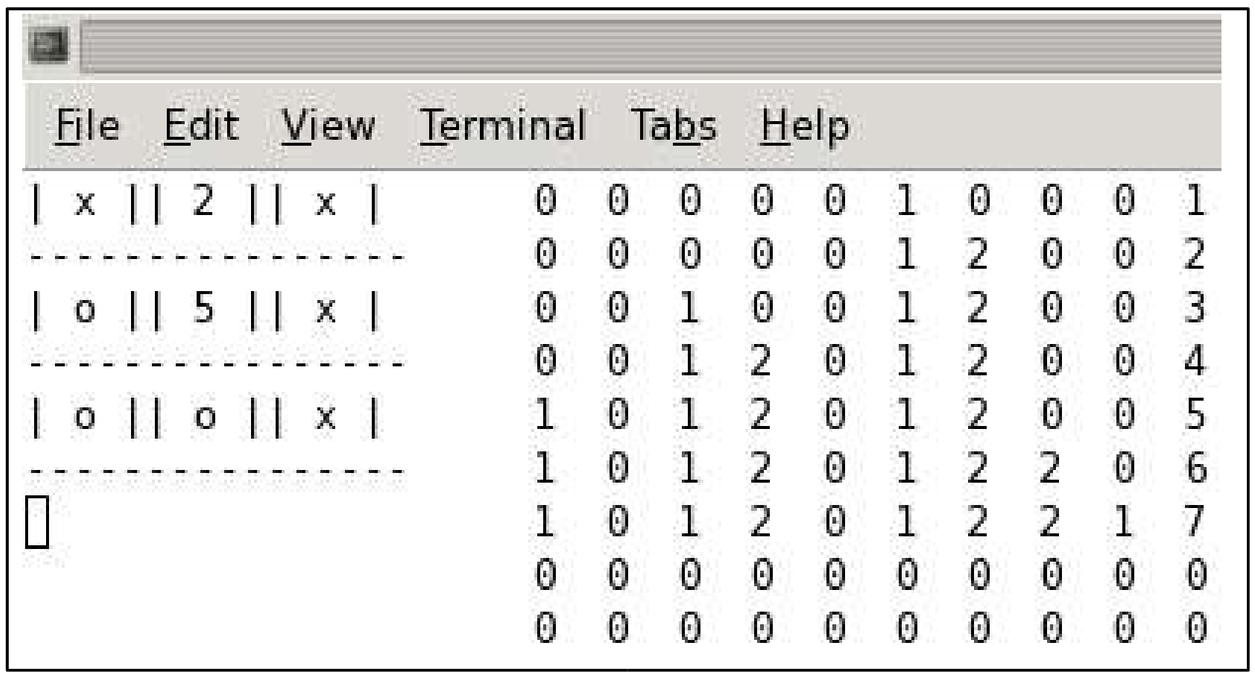
Machine Learning for an Artificial Intelligence Playing Tic-Tac-Toe

Rachel Miller: TJHSST Computer Systems Lab 2005

Problem/Purpose

The use of Artificial Intelligence hopes to make computers better at difficult tasks that typically require a human. Machine learning allows the computer to create its own logical rules, and learn from its past experiences. Machine Learning allows an AI to increase its abilities over time, even without additional direct programmer input. My project hopes to develop a proficiency at Tic-Tac-Toe. Though there are many different types of machine learning AIs, there is no concurrence on the 'best' type of machine learning algorithm; different algorithms have different strengths, such as speed of learning, adaptability, or eventual skill. My project hopes to create a new algorithm for a relatively simple game, Tic-Tac-Toe. Ideally, this algorithm will be modified according to its results to create better algorithms.



