Making a Connect-4 bot using Machine Learning techniques

Jan Moran Ricardo (1633686), Aleix Jordà Banús (), Daniel Redondo Nofre (1634361),
Ferran Bocanegra Franquesa (), Eric Herreros Vidal ()
Computació d'Altes Prestacions, GEI

Abstract

1. Introduction

In this paper, we present a Connect-4 bot developed using various machine learning techniques (ML)...

Even though the ML techniques and algorithms are complex by themselfs, the focus of the research is on the parallelization and subsequent optimization of said algorithms. For that we've used a series of shared and distributed memory techniques, such as OpenMP and MPI...

2. Methodology

In this section, we will describe the methodology used to develop the Connect-4 bot. We will cover the game itself, the machine learning techniques used, and the parallelization methods implemented.

2.1. The connect 4 game

We all know and love the connect 4 game, so this section will be brief.

The connect 4 game is a two player game that consists on the placement of discs in a 7x6 grid, where the objective is to connect 4 discs of the same color in a row, column or diagonal. The catch is that the discs are placed from the top of the grid, and they fall to the lowest available position in the column.

Coding the game was a relatively simple task. **We've defined the game board as a 2D array of integers**, where 0 is an empty cell, 1 is a player one disc and 2 is a player 2 disc. Function wise, we've defined 4 different functions:

- **startGame**: initializes the game board, and the game mode (human vs human, human vs AI, AI vs AI).
- **dropPiece**: checks if the column is valid, and if it is, drops the piece in the lowest available position in the column.
- **checkWin**: checks if the last move made by a player has resulted in a win.
- **displayBoard**: prints the game board to the console.

2.2. The Machine Learning techniques

// Determinar i explicar totes les tècniques de ML que hem fet servir.

2.3. Parallelization

// Explicar com hem paral·lelitzat el programa*. Aquí cal fer com a les PLABs, executar el programa en single thread, i veure que consumeix més. A partir d'aquir intentar millorar fent servir ML o OpenMP (jo no em liaria amb cuda, però si algú s'atreveix...)

// Les subseccions que siguin de les funcions a paral·lelitzar.

// * Potser, en comptes de paral·lelitzar IA vs IA intentar mirar de fer 1 sol moviment de la IA, ja que pot trigar molt, sobretot en seqüencial.

3. Results

// Aquí cal posar totes les taules de temps, speedup, eficiència, etc. del millor cas.

4. Conclusion