Rapport d'expérimentation pour la création d'agents RL pour le jeu du Labyrinthe

Contexte et approches testées

Dans le cadre de ce projet, différents environnements, méthodes d’entraînement et configurations ont été testés pour développer des agents de renforcement capables de jouer au jeu du Labyrinthe. L'objectif était de concevoir un agent pouvant gérer des actions complexes, avec des phases distinctes d'insertion/rotation et de déplacement, tout en respectant les contraintes de validité des actions. Cependant, chaque approche a rencontré des obstacles, empêchant d’obtenir des résultats satisfaisants.

Expériences et difficultés rencontrées

Expériences sur l’entrainement en général

1. **Séparation des phases avec des espaces d'actions distincts** :
   - Tentative : Utiliser deux espaces d'actions indépendants pour la phase d'insertion/rotation (`gym.spaces.MultiDiscrete([4, 12])`) et pour la phase de déplacement (`gym.spaces.Discrete(49)`).
   - Problème : L’algorithme PPO de Stable-Baselines3 ne prend pas en charge des espaces d'actions dynamiques, causant ainsi des incompatibilités dans la gestion de la logique de phase lors de l’entrainement.
2. **Entraînement sans masques d'actions** :
   - Tentative : Entraîner l'agent sans masquer les actions impossibles.
   - Problème : L’agent s'est retrouvé face à trop d'options possibles, nécessitant environ 200000 étapes pour gagner une partie. Le temps d'entraînement est devenu prohibitif (environ 10 minutes par partie), rendant cette méthode inefficace.
3. **Limitation du nombre d'étapes par partie** :

- Tentative : Limiter chaque partie à un nombre prédéfini d'étapes si aucune victoire n’est obtenue.
- Problème : Cette approche n'a pas aidé l'agent à apprendre, qui n'a montré aucune amélioration notable dans ses performances en raison d'un nombre de séquences d'apprentissage limité.

4. **Exploration avec une stratégie epsilon-greedy** :
   - Tentative : Utiliser un facteur epsilon-greedy pour encourager l'agent à varier ses actions et éviter des choix répétitifs (comme insérer ou se déplacer toujours aux mêmes endroits).
   - Problème : Bien que cela ait introduit de la diversité dans les actions de l’agent, cela a également faussé la perception de succès. L’agent interprétait certaines actions aléatoires comme bénéfiques, ce qui a nui à l'apprentissage de véritables stratégies de jeu.

Expériences sur la gestion des actions impossibles

1. **Pénalisation des Actions Impossibles** :
   - Tentative : Appliquer une pénalité lors de la sélection d'actions impossibles (comme un déplacement vers une cellule inaccessible).
   - Problème : Malgré cette pénalisation, l'agent continue d'essayer d'accéder à des cases inaccessibles une fois entraîné, indiquant une faible compréhension des contraintes de mouvement.
2. **Utilisation de masques d'actions** :
   - Tentative : Appliquer manuellement des masques d'actions.
   - Problème : Les masques d'actions n'étaient pas transmis à l'agent, obligeant une recréation des masques côté algorithme d’entraînement, ce qui a augmenté la complexité sans permettre un apprentissage fonctionnel.
7. **Retour du Masque en tant qu'Information Post-Étape** :
   - Tentative : Renvoyer le masque d'actions après chaque étape (`step()`) comme information pour guider l'agent.
   - Problème : L’algorithme PPO de Stable-Baselines3 ne sais pas interpréter et exploiter cette information, menant à un apprentissage inefficace et incohérent.

Environnement Actuel et Problèmes Associés

Utilisation de sb3_contrib

Pour remédier aux limitations identifiées dans les configurations précédentes, l'utilisation de `sb3_contrib` a été introduite pour gérer les masques d'actions au sein de l'algorithme Stable-Baselines3.

> 💡 `s3b_contrib` est une extension de la bibliothèque Stable-Baselines3 qui fournit des outils et des algorithmes supplémentaires pour les environnements de renforcement complexe, incluant le masquage d'actions.

Afin de surmonter les incompatibilités rencontrées avec l'algorithme PPO standard de SB3, nous avons utilisé `MaskablePPO`, une version adaptée de PPO disponible dans `sb3_contrib` qui prend en charge le masquage d'actions.

Emplacement du code

Le code de l'environnement se trouve sur la branche `env_mask`, dans le fichier `gym_env_2dim_modif.py`, tandis que le script d'entraînement est dans le fichier `entrainement_agents.ipynb`. L'adaptation du jeu pour intégrer les IA a été réalisée dans la boucle `start()` de `GUI_manager.py`, et le modèle utilisé est spécifié à la ligne 47 du fichier `game.py`.

Difficultés Rencontrées :

1. **Absence de statistiques à l'entrainement**

Lorsqu'on tente d'intégrer des statistiques d'entraînement avec des callbacks, les masques d'actions sont supprimés, ce qui entraîne des comportements inattendus de l'agent, comme l'exécution d'actions interdites. Ce problème empêche également un suivi fiable de la performance et de la progression de l'agent.

2. **Incompatibilité des masques à l'intégration**

L’utilisation de `sb3_contrib` provoque également des erreurs lors de l'intégration de l'agent entraîné dans le jeu pour des parties contre un joueur humain. Les tailles des masques ne sont pas correctement reconnues, ce qui entraîne une incompatibilité entre l’espace d’actions masqué et la structure de `sb3_contrib`.

En conséquence, nous n'avons pas pu tester le comportement des agents entraînés dans des parties jouables, ni évaluer leur performance pendant l'entraînement, ce qui limite toute analyse de leur progression et de leur efficacité en conditions de jeu réel.

Conclusion

À ce stade du projet, malgré de nombreuses configurations et approches explorées, les résultats obtenus restent limités. Les diverses configurations de l’environnement, les tentatives de masquage des actions, et les méthodes d’entraînement n’ont pas encore permis de produire des agents fonctionnels, ni un environnement stable.

L'incompatibilité entre les algorithmes de renforcement et les besoins spécifiques de masquage des actions dans ce jeu a, pour l’instant, freiné l'efficacité de l’entraînement et le développement d’agents adoptant des stratégies efficaces contre des adversaires humains.

Le dernier environnement testé, bien qu’intégrant le masquage des actions, présente encore des problèmes de compatibilité et de suivi, limitant actuellement les avancés dans ce projet.

Liens utiles

https://sb3-contrib.readthedocs.io/en/master/modules/ppo_mask.html

https://costa.sh/blog-a-closer-look-at-invalid-action-masking-in-policy-gradient-algorithms.html

https://arxiv.org/pdf/2006.14171