Case study 4 assignment

1. AlphaGo uses a single reward at the end of the game (+1 for a win, 0 for a loss). Why did the engineers choose this strategy rather than reward good moves throughout the game? Is there any disadvantage to deferring all reward until the game finishes?

The ultimate goal of the strategy is to win the game, no matter how many points it wins. So this single reward strategy is very intuitive, which could avoid being trapped in local optima when applying a greedy strategy for each step.

Since this strategy doesn't guarantee every single step is locally optimal, it could produce unnecessary or slack moves during the game, which seems to be less efficient.

2. We could also defer all reward for our Gridworld games by giving +1 when the agent reaches the finish square and 0 for all other moves. Why did we choose to give a reward of −1 for each step instead? Why is this not a concern for AlphaGo?

Because the goal of the Gridworld game is to learn the shortest trajectory rather than simply go to the target destination. It needs the reward for each move to estimate a value function to get the optimal moving policy. If only given the reward in the end, each trajectory will have the same reward, therefore no difference for the shortest trajectory.

In AlphaGo, the goal is to win the game in the end, rather than to find the shortest trajectory.

3. Games like Go are called “perfect information” games. What does “perfect information” mean? Are perfect information games easier or harder for RL agents to solve? Do biological experiments have perfect information?

Perfect information game means each player has the same information during the entire process of gaming. Each player knows exactly what’s other player’s moves and current status.

It would be easier for RL, because it could reduce the search space greatly for searching potential status configurations in the future.

Biological experiments usually don’t have perfect information. There are a lot of internal variations in the material used for an experiment that cannot be predicted and inspected explicitly. They have hidden information.

4. AlphaGo has three parts: 1.) policy neural network that recommends the next action for each state; 2.) a value neural network that predicts the probability of victory given each state; and 3.) a local search algorithm that plays ahead 50-60 moves using the policy and value networks. Our Gridworld agent did not include a local search feature. Why do games like Go require playing ahead to find good moves while Gridworld does not?

For Go, it's impossible to calculate all the variations of outcomes from the current position using an exhaustive search strategy. It needs the local search algorithm to truncate the search tree and roughly compute the winner by running the rollouts to the end. But for the Gridworld game, the game complexity isn't that high. It just takes several steps from any position to reach the end so that the local search algorithm is unnecessary.

5. With Gridworld we demonstrated how policy improvement can start with a random policy and iteratively find the optimal policy. AlphaGo was “bootstrapped” by extracting a starting policy from online games. What is the advantage of starting with a bootstrapped policy?

By training the policy network with data of online games, the initial policy simulates how human moves under specific configuration. Its performance is much better than the initial policy generated randomly. Therefore, the bootstrapped policy could reduce large amounts of training time to optimize the policy network and find a good value function via reinforcement learning.

6. DeepMind eventually built AlphaZero, an agent that learned to play Go (and chess and shogi) without bootstrapping. AlphaZero quickly learned to beat AlphaGo. Can you guess any reason why AlphaZero became superior to AlphaGo?

The policy network in AlphaGo is trained with expert data which is relatively small-scale. So the policy still simulates human behavior to some degree. AlphaZero is trained directly through self-play by only providing the perfect knowledge of the game rules. This improvement could get rid of the constraints of prior human knowledge, enabling more candidate points to search for the optimal strategy. Besides, the optimization of network architecture and reinforcement policy should also play a critical role in achieving the superity.