Yanglegeyang Agent based on reinforcement learning

Yushen Li, Jenwei Hsueh

INTRODUCTION

Yanglegeyang is an explosive elimination game, which mainly allows players to use various props and hints to eliminate obstacles and traps in each level.Reinforcement learning emphasizes on maximizing the expected benefits by continuously trying in the environment. Nowadays, the more common reinforcement learning methods are Value-based and Policy-based. The representative algorithms of Value-based include Q-learning [1] and its derivative algorithm DQN [2]. therepresentative algorithms of Policy-based include Policy Gradient [3] and PPO [4].We hope that we can apply and improve some of the above methods to implement an non-invasive agent for YanglegeYang.

We divided the implementation process into two stages:

1. Make the agent correctly recognize both dark and bright game cards by CV method, and when part of the cards were covered, it should also be correctly identified.
2. Use algorithms to make the agent to eliminate as many cards as possible, learn the game and achieve at least the same level as humans.

Methods

Recognize the Card

Take a screenshot of 16 different cards in the game as a template.

Detect the card in all colors for [147,153,123] (background clour) of the component and the component of matrix converter. For components, we consider their external matrix to be the card we recognize.

For each card, we use the method of ssmi to match the template one by one, believing that the most matching template is the label corresponding to the card face.

Policy-Based Agent

We define the state as three sets, namely the light set, the dark set, and the set of cards in the queue pool. The reward is set to 1 if 3 cards are eliminated, otherwise it stays 0.

PPO algorithm is used to train agents. In particular, the action space of input policy-net needs to be consistent.

Prior Knowledge

Obviously, one type of card is unlikely to outnumber the others, and in a game, the next card in the flip decides a great deal on success.

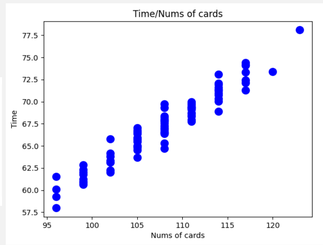
We counted all the cards we could see in 100 games, and found that the distribution of cards was generally fairly uniform.So every time a card appears, we reduce the confidence of that type of card's next appearance by a little bit. Treat this prior knowledge as part of the environment.

RESULT

We played 100 games with trained agents and recorded their performance and time. Compared with human players, it can be seen from the figure below that the agent is much better than human players in time on the basis of eliminating the similar number of cards as human players.

Player（game num）	Cards（avg）	Time/s（avg）
RL（100）	108	67
Li（10）	120	91
Hsueh（10）	99	151
Anonymous（10）	96	130

Compared with human



Scatter(Times/Cards)

REFERENCES

[1] Watkins C J C H, Dayan P. Q-learning[J]. Machine learning, 1992, 8(3):279-292.

[2] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deepreinforcement learning[J]. arXiv preprint arXiv:1312.5602, 2013.

[3] Silver D, Lever G, Heess N, et al. Deterministic policy gradientalgorithms[C]//International conference on machine learning. PMLR,2014: 387-395.

[4] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimizationalgorithms[J]. arXiv preprint arXiv:1707.06347, 2017.

[5] Babaeizadeh M, Frosio I, Tyree S, et al. GA3C: GPU-based A3C fordeep reinforcement learning[J]. CoRR abs/1611.06256, 2016.

[6] Mnih V, Badia A P, Mirza M, et al. Asynchronous methods fordeep reinforcement learning[C]//International conference on machinelearning. PMLR, 2016: 1928-1937.

[7] Anonymousplendid. https://github.com/Anonymousplendid/Yanglegeyang