Deep Checker

Apprentissage statistique et intelligence artificielle

Arthur Correnson, Igor Martayan, Manon Sourisseau

Projet de Statistiques, ENS, 2021

Introduction

- Construction d'une heuristique évaluant la qualitée des coups
- Nécéssité d'un grand nombre de partie de jeu de dames

Plan de la présentation

1. Génération de données et simulateur
2. Modèles et heuristiques
3. Régression aux k plus proches voisins
4. Réseau de perceptron

Génération de données et simulateur

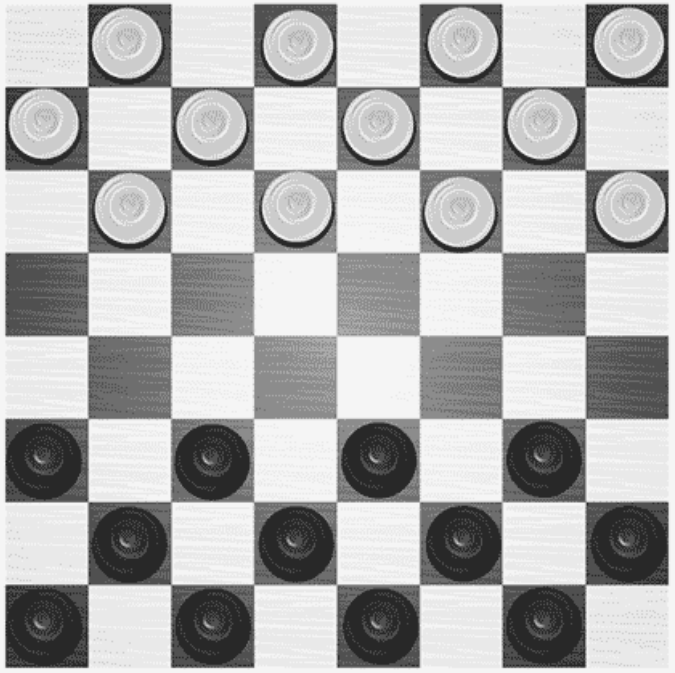
Génération de données

- Besoin d'un grand nombre de données
- Générer beaucoup de parties rapidemment et de manière compacte en mémoire
- Ecriture d'un simulateur dans le langage C

Création d'un simulateur

- 64 cases, 32 cases possibles
- 3 états possibles par cases :
  Vide, pion blanc, pion noir

Représentation de l'état du plateau par un entier de 64 bits (2 x 32 bits)
exemple : 1111 1111 1111 0000 0000 0000 0000 0000 // 0000 0000 0000 0000 0000 1111 1111 1111



Données et performances

- Les données sont stockées dans un fichier texte.
  (perte d'éfficacité contre simplicité de traitement des données)
- Performances très satisfaisantes :
  10 000 parties générées en environ 1 secondes, sur un ordinateur ordinaire.

Modèles et heuristiques

Modèles et heuristiques

On souhaite construire une heuristique qui attribut un score à un coup donné, selon la qualité du coup. Plusieurs approches pour déterminer l'heuristique :

- Régression par les K plus proches voisins (KNN)
- Réseaux de perceptron

Le but est de jouer le coup possible ayant le meilleur score.

Modélisation du problème

Étant donné un ensemble $\mathcal{X}$de parties simulées. n souhaite donner une première approcximation de l'heuristique h.

- On associe chaque coup apparaissant dans une partie $P \in X$ un score
- Le score final d'un coup c est la moyenne des scores qui lui sont attribués sur l'ensemble des parties dans X

Régression aux k plus proches voisins

Régression par KNN

KNN : méthode de régression aux K plus proche voisins. On définit la distance entre deux coups par la distance d'édition :

$$\langle c_1, c_2 \rangle_{KNN} = \| c_1 \oplus c_2 \|_1$$

$\rightarrow$ Les coups sont représentés comme des entiers de 128 bits.

- On calcule la distance du coup donné avec tous les autres coups.
- Le score attribué au coup donné correspond à la moyenne des scores des K plus proches voisins.

Résultat et performances de KNN

Victoires du joueur 1 (KNN)	Victoires du joueur 2 (Aléatoire)
18	32

Cette version de l'heuristique est peu satisfaisante

Réseau de perceptron

Nouveau calcul de score

- Nouvelle fonction d'évaluation :
  $\|.\|_i : D_i \to \mathbb{N}$ définie comme : $\|d\|_i = p(d).v(d)$
  - D_i : Ensemble des états du damier vu par le joueur 1
  - $p(d) \in \mathbb{N}$: Nombre de pions mangé depuis l'état d
  - $v(d) = 1$ si le joueur 1 gagne, $v(d) = 0$ sinon
- On construit maintenant une fonction $w_i(c)$ telle que $w_i(d) = \|c\|_i$ si $d \in D_i$ et $w_i(d) = 0$ sinon
- Pour chaque état de damier d apparaissant dans l'ensemble des parties de $\mathcal{DB}$, $h(d) = \frac{1}{N}\Sigma w_i(d)$ avec N le nombre de parties P_i tels que $w_i(d) \neq 0$ (d est l'un des états pris par le damier dans P_i)

Réseau de neurones type perceptron multicouche

- Entrées : vecteurs de 64 bits (représentant un unique damier)
- Coeur du réseau : 4 couches intermédiaires
- Noeuds du réseau : fonction d'activation relu
- Sortie du réseau de dimension 1 (régression) : combinaison linéraire des 16 sorties de la dernière couche puis d'une application de relu

Résultats du réseau perceptron

Victoires du joueur 1 (MLP)	Victoires du joueur 2 (Aléatoire)
50	0

$\rightarrow$ Heuristique bien plus satisfaisantes