# TJHSST Computer Systems Lab Senior Research Project
# Reinforcement Learning in Connect 4
# 2007-2008

Michael Yura

May 23, 2008

## Abstract

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 with the way that it will play for a given board layout. The final goal of this project will be to determine what degree of reinforcement allows for the Machine Learner to learn to play the best in the shortest amount of time.

**Keywords:**Reinforcement learning

## 1 Introduction

I will create a dynamically learning ML (Machine Learner) for Connect 4. These ML's will learn through reinforcement learning. I they are successful (and win the game), they will do what they did more often; conversely, if they are not successful (and lose the game), they will do what they did less often.

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.

## 2 Background

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

## 3 Procedures and Methodology

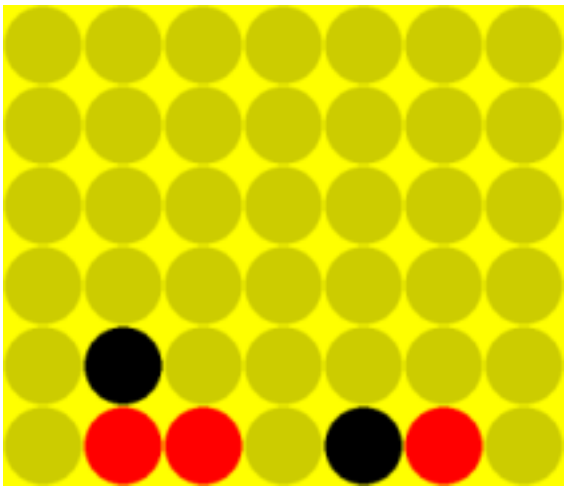
I have currently programmed the Connect 4 game itself, as well as created an ML (Machine Learner) abstract class that other ML's will be based upon, with methods to load save its board data. This ML appends all of its board data into a single text file and saves the corresponding probability data its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

ing that this is not entirely independent learning, but Supervised Reinforcement Learning.

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

..
I will keep track of when different ML's win or lose games against one another and save this data. I will analyze this data to detect any trends, as well

as see how different equations for how to adjust the probability data compare against one another; will a hastily-created set of probability data fare as well as one that was more carefully crafted and that had taken longer to create?

ML's will train against one another, so that they can rapidly change their piece placement with little human effort. I will test to see if an ML can learn by playing against itself, learning by both winning and losing at the same time.

**While I used to experience StackOverflowErrors after playing over 25 games in a row, but I have since solved the problem, which should allow me to play an unlimited number of games without error; I should be able to begin testing the various MLs at the beginning of 4th Quarter.**

# 4 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 throughly. 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. Efficiency in the creation of the ways that MLs place pieces is the final goal of this project, one that I believe I will succeed in. I hope that this project may add to the creation process of AI's.