Devoir à la maison n° 2

Cours d'Intelligence Artificielle : IA_Solve

(À remettre le 17 novembre 2022)

Ce devoir à la maison est à faire **par trinôme** et est noté sur 20 points.

Ce DM a pour objectif de vous faire programmer et tester les algorithmes MonMax, Alpha-Beta et MCTS, ici sur le jeu d'Othello. Celui-ci est suffisamment simple pour qu'il soit facile d'y jouer, mais suffisamment riche pour qu'il ne soit pas si facile de programmer un algorithme de jeu qui joue bien.
Pour ceux qui le souhaiteront, il est possible d'aller une étape plus loin et d'utiliser un réseau de neurones pour apprendre une fonction d'évaluation.

Vous pouvez programmer en partant de zéro, ou en vous inspirant des codes écrits pour le livre [Pumperla, Max & Ferguson, Kevin : "Deep Learning and the Game of Go", Manning, 2019], disponibles sur le GitHub `https://github.com/maxpumperla/deep_learning_and_the_game_of_go/blob/master/code/dlgo/goboard_slow.py`.
Ces codes couvrent la programmation du jeu de Go, l'algorithme MinMax, avec une fonction d'évaluation simple(tte), l'algorithme Alpha-Beta, l'algorithme MCTS et un apprentissage de fonction d'évaluation par un réseau de neurones comme pour AlphaGo.
Cependant, l'adaptation du jeu de Go à Othello nécessite de bien comprendre ce qui est dans le code. C'est très instructif, mais demande beaucoup de soin.

Ce qui est attendu dans le projet

1. La programmation d'un **jeu d'Othello** permettant à deux joueurs de jouer à tour de rôle. Il faut prévoir que la machine puisse jouer contre elle-même, mais aussi que l'un des deux joueurs soit un humain.
2. La programmation d'une **fonction d'évaluation** retournant un réel reflétant la promesse de gain pour le joueur considéré pour une position donnée.
3. La programmation de l'**algorithme MinMax**. Vous testerez alors la performance de cet algorithme sur un ensemble de parties et pour des profondeurs d'exploration diverses.
4. La programmation de l'**algorithme Alpha-Beta**. Vous testerez alors la performance de cet algorithme sur un ensemble de parties et pour des profondeurs d'exploration diverses.
5. La programmation de l'**algorithme MCTS**. Vous testerez cet algorithme en variant la valeur de la constante C et le nombre de simulations lors des « Roll-outs ».

Vous fournirez les codes utilisés et un petit rapport décrivant vos choix, vos résultats et vos analyses.

Vous nommerez les fichiers correspondants `UnNom_rapport_DM-2` et `UnNom_code_DM-2`.