xi_i}(s,a)\right)\right]$$

$$I_i(s,a) = \begin{cases} 1 & \text{if } V_{\varphi_i}(s) + A_{\xi_i}(s,a) < Q_\psi(s,a) \\ 0 & \text{otherwise.} \end{cases}$$

$$\Upsilon_i^a(s) = (1 - \rho.I_i(s,a))V_{\varphi_i}(s) + \rho.I_i(s,a).V_{\varphi_i}^{\mathrm{nograd}}(s)$$

$$Z(x,y) = \begin{cases} (x+y)^2 & \text{if } x \leq 0 \\ x^2 + y^2 & \text{otherwise.} \end{cases}$$

$$L_\pi(\theta) = \mathbb{E}_{s\sim\mathcal{B}}\left[\alpha \log(\pi_\theta(a_s \mid s)) - Q_\psi(s,a_s)\right].$$

2 Apprentissage off-policy

Apprentissage off-policy

Séparation de la politique de collecte des données de la politique optimisée, permettant d'apprendre une stratégie tout en en suivant une autre.

Cette méthode représente un défi majeur pour les algorithmes qui utilisent l'acteur pour proposer des actions qui maximise $Q(s, a)$.

AFU résout ce problème par le mécanisme de mise à jour conditionnelle et agit de façon plus similaire à l'algorithme Q-learning, qui est off-policy.

En *true off-policy*, on sélectionne des couples état-action par un tirage uniforme dans l'espace d'état-action.

Expérience

Hypothèse

AFU converge en *true off-policy* sur des environnements où des algorithmes classiques (DDPG et SAC) échouent.

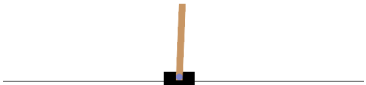
Méthodologie :

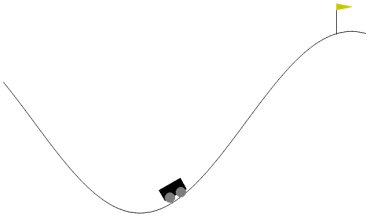
Entraînement sur *Cartpole*, *MountainCar*, *Pendulum* et *LunarLander* pendant $200k$ pas de temps à partir d'une politique de collecte de données aléatoire.

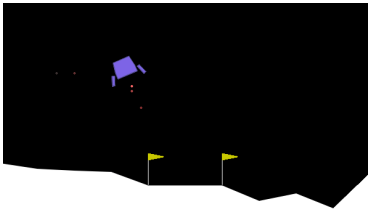
- Évaluation tous les 100 pas de temps sur 10 épisodes.
- On réalise 5 fois l'expérience pour des résultats plus significatifs.
- Ajout de `_set_state` à *Gymnasium* pour fixer l'état de l'environnement.

Si notre hypothèse est correcte, alors DDPG et SAC ne devraient pas converger et AFU devrait converger vers la politique optimale.

Expérience (ii)







Résultats

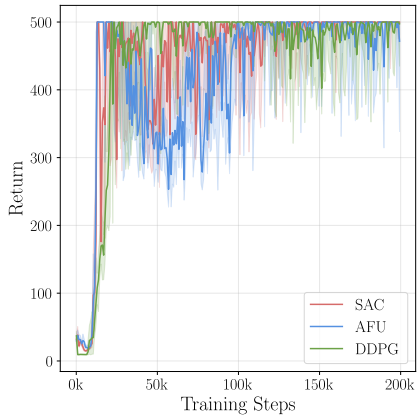


Fig. 1. – Évaluation performance sur Cartpole

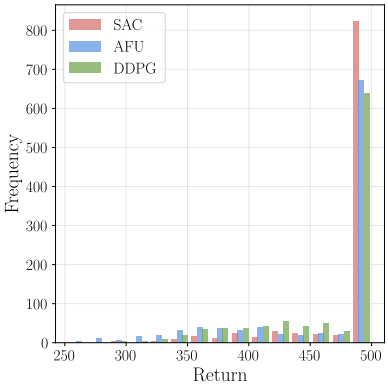


Fig. 2. – Histogramme Cartpole dernier %

Résultats (ii)

