

卒業研究報告書

題 目

深層強化学習に基づく
トレーディングカードゲーム環境の構築

研究グループ 第1研究グループ

指導教員 森 直樹 教授

令和4年（2022年）度卒業

(No. 1201201100) 西村 昭賢

大阪府立大学工学域電気電子系学類情報工学課程

目次

1	はじめに	1
2	要素技術	2
2.1	OpenAI Gym	2
2.2	Q 学習	2
2.3	Deep Q Network	3
2.4	Genetic Algorithm	4
2.5	Nondominated Sorting Genetic Algorithm II	5
3	提案手法	6
3.1	関連研究	6
3.2	深層強化学習によるデッキ内のカードパワーの測定	6
3.3	解空間の次元を削減した最適化	6
4	実験方法	7
5	結果と考察	8
6	まとめと今後の課題	9
7	結論	10
	謝辞	11
	参考文献	12

図目次

2.1 OpenAI Gym が提供している様々な環境	2
2.2 DQN のアルゴリズムの疑似コード ^[1]	4

表 目 次

1 はじめに

近年, 人工知能に関する研究分野は目覚ましい発展を遂げており様々な分野に応用されている. その中でも人間の学習プロセスに近いとされる強化学習と深層学習を融合した深層強化学習は自動運転やロボット, 推薦システム等の実生活の問題解決への応用例が数多く報告されている [2][3][4]. 実世界の問題解決への応用だけでなく, 深層強化学習はゲームへの応用も盛んである. 特に将棋や囲碁といった, プレイヤーが意思決定をする段階でそれ以前の意思決定の過程がすべて把握可能な完全情報ゲームへの応用においては AlphaGo [5], AlphaZero [6] を筆頭に現役のプロプレイヤーを圧倒する性能を残しており成果が顕著である. 最近では麻雀やポーカーのような, プレイヤーに与えられる情報が部分的である不完全情報ゲームへの応用も注目されている.

本研究では不完全情報ゲームであるトレーディングカードゲーム環境への深層強化学習の適用と深層強化学習を用いたゲームバランス調整手法を提案し, 独自に構築したトレーディングカードゲーム環境を用いて数値実験することでその有効性を示す.

以下に本論文の構成を示す. まず, 2 章では本研究で用いる要素技術について, 3 章では類似研究と提案手法について説明する. 4 章では本環境で独自に構築したトレーディングカードゲーム環境の概要を示す. 5 章で実験方法の説明をし, 6 章で実験結果と考察を示す. そして, 7 章に本研究のまとめ及び今後の課題について述べる.

2 要素技術

2.1 OpenAI Gym

OpenAI Gym^[7] は非営利企業 OpenAI が提供する強化学習のシミュレーション用ライブラリであり、強化学習の環境として図 2.1 に示すように多くのゲーム、シミュレータが登録されている。さらには提供されているインターフェースに沿って、エージェントの行動空間や状態空間、報酬などを定義、実装することで自作の強化学習環境を構築し利用することができる。様々な強化学習用ライブラリに対応しているため比較的容易に強化学習を試すことができる。

