Da-Jin Chu
Matthew Gries

Elevator Pitch

Train a single Q-learner to play Othello.

Topics

1 pt: Evaluating your approach in a well-established environment or problem with nontrivial setup, such as the OpenAI Gym.
3 pt: Implementing your own Q-learner with neural network or gradient descent for the backend

Context/Tools

I will use OpenAI Gym and import one of the several third-party Othello environments (like this one (https://github.com/pigooosuke/gym_reversi)).

The language will be python, and we will use pytorch to help us the implement the Q-learner.

Technical Sources

A paper about reinforcement learning in Othello, comparing training on self-play vs. a fxied opponent. (https://www.ai.rug.nl/~mwiering/GROUP/ARTICLES/paper-othello.pdf)

A paper about using convolutional neural networks on Othello (https://arxiv.org/pdf/1711.06583.pdf)

Tutorial for Q learning using PyTorch (https://pytorch.org/tutorials/intermediate/reinforcement_q_learning.html)

Full Disclosure

We implemented the minimax Othello in class, but have not done work on this project prior to now.

Plan

Milestone 1: Setup Othello with Baseline in OpenAI Gym. (November 9)

- OpenAI Gym doesn't come with Othello, so we just need to get a third-party environment and get it all working.
- Port the minimax code to Othello to serve as a baseline.

Milestone 2: Q-Learner (November 27)

- Build and train the Q-learner on self-play.

Milestone 3: Evaluation and Paper (Dec 4)

- Evaluate the performance of the Q-learner vs. minimax.
- Write the summary paper and prepare for lightning talk.

Evaluation

We will compare the performance of the minimax against the Q-learner. Both strategies are deterministic though, so the same player will always win given an initial board state. To get a more representative sample of performance, we will use the methodology the first paper cited in Technical Sources (Ree, 2013) and get every possible board state after 4 moves, and play the game from there. Each state will also be run twice, with the black/white colors swapped. This will give us 572 results, from which we can calculate the win ratio of the two strategies.