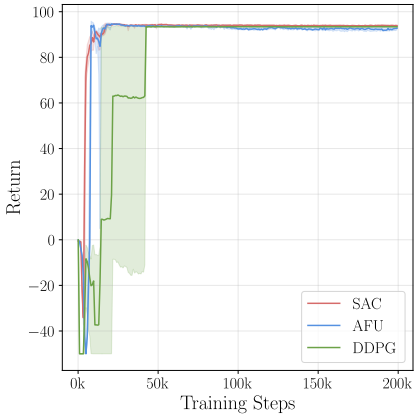


Fig. 3. – Évaluation performance sur MountainCar

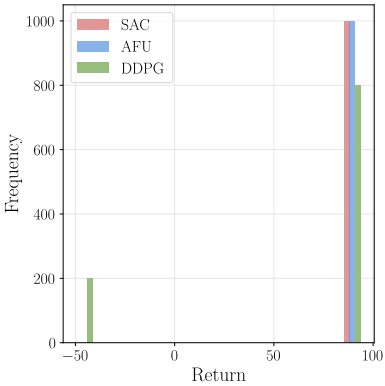


Fig. 4. – Histogramme MountainCar dernier %

Résultats (iii)

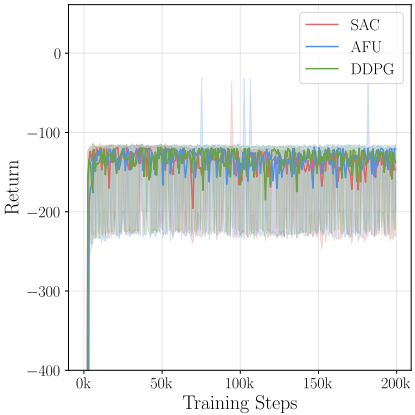


Fig. 5. – Évaluation performance sur Pendulum

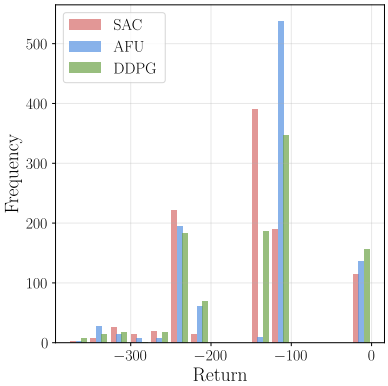


Fig. 6. – Histogramme Pendulum dernier %

Résultats (iv)