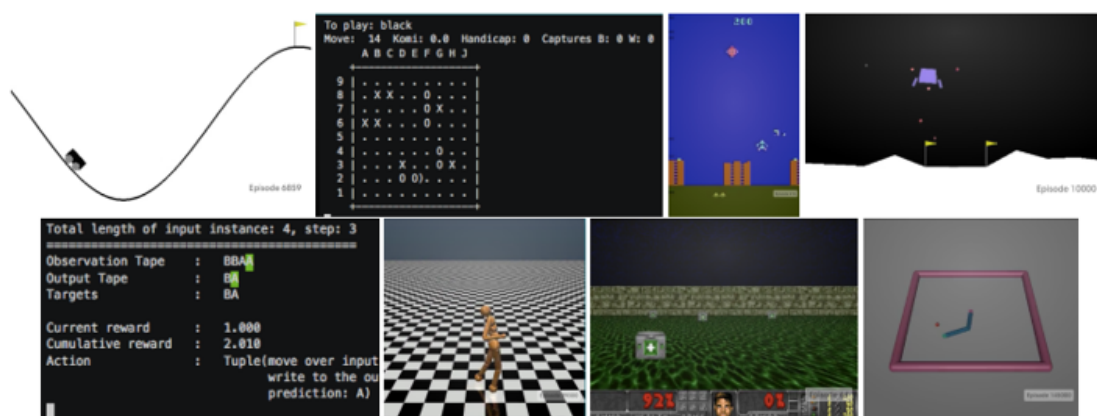


図 2.1: OpenAI Gym が提供している様々な環境

2.2 Q 学習

強化学習では、エージェントが行動することで環境における状態が変化し報酬を得る。強化学習における行動はその直後に獲得する報酬の大きさではなく、未来に渡っての報酬の総和を見積もった値である「価値」の最大化に繋がるかという観点で評価される。価値の最大化を目指す場合にはある状態 s において行動 a をとった時の価値が分かればよい。この価値のことを Q 値、あるいは行動価値関数と呼ぶ。この Q 値を基に行動を選択していく価値ベースの強化学習手法において代表的な手法が Q 学習である。 Q 学習ではエージェントの 1

ステップごとに (2.1) 式に示す更新式で Q 値を更新する.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)) \quad (2.1)$$

なお t は時間, r は報酬, α は Q 値の更新度合いを表す学習率, γ は将来の価値の割引度合いを表す割引率である.

また, Q 学習に代表される強化学習においては環境の調査を目的とする探索と, 探索により得られた良い報酬を得ることができる経験の活用をそれぞれの程度にすればよいかといういわゆる「探索と活用のトレードオフ」という問題が発生する. これを解決する手法として一般的なものが ϵ -greedy 法である. ϵ -greedy 法では確率 ϵ でランダムに行動し探索, それ以外では経験を活用して価値が最も高い行動を選択することで探索と活用のバランスをとっている.

2.3 Deep Q Network

Q 学習を実際に実装する場合, 状態と行動をインデックスとした Q 値のテーブルを作成する. しかし状態空間や行動空間が高次元である, あるいは状態や行動が離散値ではなく連続値で表現される場合には Q テーブルのメモリ量は爆発してしまう. この問題を解決した技術が Deep Q Network (DQN) [8] である. DQN ではニューラルネットワークを用いて, ある状態における行動ごとの Q 値を推定することでたとえば状態が連続値であっても学習可能としている. DQN では, エージェントが経験した過去の体験を Replay Memory に一定期間保存しておき, 過去の経験をランダムにサンプリングして学習する Experience Replay や行動を決定する Q 値のネットワークと Q 値を学習するネットワークを分けることで Q 値の過大評価を防ぐ Fixed Target Network といった工夫により安定した学習を可能としている.

Algorithm 1 deep Q-learning with experience replay

```

1: Initialize replay memory  $D$  to capacity  $N$ 
2: Initialize action-value function  $Q$  with random weights  $\theta$ 
3: Initialize target action-value function  $\hat{Q}$  with weights  $\theta^- = \theta$ 
4: for episode = 1,  $M$  do
5:   Initialize sequence  $s_1 = \{x_1\}$  and preprocessed sequence  $\phi_1 = \phi(s_1)$ 
6:   for  $t = 1, T$  do
7:     With probability  $\epsilon$  select a random action  $a_t$ 
8:     otherwise select  $a_t = \operatorname{argmax}_a Q(\phi(s_t), a; \theta)$ 
9:     Execute action  $a_t$  in emulator and observe reward  $r_t$  and image  $x_{t+1}$ 
10:    Set  $s_{t+1} = s_t, a_t, x_{t+1}$  and  $\phi_{t+1} = \phi(s_{t+1})$ 
11:    Store transition  $(\phi_t, a_t, r_t, \phi_{t+1})$  in  $D$ 
12:    Sample random minibatch of transitions  $(\phi_j, a_j, r_j, \phi_{j+1})$  from  $D$ 
13:
14:    Set  $y_j = \begin{cases} r_j & \text{if episode terminates at step } j+1 \\ r_j + \gamma \max_{a'} \hat{Q}(\phi_{j+1}, a'; \theta^-) & \text{otherwise} \end{cases}$ 
15:    Perform a gradient descent step on  $(y_j - Q(\phi_j, a_j; \theta))^2$  with respect to the network parameters  $\theta$ 
16:    Every  $C$  steps reset  $\hat{Q} = Q$ 
17:   end for
18: end for

```

図 2.2: DQN のアルゴリズムの疑似コード [1]

