

Projet - une IA qui apprend à jouer au morpion

1 Enoncé du projet

Ce projet consistera à coder une intelligence artificielle qui apprendra d'elle-même à jouer au morpion. L'idée derrière une telle machine a été originellement proposée en 1961 par Donal Michie (projet MENACE). La machine fonctionnait à l'aide de simples boîtes d'allumettes, qui servaient à mémoriser les coups gagnants au morpion. Au fur et à mesure des parties, les informations contenues dans chaque boîte d'allumettes sont affinées, jusqu'à ce que l'ensemble sache jouer parfaitement au morpion.

Une description complète du système, qu'il vous faudra lire avant de passer à la suite, est disponible ici :

<http://images.math.cnrs.fr/Une-machine-en-boites-d-allumettes-qui-apprend-a-jouer-au-Morpion.html>

2 Représenter une grille de morpion

Dans un premier temps, il vous faudra décider de la structure de données à adopter pour coder une des "boîtes d'allumettes" de la machine originale. Chaque boîte doit représenter une combinaison de croix, de cercles et de cases vides sur une grille de 3x3 cases.

On pourrait représenter, dans l'ordinateur, chaque grille par un tableau de 3x3 cases, qui contiendrait des cercles, des croix et/ou du vide. Il existe cependant une représentation plus simple et surtout, plus compacte en mémoire faisant intervenir la base numérique 3. Chaque élément de la grille représentera un nombre 0, 1 ou 2, et la position de l'élément dans le tableau représentera sa position dans le nombre base 3. Par exemple, considérons la grille suivante :

X	O	
O	X	X
	X	

Si l'on associe le chiffre 2 aux croix, le 1 aux cercles et le 0 aux espaces, et que la case en bas à droite représente les unités, nous aurons, en base 3, le nombre 210122020. Ce nombre se traduit, en base 10, comme

$$0 + 2 * 3 + 0 * 3^2 + 2 * 3^3 + 2 * 3^4 + 1 * 3^5 + 0 * 3^6 + 1 * 3^7 + 2 * 3^8 = 15774$$

Questions

Voici un ensemble de questions vous permettant de construire votre réflexion pour l'architecture de votre programme. Il n'est pas demandé de répondre directement, dans votre rapport, à ces questions, mais il faudra que votre texte ou vos diagrammes permettent d'éclairer les éléments abordés dans ces questions.

1. Combien de configurations différentes qu'il est possible de former en plaçant au hasard des cercles, des croix ou rien du tout dans une grille 3x3. Chaque configuration représente-t-elle une grille valide au morpion ?
2. Vous devriez avoir un moyen de "traduire" une grille 3x3 de morpion en un codage entier, selon la méthode expliquée précédemment, et inversement.
3. A chaque configuration possible, on souhaite associer un certain nombre de billes correspon-

gant aux cases libres, comme dans le système MENACE. Ce nombre de billes représentera la probabilité de choisir au hasard le prochain coup à jouer lorsque l'on est dans une configuration précise. Quelle structure de données allez-vous adopter afin de représenter les configurations de grilles et les billes associées ?

3 Faire le tri parmi les configurations

Certaines grilles sont identiques, à une transformation près, à d'autres grilles. Par exemple, ci-dessous, la grille du centre est identique à la grille de gauche à une rotation de 180° près, tandis que la grille de droite est identique à la grille de gauche à une symétrie verticale suivie d'une rotation de 90° près.

O	O	
O	X	X

X	X	O
	O	O

	X	
	X	O
	O	O

On parlera alors de configurations "similaires". Les transformations possibles entre les grilles sont les rotations de 90 , 180 ou 270 degrés, combinées avec la symétrie verticale.

