Recon Blind Chess - Dumbot

Sean Ye
CS 7649 - Robot Intelligence Planning
Prof. Matthew Gombolay
Date: 04/17/2019

Problem Statement

Creating a good recon blind chess agent was the goal of this project. In this version of chess, board positions of the opposing side are unknown and must be sensed each turn. The approach I used consisted of two main portions: reasoning about the board through what the agent has sensed and creating a good evaluation function for each board state in the game. This would ideally allow the agent to choose the best move given knowledge of the board. While these techniques have been used in traditional chess, this methodology has not been evaluated in Recon Blind Chess.

Approach

My approach was based on the paper by Sabatelli et al. [1] where they trained several neural nets to evaluate chess positions based on stockfish's evaluation of boards played by professionals. To produce my dataset, I modified a stockfish bot [2] to play Recon Blind Chess and saved every board position along with its evaluation score. I combined this data with boards from professional chess games [3] to create a dataset of a million board positions. Dumbot uses a neural net trained from these positions to evaluate each of its move options with a depth of one and chooses the move that produces the board with the best evaluation. Dumbot senses squares where its piece was taken. Otherwise it senses randomly. Dumbot also uses four different quick attack openings meant to capture the opposing king in 4-5 moves. If it fails, it continues the game with the neural net.

Mastery of Course Material and Future Work

I utilized a neural network of which backpropagation is a crucial factor in. I built an agent that reasons about sensed board positions and also predicts the state of future board positions. The agent itself does not achieve the same mean-squared-error as the Sabatelli et al. paper and often chooses odd moves such as pushing the king to the center of the board. This could be due to a shortage of data as Sabatelli et al. used 3 million board positions. Additionally, further work can be done on predicting where the opponent pieces are based on sensed locations. I implemented a basic agent that could reason about the game through easy rules (ie if a new bishop is sensed, remove the old bishop on the same color square), but PDDL or constraint propagation could be used to better predict opponent locations.

Contributions

I completed all portions of this project by myself.

Attachments

Code for agent located in dumbotv3.py, weights for neural net named blank, neural network for evaluating positions in blank, helper functions are located in blank and blank.

References

[1] Sabatelli, M., Bidoia, F., Codreanu, V., & Wiering, M. (2018). Learning to Evaluate Chess Positions with Deep Neural Networks and Limited Lookahead, 276283. https://doi.org/10.5220/0006535502760283

[2] https://github.com/reconnaissanceblindchess/reconchess/blob/master/reconchess/bots/trout_bot.py

[3] http://www.ficsgames.org/download.html