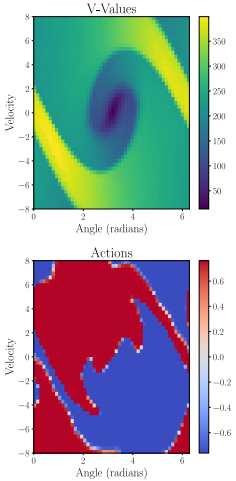


Fig. 7. – V et π de DDPG sur pendulum

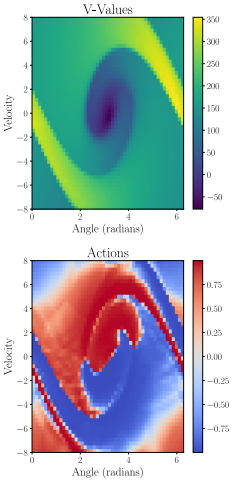


Fig. 8. – V et π de SAC sur pendulum

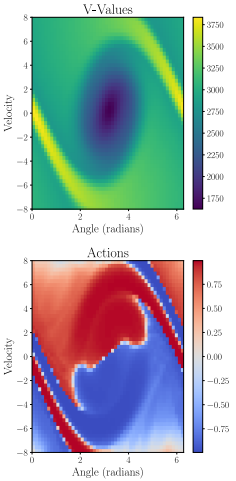


Fig. 9. – V et π de AFU sur pendulum

Interprétation

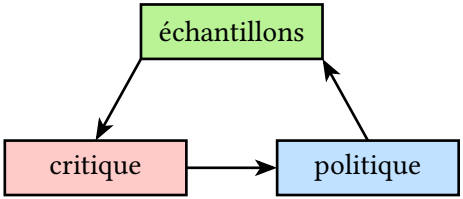


Fig. 10. – Critique pur

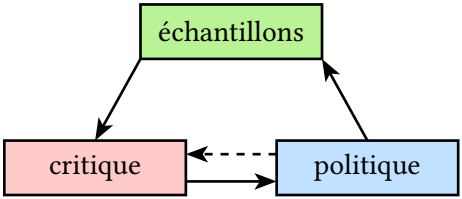


Fig. 11. – Acteur critique

Note : schéma issu du cours de M. Olivier Sigaud

Résultats Lunar Lander

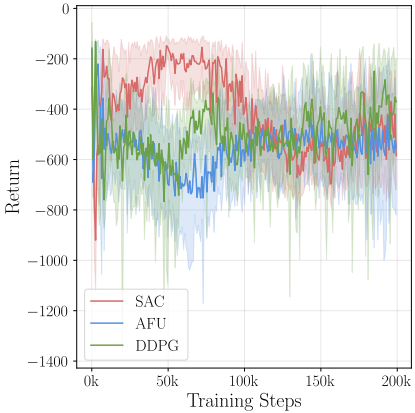


Fig. 12. – Évaluation performance sur LunarLander

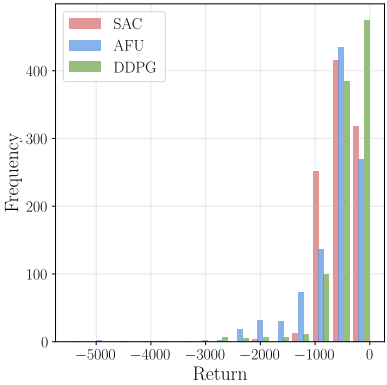
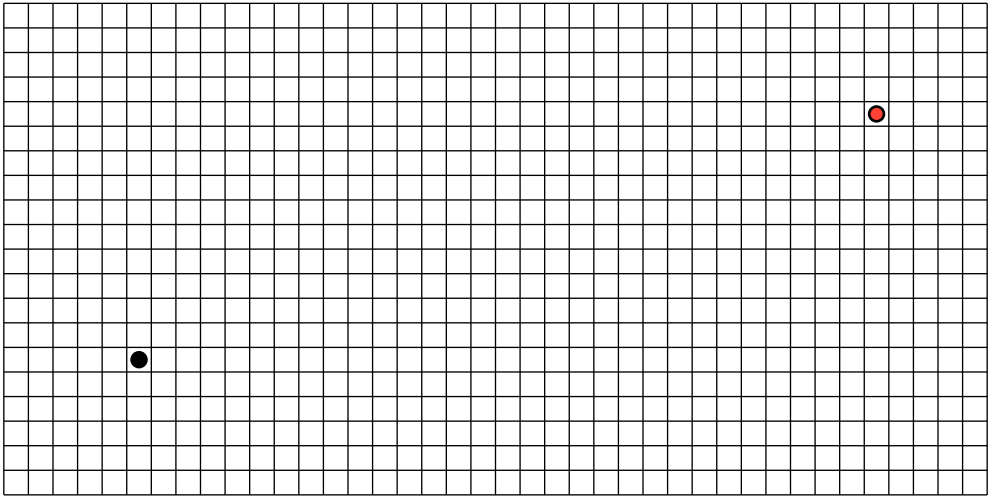


Fig. 13. – Histogramme LunarLander dernier %

Interprétation résultats



3 Apprentissage offline

Apprentissage offline

L'apprentissage offline entraîne un agent sur un ensemble fixe de transitions sans aucune interaction avec l'environnement.

Cela permet d'acquérir, en se basant sur un jeu de données produit par un expert, un niveau de base élevé pour un acteur ou pour un critique.

On peut alors déployer l'agent dans l'environement pour continuer son apprentissage avec ses connaissances.

Le défi principal est le *distribution shift* : lorsque l'agent commence à interagir avec l'environnement, il rencontre des états et actions non représentés dans les données initiales, ce qui peut causer une réduction dramatique de la performance.

Cal-QL & IQL

IQL

- Utilise l'*expectile regression* pour estimer $\max_a Q(s, a)$
- Évite d'évaluer des actions non vues pendant l'entraînement
- Fonction de perte asymétrique: $L_2^{\tau'}(u) = |\tau' - \mathbb{1}_{u<0}| \cdot u^2$

Cal-QL

- Extension de CQL (Conservative Q-Learning)
- Utilise des retours Monte Carlo comme référence $V_\mu(s)$
- Fixe une borne inférieure pour éviter la sous-estimation excessive

$\Rightarrow$ AFU n'est pas une approche conservatrice. On utilise une décomposition de la fonction avantage ce qui permet théoriquement une transition offline/online plus stable.

Expérience

Hypothèse

AFU a une transition entre la phase d'apprentissage offline et online plus stable par rapport à des approches conservatrices (IQL et Cal-QL).

Méthodologie :

Génération d'un jeu de données à partir de nos politiques entrainées sur *Pendulum* et *LunarLander*, pendant l'apprentissage pour varier les types d'épisodes présent. Entraînement en offline ($200k$ pas) sur *Pendulum* et *LunarLander* à partir des jeux de donnée puis passage en online ($200k$ pas).

- Évaluation tous les 100 pas de temps sur 10 épisodes.
- On réalise 5 fois l'expérience pour des résultats plus significatifs.

Si notre hypothèse est correcte, alors IQL et Cal-QL devraient avoir une dégradation de performance supérieure à celle de AFU.