Pour chaque grille, on peut donc trouver sept configurations similaires par symétrie : la grille tournée de 90° , 180° ou 270° , ainsi que la grille transformée par symétrie verticale, tournée de 0° , 90° , 180° ou 270° . Il est impératif pour que votre intelligence artificielle fonctionne, que vous soyez capable d'identifier les configurations similaires, pour les représenter par le même "objet", avec les mêmes "billes", dans votre programme. Lors que vous modifierez le nombre de billes associé à une case libre d'une configuration, il faudra que les cases correspondantes dans les configurations similaires profitent de la modification.

Vous trouverez, joint à ce projet, des fichiers de code pour vous aider sur cette partie. Le programme permet de générer, pour toutes les configurations de grilles possibles (représentées sous forme de tableaux de caractères), toutes les grilles similaires et de les écrire, sous forme de chaîne de caractères, dans un fichier (qui est aussi joint). Vous pouvez reprendre librement le code qui a été donné, et le modifier à votre convenance.

Questions

Comme précédemment, les points suivants sont présents pour vous aider à construire votre réflexion sur le sujet, et il n'est pas demandé d'y répondre directement. Cependant, votre rapport ou vos diagrammes devront permettre de comprendre les solutions que vous apportées aux problèmes soulevés dans ces questions.

- Vous devrez avoir un moyen de décider si une configuration donnée (et ses configurations similaires) est une fin de partie ou non (victoire d'un joueur ou partie nulle).
- Les configurations "similaires" (à une rotation/symétrie près) devront partager leur nombre de billes. Par exemple, considérons les trois grilles de la figure précédente. La case en haut à gauche de la grille de gauche doit partager le même nombre de billes que la case en bas à droite de la grille du centre et que la case en bas à droite de la grille de droite.

4 Le déroulement d'une partie

Le programme proposera soit à l'utilisateur de jouer contre la machine, soit de laisser la machine apprendre les meilleurs coups au morpion en jouant contre elle-même. Dans le premier cas, la machine choisit au hasard qui commence la partie (elle ou la joueur humain), et cette dernière utilise son "intelligence artificielle" pour tenter de gagner. A l'issue de la partie, elle raffiner ses "billes" afin d'apprendre de ses erreurs si elle avait perdu.

Dans le second cas, la machine demande à l'utilisateur une valeur N , et jouera alors N parties contre elle-même pour tenter de raffiner ses "billes" et devenir plus "intelligente".

Le programme proposera aussi à l'utilisateur de quitter : dans ce cas, le programme sauvegardera le nombre de billes associé à chaque case libre de chaque configuration de grille dans un fichier pour les récupérer la prochaine fois. Vous pouvez aussi sauvegarder dans votre fichier, si vous le souhaitez, tout élément qui vous semble pertinent pour accélérer l'initialisation de votre programme au prochain démarrage.

A chaque démarrage, le programme vérifiera si un fichier de configuration existe, et le chargera le cas échéant. L'utilisateur pourra, parmi les options du programme, choisir de détruire le fichier pour commencer avec une intelligence artificielle toute neuve.

5 Ce qui est à faire

Vous devrez proposer un programme qui répond au cahier des charges précédent. Votre programme doit proposer à une personne de jouer au morpion contre l'intelligence artificielle (en affichant la grille de morpion à chaque tour), ou bien doit apprendre, en jouant contre lui-même, les meilleurs coups au morpion. Si vous le pouvez, vous pouvez aussi proposer à l'utilisateur de modifier les règles du jeu (taille de la grille, nombre de pions à aligner pour gagner, ...) et laisser la machine apprendre de ces nouvelles règles.

Vous devrez rédiger un rapport complet sur votre solution : description des structures de données adoptées, choix des structures utilisées à différentes étapes de votre programme, extraits de code qui vous paraissent importants d'expliquer dans le rapport. Votre rapport doit être un journal de bord permettant d'expliquer chacun de vos choix et chacune des difficultés rencontrées. N'hésitez pas à bien regarder sur Moodle le guide de rédaction d'un rapport, afin de comprendre ce qui est attendu de vous. La note de votre projet sera le minimum entre votre note de rapport et votre note de code.