CS 478 Project Proposal: Imperfect-Information Reinforcement Learning

Tyler Etchart

I. Project Description

Reinforcement learning has become a big area of study in modern deep learning. It was mostly popularized by Google Deepmind's work with the Deep Q Network and its application to the Atari games [2]. Since then, researchers have been using reinforcement learning to advance many fields. I'm specifically interested in how reinforcement learning can be applied in the field of imperfect-information games. The most recent example of which is the success of Libratus in Pittsburgh [1]. Libratus was an AI that was developed by Tuomas Sandholm and Noam Brown at Carnegie Mellon. Its goal was to beat the best Poker players at No-limit Texas Hold'em. It succeeded by a collective $1.7M against 4 of the most talented players in the world. I propose developing a RL agent that will learn to master an environment I will call the "Coin Toss" 1.

II. Learning Environment

In order to have a proper introduction into deep learning with imperfect-information games, the learning environment needs to be simple. We don't want to spend an inordinate amount of time engineering the environment and the APIs. We want to focus our efforts more on building and experimenting with the RL agent itself. We will use one of the most basic imperfect-information games, the "Coin Game". It is a simple adversarial game that was used in the paper "Safe and Nested Endgame Solving for Imperfect-Information Games" [4]. There are two players. Player 1 flips a coin and gets either heads or tails. He can then decide whether to keep the toss to himself and receive a hidden reward (-0.5 for tails and +0.5 for heads), or he can let Player 2 guess what the outcome was and receive a reward according to Player 2's choice. If Player 2 guesses the toss correctly (i.e. the coin landed on heads and Player 2 guessed heads), then Player 1 will get -1.0 reward. If Player 2 guesses incorrectly or forfeits the round, Player 1 will get +1.0 reward. Player 1's objective is to maximize his reward. Player 2's objective is to minimize Player 1's reward. The winner is determined after a specified number of rounds is played.

III. Agent

We will be writing one agent architecture for this project with 2 different control APIs. The agent will have a control API that corresponds to Player 1 and one that corresponds to Player 2. We will use the same architecture for both versions of the agent (with the exception of the output layer that will connect to the control API) to force the architecture to be capable of more general learning. We will write the agent as a Deep Q Network with the Tensorflow library. We will implement check points so that we can save the agent's weights and later benchmark it against random players and human players.

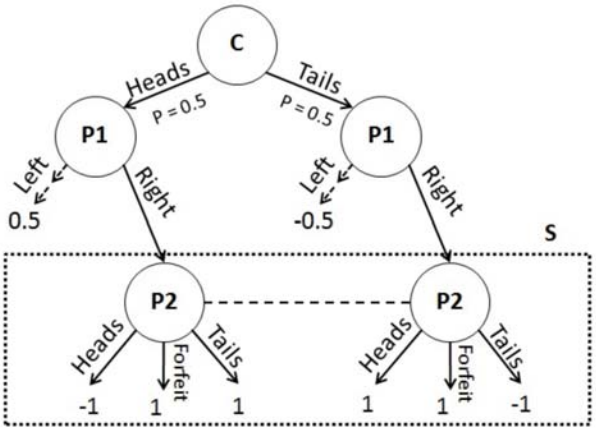

Fig. 1. Coin Toss

IV. Reach Goals

If the project appears to be going well and there is still time in the semester, a possible reach goal would be to pit the DQN agent against a Bayesian agent. The Bayesian agent will use black box variational inference [3]. It would be a powerful benchmark to test our DQN against a more traditional machine learning Bayesian method to find out if modern techniques are any better than the tried-and-true statistical methods.

References

[1] Cade Metz. Inside libratus, the poker ai that out-bluffed the best humans, 2017.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

[3] Rajesh Ranganath, Sean Gerrish, and David M. Blei. Black Box Variational Inference. December 2013.

[4] Noam Brown Tuomas Sandholm. Safe and nested endgame solving for imperfect-information games. 2017.

I WOULD PUT IN A LOT OF EFFORT FOR THIS PROJECT.