Reinforcement Learning in Connect 4

Michael Yura
2007-2008

Introduction:

Although an AI is often thought of as being only as intelligent as its programmer, this is not exactly the case; this project will attempt to create an dynamically learning Machine Learner for Connect 4 by using supervised reinforcement learning, with each Learner saving the way that it will play into text files, each directing the way that it will play for a given board layout.

Background:

I expect to have an ML that throughly and hopefully quickly learns to play Connect 4 to an advanced level. Through this project, I hope to learn how fast and to what quality reinforcement learning allows for the learning of a simple game; these methods can hopefully be extended to other, more complex tasks for machines to learn.

Connect 4 has already been solved by James D. Allen and Victor Allis; I will attempt to compare the way the ML plays to the strategies outlined in Allis's *A Knowledge-based Approach of Connect-Four*

Procedures:

I have currently programmed the Connect 4 game itself, as well as created an “ML” (Machine Learner) abstract class that other ML's will be based upon, with methods to load save its board data. This ML appends all of its board data into a single text file and saves the corresponding probability data its own smaller file, which is rewritten after the ML plays a game.

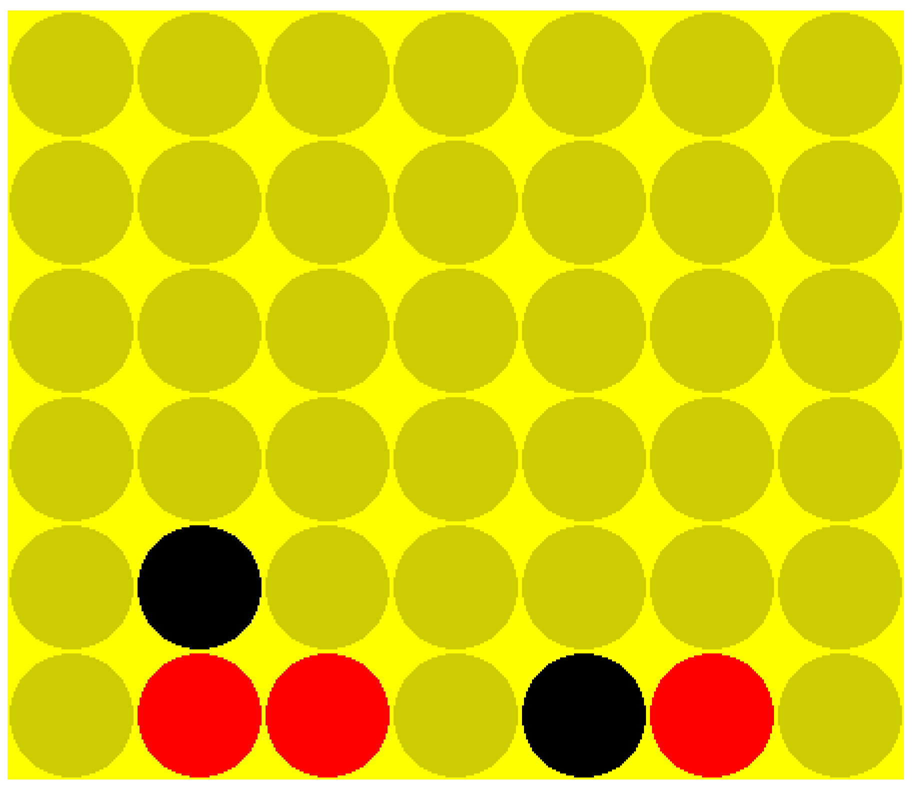
I currently have an ML that does not change the way it places pieces, playing completely randomly, as well as an ML that changes its probabilities uniformly, multiplying or dividing the probabilities of the places where it plays by a single number, whether it wins or loses respectively.

I plan to program an ML that changes probabilities more for ones used later in the game and less for ones used in the beginning. These ML's will change their data to different degrees, some radically changing their strategies after each game, and others doing so to a more moderate degree. The way that each ML changes its strategy will be written by myself, meaning that this is not entirely independent learning, but Supervised Reinforcement Learning.

Expected Results:

Through this project, I hope to find a degree of reinforcement learning that allows the computer to learn to play connect 4 quickly and thoroughly. Although I am quite sure that an ML that learns more progressively will in the end turn out to be better, it may not be the most efficient, due to the time and the size that it would take to create one that would surpass the abilities of an ML whose data is more hastily created. I hope that this project may add to the creation process of AI's.

A board of:



Would be represented in the board data file as:

1 [0,0;0][0,1;0][0,2;0][0,3;0][0,4;0][0,5;0][1,0;1][1,1;2][1,2;0][1,3;0][1,4;0][1,5;0][2,0;1][2,1;0][2,2;0][2,3;0][2,4;0][2,5;0][3,0;0][3,1;0][3,2;0][3,3;0][3,4;0][3,5;0][4,0;2][4,1;0][4,2;0][4,3;0][4,4;0][4,5;0][5,0;1][5,1;0][5,2;0][5,3;0][5,4;0][5,5;0][6,0;0][6,1;0][6,2;0][6,3;0][6,4;0][6,5;0]

Similarly, a probability data file of:

[94.0,15.6,77.2,92.8,100.0,43.3,0.1,]

Would represent a:

94.0/423.0 (22.22%) chance of placing in Column 0
15.6/423.0 (3.69%) chance of placing in Column 1
77.2/423.0 (18.25%) chance of placing in Column 2
92.8/423.0 (21.94%) chance of placing in Column 3
100.0/423.0 (23.64%) chance of placing in Column 4
43.3/423.0 (10.27%) chance of placing in Column 5
0.1/423.0 (0.02%) chance of placing in Column 6