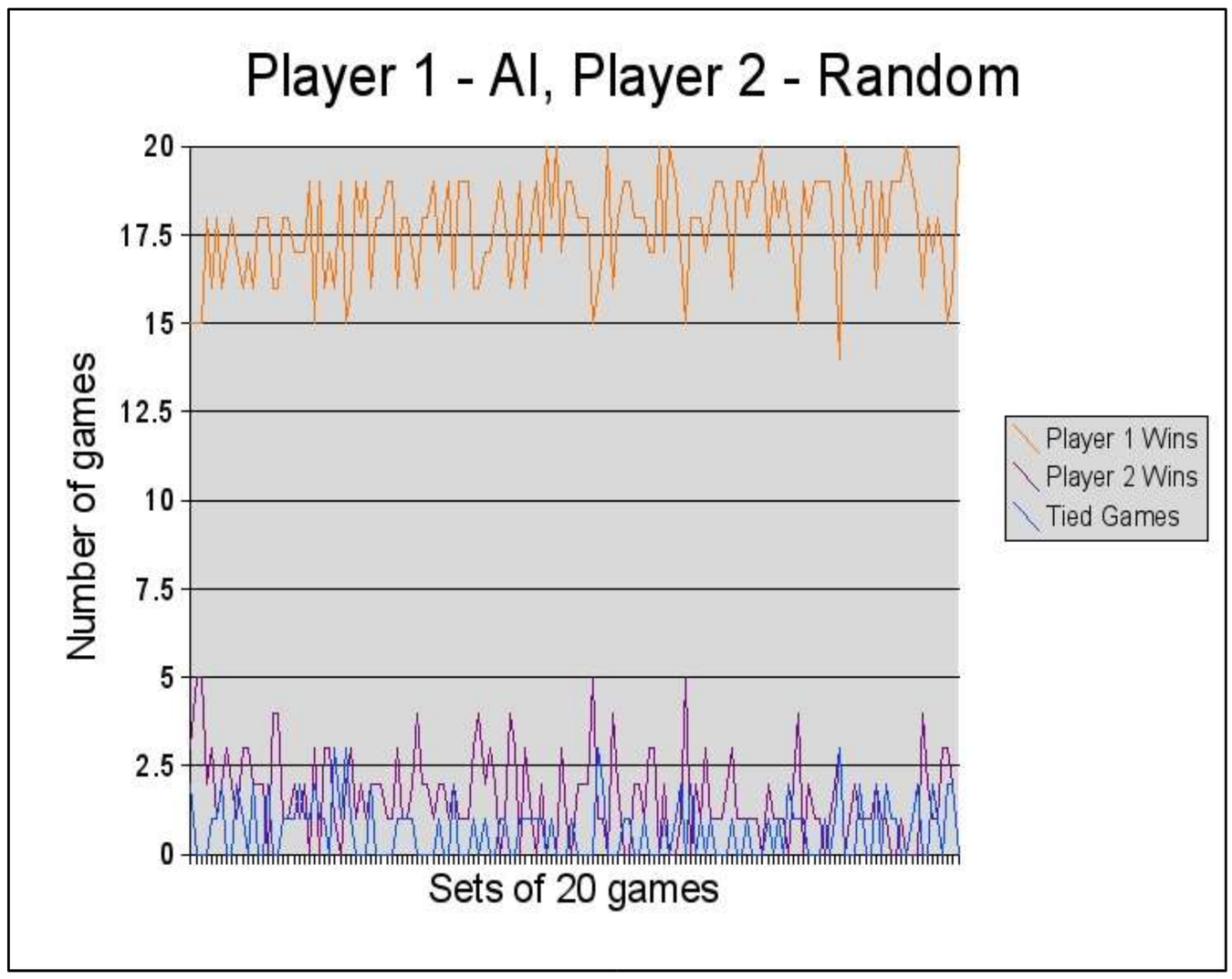
Background

AI learning algorithms include decision trees, neural networks, or genetic learning, among others. One-ply searches of libraries of board positions are most traditionally used for two player games such as checkers or chess. I therefore found this library and search technique most applicable to my program, which would also be designed to play a two player game. Its relative simplicity was also desired, so potentially multiple algorithms could easily be employed, creating new algorithms and test them against each other

Methodology

This project collects data over time, creating its own library of board position values. It will run a game to its finish between two computers, two humans, or a computer and a human. Each position in the board will be considered for its proximity to the final outcome. For example, a position right before a win would be considered valuable, while a position right before a loss would be considered undesirable. If the position already exists in the library, the value of the position in this game will be added to the total value for the position. If the position does not already exist, the algorithm will add it to the library.

Board positions are considered as a one dimensional array of values of 0, 1, and 2. As the computer moves, it considers each possible move. It looks for each move in the library, and chooses the one it thinks has the greatest value.



Procedure

This program started with the basic structure of Tic-Tac-Toe. I then added storage and addition of each board position to the library. Each position was assigned relative values. The AI then references the library, to find the best possible move. Running the AI many times then shows win rates over time.

Expected Results

Expected Results for the program showed a marked increase in playing ability over time, and creation of an extensive library of board positions.