Machine Learning Project: Report 2

Ignace Bleukx Quinten Bruynseraede

May 3, 2020

1 Introduction

1.1 Evaluation metrics

Evaluation of agents is traditionally done using **NashConv** and **exploitability**. We introduce these concepts here, using terminology consistent with

Given a policy π,

1.2 Algorithm 1: Fictitious Self-Play

In this algorithm the agent plays against itself and tries to improve its strategy by exploiting its own weaknesses. The variant of this algorithm implemented in Open Spiel is the Extensive-Form Fictitious Self-Play algorithm (XFP). In this algorithm the agent performs two steps at every iteration [2]. In the first step, the best response to the current policy π is generated, in the second step the policy is updated based on the best response.
To find the best reponse to the current strategy, the game state tree is traverse and the set of best responsed is calculated. Out of this set, a best reponse is selected. As the agent plays against itself, the game state tree is known as the probabilities for all actions in all states are known.
The seconds step is to update the average policy by using the calculated best response. The update rule is the following: $\pi_{i+1}(u) = \pi_i(u) + \frac{\alpha_{i+1} x_{\beta_{i+1}}(\sigma_u)(\beta_{i+1}(u) - \pi_i(u))}{(1-\alpha_{i+1}) x_{\pi_i}(\sigma_u) + \alpha_{i+1} x_{\beta_{i+1}}(\sigma_u)}$.
Where $\pi_n(u)$ indicates the action probabilities of the policy at iteration n, β_n the best reponse calcuated in step 1 and σu a sequence of actions to reach state u. $x\beta(\sigma_u)$ is the realization plan for this sequence, given best response β.
The main issue with this form of fictitous self-play is the computational effort needed to calculate the best response. Therefore, many extensions and variations exist to effeciently calculate this best response and update the policy. As [2] points out, the FSP framework consists of machine learning approches to compute both the best response, as well as update the policy. In the next section, this is done by ANN's.

1.2.1 Extension: Neural Fictitious Self-Play

In the NFSP algorithm, the steps described above are approximated by using two neural networks.

Best response As stated in the previous section, the FSP framework consists of two steps, the first of which is calculating the best response for a given state. In the NFSP setting this step is done by training a Deep Q-network or DQN. This agent learns an ϵ-greedy policy [1]. This can be done as the games considered have perfect recall and the history of the game is thus available to the agents. By using this history the agent gains experience and approximates the best response for a given state. The implementation of NFSP in OpenSpiel uses a multilayer perceptron network (MLP).

Average Strategy To compute the action probabilities for a given state in the game, NFSP trains another ANN which maps these state to the corresponding probabilities. This network is trained by using information of past actions performed. As the average policy is approximated, the network does not need to be consulted every iteration of the learning process. This means the update can be much more effecient when using a good ANN. In the OpenSpiel implementation of NFSP this is an MLP.

1.3 Algorithm 2: Counterfactual Regret Minimization

1.3.1 Extension: Regression Counterfactual Regret Minimization

1.3.2 Extension: Counterfactual Regret Minimization against best responder

1.3.3 Extension: Deep Counterfactual Regret Minimization

2 Kuhn Poker

- Which algorithm is most suitable to develop an agent to play Kuhn Poker, minimizing exploitability?
- Can we exploit properties of Kuhn Poker to optimize parameters?

3 Leduc Poker

- Which algorithm is most suitable to develop an agent to play Leduc Poker, minimizing exploitability?
- Can we exploit properties of Leduc Poker to optimize parameters?
- Can we combine agents into an ensemble that minimizes exploitability further than its parts?

References

[1] Johannes Heinrich and David Silver. Deep reinforcement learning from self-play in imperfect-information games. 2016.

[2] Johannes Heinrich, Marc Lanctot, and David Silver. Fictitious self-play in extensive-form games, 2016.

Appendix

3.1 Time spent