Résultats Pendulum

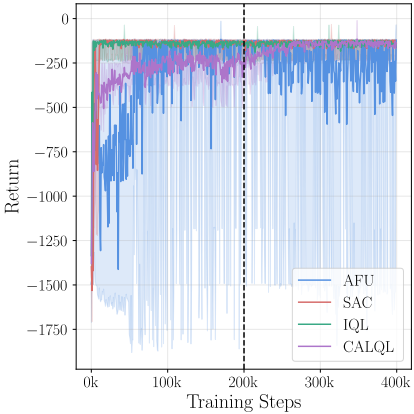


Fig. 14. – Évaluation performance sur Pendulum en offline/online

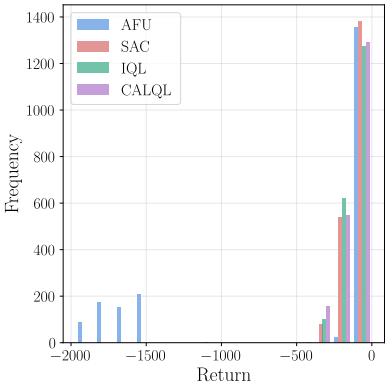


Fig. 15. – Histogramme Pendulum dernier % du online

Interprétation résultats

Pendulum

- Tous les algorithmes obtiennent de bonnes performances en offline.
- SAC atteint presque une politique optimale même en phase offline.
- IQL reste stable pendant la transition, sans déclin attendu.
- CAL-QL, conservateur, montre des progrès limités en phase offline.
- AFU converge partiellement en offline, cohérent avec ses difficultés sur Pendulum.

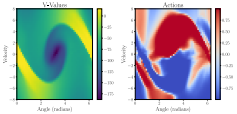


Fig. 16. – V et π de IQL sur pendulum

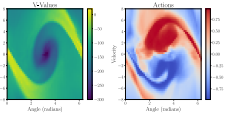


Fig. 17. – V et π de CAL-QL sur pendulum

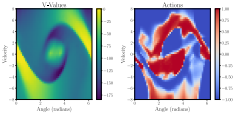


Fig. 18. – V et π de AFU sur pendulum

Résultats Lunar Lander

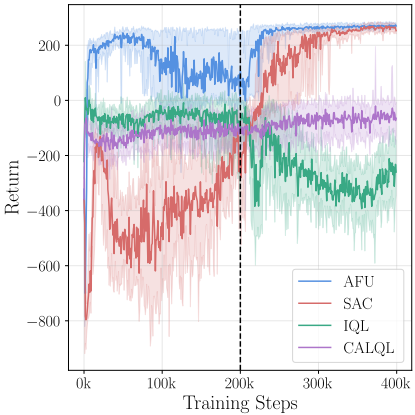


Fig. 19. – Évaluation performance sur Pendulum en offline/online

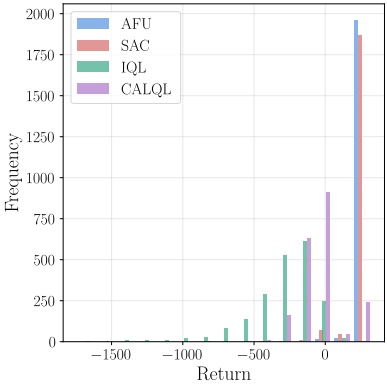


Fig. 20. – Histogramme Pendulum dernier % du online

Interprétation résultats

Lunar Lander

- SAC converge vers une politique de « vol stationnaire » en offline, puis optimale en online.
- IQL reste bloqué sur une politique de compromis, c'est une approche conservatrice.
- CAL-QL maintient une performance intermédiaire, c'est une approche conservatrice,
- AFU atteint une performance élevée dès la phase offline, avec une légère baisse puis une reprise immédiate en online.

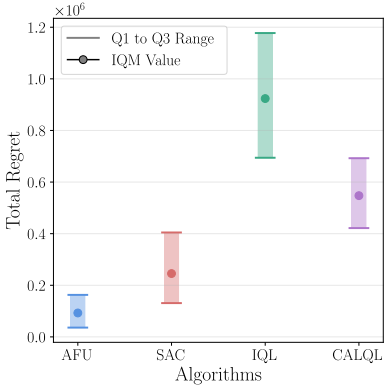


Fig. 21. – Aire sous la courbe après la transition offline/online

4 Conclusion

Conclusion

Résultats principaux

- Apprentissage off-policy: tous les algorithmes performent bien sur des environnements simples, mais échouent sur des environnements complexes nécessitant des séquences d'actions coordonnées.
- Transition offline/online: AFU démontre une stabilité supérieure sur certains environnements, confirmant notre hypothèse sur l'avantage de sa décomposition avantage-valeur plutôt que du conservatisme pour résoudre le problème de la *distribution shift*.

Remerciements

Merci à M. Olivier Sigaud pour son encadrement durant ce projet.

Merci à l'équipe pédagogique pour votre écoute.

Merci finalement à nos familles pour leur soutien.