Project Description

Student: Nick Sidawy

Firm: Computer Systems Lab, Rm 115, TJHSST

Mentor: Mr. Randy Latimer

Title: Machine Learning Applications with Othello

Background:

Machine learning is an extensive field of study. Most of what is done with machine learning is tied to artificial intelligence. For example, recently at MIT, students created a small robot child with an artificial intelligence. The goal of the project was to implement machine learning so the robot would be able to teach itself to walk using trial and error.

Othello is a good vessel for my research because it is a simple game. However, it has enough depth in it for a variety of machine learning applications in different aspects of the game. Although the robot child seems as if it is miles away from an Othello artificial intelligence, it is closer than one might think.

Description:

The project is split up into three parts that affect the artificial intelligence in very different aspects.

The first part of the project is creating the algorithm which is the basis for the AI, which is called the "forward-checking" or "minimax" algorithm. The goal of the forward checker is to find the best possible move. In other words, the goal is to traverse a tree of possible moves *n* levels and picking the move that leads to the most advantageous move. The effectiveness of the AI increases with larger values of *n,* but slows down significantly. The trick about this algorithm it picks the move which is best for the player that the current tree level is simulating for. The computer assumes that its opponent plays perfectly.

The forward-checker relies on the evaluation algorithm to return a rating of each board it reaches. The rating is based on the positions of the pieces and the amount of pieces each player has. It is important how the evaluation function weights each aspect of the board in determining how good of a move the AI picks. The Genetic Algorithm(GA) is used to find the best evaluation function. This works using Darwinian principles. It tests the quality of different sets of evaluation values and then creates a new generation by splicing the sets from the previous generation. Those that perform stronger have a better chance of being included in the next generation, hence survival of the fittest. As more generations are created, the evaluation values become more optimized.

The final part focuses on increasing the efficiency and effectiveness of the forward-checker. To do this, the AI learns from each move it takes by recording the board, move and its associated evaluation. This data is stored into a text file. After enough data is collected, the forward-checker can access it using a HashMap. If the AI encounters a board for a familiar it is able to save time spent traversing through the tree of moves and even search the tree more deeply.