Solving Hanabi with Neural Network

Ciruzzi Michele

December 10, 2020

Abstract

Try to exploit a board game with incomplete and asymmetric information is not simple, as stated in a paper [2] that I've read too late. Nevertheless I've tried to solve it, obviously failing. This report will show (some of) the attempts that I made and some of the lesson I've learned.

1 Problem

Hanabi is a cooperative game for 2 to 5 players with incomplete and asymmetric information. For this features it has been proposed by DeepMind group [2] as a new challenge for Reinforcement learning. The problem has been tackled both with deterministic (e.g [3]) and ML-learning based algorithm (e.g. [4]). The game has been demonstrated to be NP-Hard in general [1]. As a rule of thumb a good algorithm is able to achieve a perfect score (25 points) in more than the 90% of games. My aim is to train a neural network able to realize a good strategy in playing Hanabi using Deep-Q Learning algorithm: which is exactly what Bard et al. [2] have tried to do (cfr. *Rainbow* policy in [2]) with low quality results (I've noticed this only during the last reading of the paper to write this report). The code used for the last attempts is in the zip file attached and on colab LINK, while some of the first attempts are available int he different branches of `https://github.com/TnTo/Hanabi/` (particularly v0.2, v0.3 and master).

2 Methods

Even if more sophisticated algorithms are available, I have implemented the Deep-Q-Learning algorithm introduced by Mnih et al. [5]. As far as I know, the source code used is not available and not every step of the algorithm is deeply explained, so some difference between the algorithm presented in the paper and my implementation are present (more in last section). I've implemented the algorithm from scratch using keras and numpy, which has critically improved the performance compared with a previous pure python implementation. The algorithm can be summarized as follow:

1. (Optional) Play some games performing random actions to populate the memory
2. Select a subset of the memories, calculate the target value of each action for each board state in memory (see below) and then train the network
3. Initialize a given number of new games and retrieve some board states from the memory, play these games with a given policy (see below) until they end appending to the memory each board state observed
4. repeat step 2 and 3, eventually playing some new games after the step 2 to test the ability of the network to improve the achieved score in subsequent iterations.

The target value for step 2 is

$$Q(s_t, a_t) = score(s_{t+1}|s_t, a_t) - score(s_t) + \gamma \max_{a_{t+1} \in actions} Q(s_{t+1}, a_{t+1})$$

where s_t is the board state at time t and a_t is the action performed. γ is a discount factor used to account for long-term outlook. The policy used in step 3 is the following: with probability ϵ the action is chosen

as random, otherwise is $argmax_{a_t \in actions} Q(s_t, a_t)$. In fact, the goal of learning is to approximate Q as well as possible with a neural network. For the neural network a MSE loss and a RMSProp learning has been used with different leaning rates (generally between 10^{-2} and 10^{-4}), numbers of hidden layers (from 1 to 3), number of neurons (from 32 to 1024) and activations (sigmoid, tanh, ReLU and PReLU has been tried).

3 Results

4 Conclusions

References

[1] Jean-Francois Baffier et al. "Hanabi is NP-hard, Even for Cheaters who Look at Their Cards". In: (Mar. 7, 2016). arXiv: 1603.01911v3 [cs.DM].

[2] Nolan Bard et al. "The Hanabi challenge: A new frontier for AI research". In: *Artificial Intelligence* 280 (2020), p. 103216. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2019.103216. URL: http://www.sciencedirect.com/science/article/pii/S0004370219300116.

[3] Christopher Cox et al. "How to Make the Perfect Fireworks Display: Two Strategies forHanabi". In: *Mathematics Magazine* 88.5 (Dec. 2015), pp. 323–336. DOI: 10.4169/math.mag.88.5.323.

[4] Adam Lerer et al. "Improving Policies via Search in Cooperative Partially Observable Games". In: *AAAI 2020* (Dec. 5, 2019). arXiv: 1912.02318v1 [cs.AI].

[5] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (Feb. 2015), pp. 529–533. DOI: 10.1038/nature14236.