Cahier des charges

Présentation du projet

Contexte. L'apprentissage par renforcement profond a connu des avancées significatives ces dernières années, notamment grâce à des algorithmes comme *SAC* (Soft Actor-Critic) et *TD3* (Twin Delayed Deep Deterministic) qui ont établi l'état de l'art dans le domaine. Ces algorithmes reposent sur une architecture *acteur-critique*, où un réseau de neurones (l'acteur) apprend à sélectionner des actions tandis qu'un autre (le critique) évalue leur qualité.

Dans ce contexte, l'algorithme *AFU* (Actor-Free critic Updates), récemment développé par M. Perrin à l'ISIR, propose une approche innovante qui s'écarte de ce paradigme. AFU se distingue par sa capacité à apprendre sans dépendre explicitement d'un acteur pour la mise à jour du critique, un caractéristique qui pourrait lui conférer des avantages significatifs en termes de stabilité et de généralisation.

Ce projet s'inscrit dans le cadre du parcours AI2D du Master d'Informatique de Sorbonne Université Sous la direction de M.Sigaud, membre de l'ISIR, nous chercherons à valider expérimentalement les propriétés théoriques d'AFU et à explorer ses avantages potentiels par rapport aux approches traditionnelles.

Objectif de recherche. Notre étude se concentre sur deux hypothèse principales qui, si elles sont validéses, pourraient positionner AFU comme une alternative sérieuse aux algorithmes actuels.

(Apprentissage à partir de données aléatoires) La première hypothèse concerne la capacité d'AFU à apprendre à partir de données générées de manière complètement aléatoire. Dans les algorithmes traditionnels d'apprentissage par renforcement, la qualité des données d'apprentissage dépend fortement de la politique d'exploration utilisée. Une limitation majeure des approches actuelles est leur difficulté à apprendre à partir de donnée très éloignées de leur politique courante. AFU pourrait surmonter cette limitation grâce à sa conception qui sépare complètement l'apprentissage du critique de la politique d'exploration.

(Transition hors-ligne/en-ligne) La seconde hypothèse porte sur la stabilité d'AFU lors de la transition entre apprentissage hors-ligne et en-ligne. Les algorithmes actuels souffrent souvent d'une dégradation significative de leurs performances lors de cette transition. Cette dégradation s'explique par le changement brutal dans la distribution des données d'apprentissage. AFU. grâce à sa nature véritablement *off-policy*, pourrait maintenir des performances stables durant cette phase.

Impact attendu. La validation de ces hypothèse aurait des implications pour le domaine :

– Une plus grade flexibilité dans la collecte de données d'apprentissage
– Une transition plus fluide entre les deux phrases d'apprentissage hors-ligne et en-ligne

Ces avancées pourraient être utiles pour des applications robotiques où la collecte de données est coûteuse et où la stabilité de l'apprentissage est importante.

Cadre technique

Environnement de développement.
– Framework principal: BBRL ;
– Implémentation en python avec PyTorch ;
– Utilisation de Gymnasium pour les environnements de test.

Implémentation.
– Développement de différents algorithmes de DRL avec BBRL ;
– Mise en place d'outils de mesure de performance ;

Protocole Expérimental

Étude de l'apprentissage aléatoire.
– Modification des environnements pour permettre un échantillonnage uniforme ;

Analyse de la transition hors-ligne/en-ligne.
– Phase d'apprentissage hors-ligne avec données pré-collectées
– Transition progressive vers l'apprentissage en-ligne
– Suivi continu des performances pendant la transition

Environnemnets de test.
– Cartpole (version continue)
– Pendulum
– Autres environnemnets MuJoCo selon les besoins

Méthodologie d'évaluation

Métrique de performances.
– Efficacité d'apprentissage
– Stabilité pendant la transition
– Qualité de la politique finale

Analyse comparative.
– Comparaison avec SAC, TD3 et IQL
– Analyse statistiques des résultats
– Visualisation des courbes d'apprentissage

Organisation du projet

Planning.

Phase 1 :
– Mise en place de l'environnement
– Implémentation d'AFU
– Développement des outils de tests

Phase 2 :
– Réalisation des expériences
– Collecte des données
– Analyse préliminaires

Phase 3 :
– Analyse approfondie
– Rédaction du rapport
– Préparation de la soutenance

Livrables.
– Code source documenté
– Résultats expériementaux
– Rapport final
– Présentation pour la soutenance

Critères de réussite

Le projet sera considéré comme réussi s'il :
1. Démontre clairement les capacité ou non d'AFU à apprendre à partir de données aléatoires
2. Quantifie la stabilité d'AFU pendant la transition hors-ligne/en-ligne
3. Fournit des résultats reproductibles
4. Livre une implémentation robuste d'AFU dans BBRL