2022 年 11 月 29 日 B3 西村昭賢

進捗報告

1 今週やったこと

- 自作カードゲーム環境の改良
- モンテカルロ法の実装

2 自作カードゲーム環境の改良

2.1 カードゲームルールの変更

先週の段階ではゲームの終了条件を「お互いのプレイヤーのデッキ, 手札の両方からカードが無くなったらゲーム終了」としていた. しかし, 学習を勧めていく中で相手の盤面, 手札, デッキにカードが無いかつ自分の盤面にカードが存在し手札にカードが存在する, すなわち相手が何もできず自身が盤面有利の際に何もせずターンエンドを繰り返し盤面有利の報酬を獲得し続ける局所解に陥った. そこで, 変更した点が以下の通りである,

- プレイヤーの行動に「ターンエンド」を含んでいたが, 盤面にあるカードそれぞれについて「何もしない」という選択肢を追加しターン中に行動可能なカードが全て行動すると自動的にターンエンドするように変更.
- 終了（敗北）条件を「自盤面に盤面にカードが無いかつ手札とデッキにカードが無い場合に敗北」と変更.

2.2 報酬の変更

終了条件の変更に伴い報酬も以下のように変更した.

$$reward = 0.0 \quad 1\text{エピソード終了後}\ reward = \begin{cases} 30.0 & \text{(先述の勝利条件で勝利していた場合)} \\ -30.0 & \text{(先述の勝利条件で敗北していた場合)} \end{cases}$$

2.3 追加で改良した点

今までは先攻プレイヤーの行動しか学習できなかったが, 後攻プレイヤーの学習も行えるように変更した.

2.4 実験 1

環境がバグ無く動いているか DQN を用いて実験した. 表 1 に示すデッキにおいて α を 1 - 4 に変化させ 150000 ステップ 先攻プレイヤーの学習を行い, 10000 エピソード検証し勝率を計算した. なお, 実験で用いた DQN のパラメータは表 に示す.

結果は表のようになった.

表 3: 実験 1 の結果

α	1	2	3	4
勝率	1.0000	1.0000	0.0005	0.0000

表 1: 先攻, 後攻プレイヤーのデッキ
() 内の数字は (攻撃力 , HP) を意味している.

先攻	後攻
(3 , 3) × 5	(3 , α) × 9
(2 , 3) × 5	(1 , α) × 3
(2 , 4) × 5	(4 , α) × 3

表 2: DQN のパラメータ

方策	ε -greedy
ε	0.1
全結合層の活性化関数	ReLU
全結合層の次元	64
最適化アルゴリズム	Adam
学習率	1e-2
Experience Replay のメモリ量	1000000

2.5 実験 2

後攻プレイヤーの学習が可能になったため, 表 4 に示すデッキでステップ数を変化させ勝率を比較してみた.

表 4: 各プレイヤーのデッキ
() 内の数字は (攻撃力 , HP) を意味している.

学習するプレイヤーのデッキ	相手プレイヤーのデッキ
(3 , 3) × 5	(3 , 3) × 6
(2 , 3) × 5	(1 , 5) × 3
(2 , 4) × 5	(4 , 2) × 3
	(3 , 2) × 3

結果は表 5 に示した. 相手プレイヤーのデッキのほうがカードパワーを強く調整している実験設定であるとはいえ, ランダムな行動を取る相手に対して 5 割を切る勝率は低い. これは後述する学習エピソード数の少なさによるものと考えられる.

表 5: 実験 2 の結果

学習ステップ数	100000	150000	200000
先攻勝率	0.4723	0.4804	0.4875
後攻勝率	0.3631	0.5050	0.4196

3 モンテカルロ探索の実装

以前にアドバイスを頂いたモンテカルロ法による学習を実装した [1]. エピソードのシミュレーションにおいて行動選択の方策として ϵ - greedy 法を採用し, Q 値の更新は学習率 α , 割引率 γ , エピソードから得られた

割引現在価値 G を用いて以下の式に従った.

$$\mathrm{Q}(s,a) = \mathrm{Q}(s,a)(1-\alpha) + \alpha * G$$

先攻プレイヤーの学習を行い, 1000000 エピソード学習し, 10000 回検証した結果 0.8011 という高い勝率を叩き出した. 図 1 に学習中における 10000 エピソード中の獲得報酬平均の推移を示す. 図 1 からわかるように獲得報酬平均が徐々に大きくなるように学習が進んでいることが分かる. DQN を用いた場合より遥かに高い勝率を叩き出した理由としては学習エピソード数の違いであると考えられる. DQN の学習は時間がかかるため 150000 ステップ, 約 4000 弱のエピソード数で実験を行っていた. DQN においても 1000000 エピソードくらいの規模感で学習を行うとどうなるか興味深い.

