CS381K Artificial Intelligence
Assignment 3, due 11/22/2000

Machine Learning in Game Playing

In this assignment, you will apply one or more machine learning techniques to improve the play of your connect-four program of homework 1.

1. Select a learning technique and come up with a plan on how to use it to improve your earlier program. What techniques you use is up to you. You can come up with your own ideas, or implement some of the suggestions below:

   - Design a way to learn macros from playing the game. After your program has won or lost a game, it could try to explain what moves led to wins and losses, and remember those as patterns, or macros. Try to generalize as much as possible.
   - Come up with a set of examples of good moves e.g. by hand, or by collecting samples from actual games played by two programs. Train a neural network through backpropagation to evaluate moves. You could use the final outcome of the game as a target, or the outcome multiplied by the number of moves it took to finish the game, or your own evaluation of the move.
   - Similarly, you could build a decision tree from the same examples to decide which move to make at each situation.
   - Learning to play using reinforcement learning such as Q-learning, TD-λ, or AHC. You should design a representation for the values of the moves (e.g. Q-values in Q-learning), and update them by playing games against opponents in the tournament directory. You might also let the program play itself (or suitably randomized version of itself, which usually allows learning more robust play).
   - Using Genetic Algorithms to evolve a heuristic evaluation function (e.g. as a set of rules, or as a neural network), or a neural network that can rank the available moves, or serve as a filter for minimax. Again, evaluation could be done by playing against other programs or the system itself.

2. Once you have a few ideas of what you would like to do, send email to Risto, or catch him at his office hours (this part is not required, but recommended). He can give pointers to literature and existing code, and generally feedback and suggestions on how to carry out the implementation and experiments.

3. After that, implement your idea: collect the training data, implement the learning algorithm, train the system, and apply the result to your earlier connect-four program.

4. Test your improved program against the other programs in the tournament website; print out log output and analyze what it is doing well and where it falls short.

5. Write a 2-3 page report explaining what you chose to do and why, how you implemented it, and what experiments you ran. Analyze the pros and cons of this approach and outline possibly future improvements. Use samples of log output to illustrate the points (or refer to particular locations in the log output file).

6. Turn in the report, supplemented by the source code and the relevant data files and log output to `martym` using the turnin program. Also turn in a compiled version of the program for the tournament (remember to turn off log output, and make sure it runs against other programs under the interface!)

This assignment is more research oriented than the others in this class. You should begin early, because such an assignment probably takes more time than you think, and it is fun only if you have enough time to work on it. We will again run a tournament and the winners will get extra points!