EISTI Ecole Internationale des Sciences du Traitement de l'Information

Master ADEO2 2018-2019

Machine Learning (Scala)

Bart George

Scala / Machine Learning Project: Connect Four with AI

Project objectives

The goal of the project is to release, in Scala, a program that allows the computer to play a game of Connect Four (either with a human player or with itself) and learn how to play better from its previous games.

Connect Four : Overview and rules

"Connect-Four is a game for two persons. Both players have 21 identical men. In the standard form of the game, one set of men is yellow and the other set is red. The game is played on a vertical, rectangular board consisting of 7 vertical columns of 6 squares each. If a man is put in one of the columns, it will fall down to the lowest unoccupied square in the column. As soon as a column contains 6 men, no other man can be put in the column. Putting a man in one of the columns is called: a move.

The players make their moves in turn. There are no rules stating that the player with, for instance, the yellow men should start. Since it is confusing to have to identify for each new game the colour that started the game, we will assume that the sets of men are coloured white and black instead of yellow and red. Like chess and checkers (and unlike go) it is assumed that the player playing the white men will make the first move.

Both players will try to get four connected men, either horizontally, vertically or diagonally. The first player who achieves one such group of four connected men, wins the game. If all 42 men are played and no player has achieved this goal, the game is drawn."

(source : *"A Knowledge-based Approach of Connect-Four : The Game is Solved: White Wins"*, Victor Allis, Master Thesis, Department of Mathematics and Computer Science, Vrije Universiteit, Amsterdam, October 1988)

Project Requirements

You must write a program in Scala that allows two players to have a game of Connect Four according to the rules described above, with a board of 7 columns * 6 rows, with 21 discs for each one of the two contenders.

At most one of the players can be human. Which means the program must be able to run "human vs. AI" and "AI vs. AI" games.

The computer AI must be able to learn from its previous games. You are totally free to choose the method you want, as long as it concerns Machine Learning (neural networks, decision trees, genetic algorithms, deep learning, k-nearest neighbours, etc...).

The development language must be Scala. This being said, you are totally free to choose whatever framework or library you want. However, you must ensure that I can run the program on my machine. Also, you must bring your own ideas and your own code. Plagiarism will be punished harshly.

The final delivery must include a runnable program, the source code, and a technical report, in which you will explain which method, framework and libraries you chose, and describe your results, as well as the advantages and disadvantages of your program.

Among the factors that will have an influence on the final mark, the quality and robustness of the AI, and its capacity to learn from its previous games is the most important. Graphical UI does not play such a big part (although you shouldn't neglect the overall ergonomy of the program). To say it another way : a program playable only on console, but with a great fast-learner AI will have a good mark, while a program with astonishing graphics, but no AI or a poor AI will have a very, very bad mark. Another lesser factor of influence may be the inclusion of different rules, variants and board dimensions (5 columns x 4 rows, 6 columns x 5 rows, infinite connect-4...) as well as the AI's capacity to adapt quickly and learn from these variants.

Number of students per team : 1, 2 or 3.

Delivery date : ***February 24th, 2019***, by email to bge@eisti.eu (if the files are too big, you can email me a link to a downloadable file). Delays will not be tolerated.