Final Project: Training an AlphaZero Agent

February 18, 2025

In this project, you will complete the training and evaluation of your AlphaZero-style agent. This builds on Exercises 7 and 8, where you implemented the game environment and the PUCT algorithm.

1. **Debugging and Preliminary Training**
   - Make sure that the vanilla MCTS player plays at a reasonable level. This may require you to implement features like smart rollouts and move sampling.
   - Pre-train the network using self-play games generated by the vanilla MCTS player. Think about what the value targets and policy targets should be.
   - Debug the PUCT player by training it (if possible) on a small, toy version of your game. For example, instead of Gomoku start by training the agent on tic-tac-toe.
   - Make sure that your agent plays reasonably on your chosen board size. If it doesn't, stop and THINK: Is it the architecture? The encoding? Some unique feature of your game? How can you help the agent? You may need to apply some tricks. For example, manually selected features.
2. **Running the Training Loop**
   - Implement the self-play training loop for the PUCT agent.
   - Recall the difference in AZ between train and test: The added noise (if you choose to add it), and sampling the moves using the visit counts rather than choosing the best move.
   - Make sure you have enough computational resources. This part is computationally expensive because if you have (say) 1000 PUCT iterations per move, you have 1000 forward passes for each piece of data. I recommend using the college cluster.

3. **Evaluating Your Agent**

   - Track performance by playing your agent yourself, as well as having the agent play games against previous versions.
   - (Optional) Use the **ELO rating system**:

     $$E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}}$$

     where R_A and R_B are the ratings of two agents. Get chatGPT to explain it to you. Basically, each player has a rating and when a player wins/loses a game, their rating goes up or down. In this way, you can meaningfully assess your agent's strength.