Introduction

- Deep reinforcement learning has achieved superhuman, state-of-the-art performance in many perfect-information games (e.g., Chess, Go, Shogi), but imperfect-information games such as Poker are still dominated by more traditional reinforcement learning techniques (CPR).
- State-of-the-art reinforcement learning algorithms (Q-learning, Alphazero et al.) fail for imperfect-information games.
- Heinrich and Silver (2017) developed Neural Policies for Poker (NPF), which uses a Q-learning policy network to achieve stable learning.
- Our goal: adapt the Alphazero algorithm to use in imperfect-information games and apply it to two variants of poker.

The algorithm

- Single policy-value network takes the players' information set as an input and returns a policy and an estimate of the value of the position.
- Opponent range network also takes the players' information set as input and returns an estimate of the opponent's cards.
- A Monte Carlo Tree Search (MCTS) is used to estimate a best response to the current policy $p$. The player uses an upper-confidence bound (UCB) strategy guided by the policy $p$ and value estimate $v$, while the opponent uses Monte Carlo sampling of the policy $p$ given cards sampled using the opponent range network.
- Games are mostly played using the policy $p$, with only a small fraction ($\alpha = 0.1$) using the best response found via the tree search (antiparametric learning).
- The policy $p$ is trained to approximate the average of all best response policies from the entirety of training. However, sampling is used when the data exceeds the memory buffer.
- The value estimate $v$ and opponent range estimate are trained using a smaller dataset of all recent games (regardless of which policy was used) to approximate the actual outcomes of games and opponent cards respectively.
- Cross entropy loss is used for policy and opponent range. L2 loss is used for value estimates. L2 regularization.
- The temperature of the best response policy $\pi^*(x)$, where $x$ is the number of times action $a$ was taken in information set $x$ in the tree search, was gradually decreased over time to speed up learning. To avoid biasing the best response policy in favour of action with higher policy probability, the root node initial policy was replaced by $p$.

Results

- Performance at Leduc Poker with the MCTS temperature parameter set to $T = 1$ (left) and $T = 0$ (right).
- Zero exploitable corresponds to the Nash equilibrium solution.
- Preflop strategy for Heads-Up Limit Texas Hold’em after 500k games. The upper right triangle shows the strategy for suited cards, the lower left triangle shows the strategy for unsuited cards.

Discussion

- Our algorithm successfully converged to an approximate Nash equilibrium in Leduc Poker. It provides an alternative approach to NPF for deep reinforcement learning in imperfect-information games.
- The performance at Leduc Poker was not quite as high as for NPF. However, as we have optimized hyperparameters and algorithmic details, the performance of our algorithm has steadily improved. We are confident that further performance gains are possible.
- MCTS algorithms are most effective for very deep games. This suggests our algorithm should be even more useful in ‘real’ games, and especially in even deeper imperfect-information games such as various board games.
- Our algorithm appears to be learning Heads-Up Limit Texas Hold’em successfully but we were limited in our ability to measure performance, because of the time required to simulate sufficient games given our limited computational power.

Future

- Future plans include eliminating the optimistic performance on Leduc poker. Just in the last few days, various optimizations have improved the exploitability achieved by a factor of two.
- We will also look to see our algorithm’s performance at Limit Texas Hold’em against state-of-the-art poker bots.
- We will apply our algorithm to Heads-Up No-Limit Texas Hold’em, where performance improvements have only been achieved in the last year, as well as poker games with more than two players, where learning still remains expensive.
- Finally, the algorithm can be easily adapted to various other imperfect-information games, particularly games where very deep tree searches are vital to success.

References