2.4 Genetic Algorithm

Genetic Algorithm (GA) とは、生物の進化と進化の過程を模した最適化手法であり、主に組み合わせ最適化問題に対して適用される。GA では 1 つの解を 1 つの個体として表現し、多数の個体からなる個体群を用いて解空間の多点を同時に探索する。各個体はどの程度良い解であるかという指標として適用度を持つ。また、各個体は染色体と呼ばれる配列で表される。この染色体を構成する要素を遺伝子、染色体上の遺伝子が収まる座標を遺伝子座と呼ぶ。探索に

においては、個体群に対して選択、交叉、突然変異と呼ばれる 3 種類の遺伝演算子を適用させ世代と呼ばれる探索ステップを進めていく。

選択では、探索において適用度が良好な個体が存在する部分を重点化するように現在の個体群から個体を選び、次世代の個体群を生成する。選択方法としては、個体の適応度に比例する確率で個体を選択するルーレット戦略、個体群からランダムに数個個体を抽出しその中で最も良好な個体を選択するトーナメント戦略がある。またこれらと併用される戦略として各世代の最良個体を保存するエリート保存戦略がある。エリート保存戦略により探索で発見した良い個体が失われることを防ぐことができる。交叉では、2 つの個体からそれらの形質を受け継いだ新たな個体を生成する。交叉においても染色体のランダムな 1 点で染色体を切断し部分列を染色体同士で交換する 1 点交叉、ランダムな 2 点で切断する 2 点交叉、遺伝子座ごとランダムに交換する一様交叉など様々な方法がある。突然変異では、各遺伝子座の遺伝子を許容された範囲の遺伝子の内容に置換する。交叉や突然変異にはそれぞれ確率が設定されていることが一般的である。

2.5 Nondominated Sorting Genetic Algorithm II

3 提案手法

本研究では, Magic : The Gathering¹ といったトレーディングカードゲーム (Trading Card Game : TCG) を参考にした.

3.1 関連研究

カードゲームにおけるバランス調整に関して, Fernando らは HearthStone² 環境内においてデッキ間の勝率が 50 % となるように, 3 つのデッキの計 64 枚のカードにおける 180 個の調整可能なパラメータを 180 個の要素を持つ 1 次元配列として GA, NSGA-II を適用した^[9]. GA を用いた場合, デッキ間の勝率をほぼ 50 % に近づけるという目標は達成したが, 180 個の調整可能パラメータの総変更量は 402 となり元のデッキの原型が無くなった. NSGA-II において勝率とパラメータの総変更量の 2 つの目的関数を最適化するように設定した場合, 勝率は GA を用いた場合とほぼ変わらず, パラメータの総変更量は 402 から 154 へと減少した.

しかし, NSGA-II を用いることで GA に比べてカードのパラメータの総変更量は減少したが, 変更を及ぼす領域は調整対象のカード全体に及んでいる. 本研究における提案手法は, 遺伝的アルゴリズムの解空間の次元を深層強化学習により削減することで, 調整を施すカードの数を減らすことを目的としている.

3.2 深層強化学習によるデッキ内のカードパワーの測定

3.3 解空間の次元を削減した最適化

¹<https://magic.wizards.com>

²<https://hearthstone.blizzard.com>

4 実験方法

5 結果と考察

6 まとめと今後の課題

7 結論

謝辞

年 月 日

参考文献

- [1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dhharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, Vol. 518, No. 7540, pp. 529–533, February 2015.
- [2] Andreas Folkers, Matthias Rick, and Christof Büskens. Controlling an autonomous vehicle with deep reinforcement learning. *CoRR*, Vol. abs/1909.12153, , 2019.
- [3] Shixiang Gu*, Ethan Holly*, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Piscataway, NJ, USA, May 2017. IEEE. *equal contribution.
- [4] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference, WWW '18*, p. 167–176, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.
- [5] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, Vol. 550, pp. 354–, October 2017.

-
- [6] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharmashan Kumar, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, Vol. 362, No. 6419, pp. 1140–1144, 2018.
 - [7] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv e-prints*, p. arXiv:1606.01540, June 2016.
 - [8] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. 2013. cite arxiv:1312.5602Comment: NIPS Deep Learning Workshop 2013.
 - [9] Fernando de Mesentier Silva, Rodrigo Canaan, Scott Lee, Matthew C. Fontaine, Julian Togelius, and Amy K. Hoover. Evolving the hearthstone meta. *CoRR*, Vol. abs/1907.01623, , 2019.