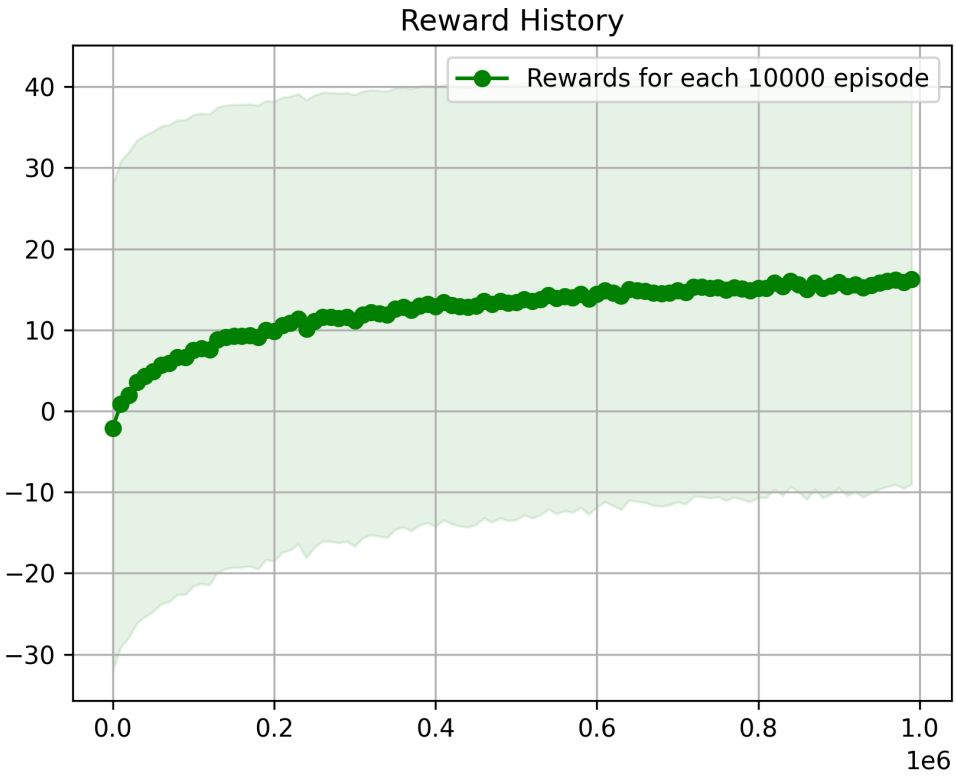

図 1: ステップ数 10000 の際の推移

4 今後の課題

今週の実験で大きなバグも現時点では見つからず, 1e6 程度のオーダーのエピソード数学習を行えばランダムに行動する相手に対して高い勝率を誇る AI を作成できることが分かった. 研究テーマが動的な難易度調整であるため, その方法とどのようなデータを取れば本研究の有効性や新規性が示せるかの検討をしなければならない.

動的難易度調整の研究は図 2 に示すフローモデルに基づいていることが多い. フローとはユーザがある活動を実行しているときに，タスク実行中に没頭し，集中し, 満たされた気持ちになる精神状態のことである. プレイヤーをフロー内に維持し、退屈（課題が全くない）やフラストレーション（課題が難しすぎる）に達するのを避けるために、ゲームのレベルをコントロールすることが主な目的である [2]. 難易度調整の際にはその都度プレイヤーが直面している状況の難易度を定義する必要があり, ヒューリスティックな知識を用いて評価

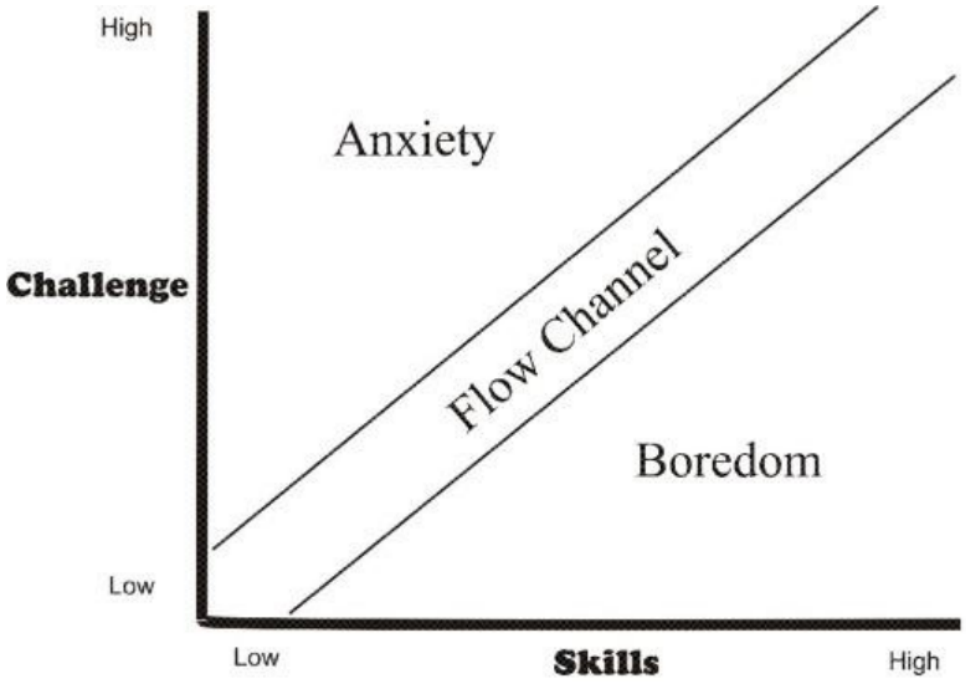

図 2: フローモデル

関数を作成するか, プレイヤーのボディランゲージといったプレイヤーの人間の行動から判断するかの 2 種類がある [2].

ターン制のカードゲームの場合はどのような評価関数でプレイヤーが直面している難易度というものを定義すればよいのか, また敵カードの HP, 攻撃力, プレイする枚数といったパラメータの中でどれを調整すればよいのか検討する必要がある.

参考文献

[1] 久保隆宏. Python で学ぶ強化学習 [改訂第 2 版] 入門から実践まで. 講談社, 2019.

[2] Mirna Paula Silva, Victor do Nascimento Silva, and Luiz Chaimowicz. Dynamic difficulty adjustment through an adaptive ai. In *2015 14th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, pp. 173–182, 2015.