

Développement d'un moteur d'échecs adaptatif personnalisé par apprentissage automatique

Encadrant : Nicolas Baskiotis, ISIR, Sorbonne Université
(prenom.nom@sorbonne-universite.fr)

Mots-Clés : Jeux d'échecs, Apprentissage automatique, Modélisation du joueur, Adversaire adaptatif, Reinforcement Learning

Nombre d'étudiants : 2

Résumé : L'objectif du projet est de développer un moteur d'échecs capable de s'adapter dynamiquement au niveau et au style de jeu d'un joueur spécifique, en combinant analyse de parties historiques et adaptation en temps réel, puis d'évaluer rigoureusement sa capacité à maintenir des parties équilibrées et engageantes.

Sujet :

Les moteurs d'échecs modernes comme Stockfish, AlphaZero ou Leela Chess Zero [5] ont atteint un niveau surhumain, rendant difficile leur utilisation pour l'entraînement de joueurs humains. Bien que ces moteurs puissent être limités artificiellement (par profondeur de recherche ou temps de calcul), cette approche ne produit pas des adversaires réalistes : un Stockfish limité joue soit parfaitement soit fait des erreurs aléatoires, sans reproduire les patterns d'erreurs humaines. Des projets comme Maia [1] ont proposé d'apprendre à imiter les joueurs humains à différents niveaux Elo, mais ces modèles restent génériques et ne s'adaptent pas au profil individuel d'un joueur.

Un adversaire idéal pour l'apprentissage devrait combiner plusieurs propriétés : (1) s'adapter au niveau réel du joueur pour maintenir des parties compétitives, (2) reproduire des erreurs humaines exploitables mais non triviales, (3) créer des positions tactiquement intéressantes, et (4) s'ajuster dynamiquement si le joueur progresse ou joue en dessous de son niveau habituel. Ce projet vise à développer un tel système en exploitant les techniques d'apprentissage automatique pour modéliser et s'adapter à un joueur spécifique.

L'approche proposée repose sur deux phases complémentaires. La première phase consiste à analyser l'historique de parties d'un joueur pour construire un profil détaillé : types d'erreurs récurrentes (tactiques, positionnelles, finales), style de jeu (agressif, défensif), performance selon la phase de partie (ouverture, milieu, finale), et variance de niveau. Ce profil servira de base pour initialiser le comportement du bot. La seconde phase implémente l'adaptation en temps réel : pendant une partie, le bot ajuste son niveau en fonction de la qualité des coups joués, en utilisant des mécanismes comme la limitation sélective de profondeur de recherche, l'injection contrôlée d'erreurs similaires à celles du joueur, ou l'utilisation de réseaux de valeur personnalisés.

Le projet comparera plusieurs approches techniques : (1) des méthodes basées sur la limitation adaptative de Stockfish, (2) le fine-tuning de modèles pré-entraînés comme Maia sur les parties du joueur cible, (3) l'apprentissage par renforcement d'un agent optimisant l'équilibre des parties, inspiré de travaux comme KataGo [4], et (4) des architectures hybrides combinant un moteur fort avec un module de décision appris. L'évaluation constituera un défi méthodologique

important : au-delà des métriques classiques (winrate, distribution des évaluations), il faudra mesurer la qualité de l'expérience (complexité positionnelle, opportunités tactiques créées) et la capacité d'adaptation réelle du système.

Le projet comprendra les étapes suivantes :

1. Collecte et préparation d'un dataset de parties annotées depuis Lichess avec métadonnées Elo
2. Développement d'un module d'analyse de profil joueur (extraction de features, classification d'erreurs)
3. Implémentation de baselines : Stockfish à profondeur/temps limité, Maia au niveau Elo correspondant
4. Développement du bot adaptatif avec plusieurs stratégies d'adaptation (limitation sélective, injection d'erreurs, RL)
5. Conception d'un protocole d'évaluation automatique rigoureux (métriques d'équilibre, qualité, adaptation)
6. Analyse comparative des approches et identification des limites et perspectives d'amélioration

Ce projet permettra aux étudiants de se familiariser avec les problématiques d'apprentissage automatique appliquées aux jeux, de modélisation du comportement humain, d'évaluation de systèmes interactifs, et d'intégration de moteurs de jeux existants avec des composants ML. Il combine des aspects techniques variés (analyse de données, ML supervisé et par renforcement, évaluation empirique) et offre des perspectives d'extension vers la génération d'exercices personnalisés ou le commentaire automatique de parties.

Références

- [1] McIlroy-Young, R., Sen, S., Kleinberg, J., and Anderson, A. (2020). Aligning Superhuman AI with Human Behavior : Chess as a Model System. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20), pp. 1677–1687.
- [2] Stockfish Team. Stockfish : Open source chess engine. <https://stockfishchess.org/>
- [3] Silver, D., Hubert, T., Schrittwieser, J., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419), 1140–1144.
- [4] Wu, D. J. (2019). Accelerating Self-Play Learning in Go. *arXiv preprint arXiv:1902.10565*.
- [5] Pascutto, G. C., and Linscott, G. (2017). Leela Chess Zero. <https://lczero.org/>
- [6] McIlroy-Young, R., Wang, R., Sen, S., Kleinberg, J., and Anderson, A. (2022). Learning Models of Individual Behavior in Chess. In Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '22), pp. 1253–1263.