CS234 Project Proposal: ZSY

Leon Lin
Mentor: Sudarshan Seshadri email: ssesha@stanford.edu

Problem

争上游 (ZhengShangYou, or "Competition Upstream") is a Chinese card game that is part strategy, part luck. Each player is dealt about 18 cards; they get rid of cards by matching patterns; the player who gets rid of all their cards first wins. I aim to train a deep learning agent to have an above 50% win rate against humans in a 2-player version of this game.

There are three main challenges to learning this game. First, it is stochastic with a large state space. With 2 players each being dealt 18 cards, there are about 151 trillion possible initial states[1]. Second, it's partially observed. A player can only see their own cards. Third, the test environment (playing against a human) cannot be used to train it because of the volume of data required.

Data

The data will be obtained by having agents play against each other in a simulator I wrote.

Methods

I have previously attempted this but only achieved about a 30% win rate against a human player (me). It was a rather shallow network (3 layers), fully connected, and didn't use advance RL methods like fixed targets. Building upon this work, there are many improvements to be tried. In order:

Online Experience Replay: Previously, I just had it alternate between running 100,000 simulations and training. This time, I will have a replay buffer and a more active way of alternating.

CNNs: Applying a CNN will allow a network to more easily pick up on card patterns.

Monte Carlo: Each game is a fixed episode. Instead of fiddling around with epsilons, it makes much more sense to use FVMC to set values. Additionally, with partial observability, the problem is certainly not Markov.

Automatic Selection and Battling: Previously, I started many networks independently with different hyperparameters and trained them all in tandem, manually getting rid of the worst performing ones as time went on. This was highly inefficient and didn't utilize the antagonistic nature of the game. I can instead create many networks in parallel, and have them play each other to generate data for them all to use. This also creates a different method of exploration, with different networks making different choices.

Literature

I will carefully review the methods of the DQN Atari papers, as well as the three papers mentioned in class on double DQN, prioritized replay, and dueling DQN

Evaluation

Initially, the agent can only be evaluated against other agents. The first benchmark is to beat a greedy agent; the second is to beat the network from the last attempt. Afterwards, I plan to test it against many humans. I have previously made a graphical version of the game for mobile, but it couldn't collect data. I plan remake that in Unity but with additional data collection methods and send it to people to play against.

[1] Exactly 151,632,049,354,500, calculated recursively here.