2022 年 12 月 20 日 B3 西村昭賢

進捗報告

1 今週やったこと

- 研究発表会の準備 (スライド, 資料)
  資料とスライドに関して修正, アドバイス頂いた方々ありがとうございました.
- 実際の TCG ルールに寄せた環境におけるルールベースで作成した敵に対するエージェントの学習実験

2 先週までの進捗とステップを増やした場合の実験

先週は以下に示すようにプレイヤーの HP , マナコスト, カードの特殊効果など実際の TCG ルールに寄せた環境を作成し実験した.

- プレイヤー
  - HP
    最大 20, 0 となればゲーム敗北
  - マナコスト
    ゲーム開始時 1 , 最大 5 , ターンごとに 1 増加
  - ライブラリ
    ライブラリは 30 枚のカードを持つ
- カード
  - コスト
    盤面にプレイする際にカードのコスト分プレイヤーのマナコスト減少
  - 特殊効果
    - 盤面に出したら (攻撃力 , HP) = (1 , 1) のユニット追加で出す. (召喚)
    - 盤面に出したら自プレイヤーの HP を 2 回復 (治癒)
    - 盤面に出したら敵プレイヤーの HP を 2 削る (攻撃)
    - 盤面に出したら自プレイヤーは 1 枚カードをドロー (循環)
    - 盤面に出たターンに攻撃できる (速攻)
- 終了条件
  どちらかのプレイヤーの体力が 0 以下となった , または デッキ切れの状態でドローしようとした時

2.1 実験条件

お互いのライブラリは等しくした. 表 1 にライブラリの内容を示す.

また, ゲームバランス調整のためのシミュレーションのために学習したエージェントを用いる予定のため学習相手にはルールベースで作成した好戦的な行動ルーチンを組んだ. アルゴリズム 1 に作成した対戦相手の行動ルーチンを示す.

表 1: ライブラリの内容

攻撃力	HP	コスト	特殊効果	枚数
1	1	0	無し	2
2	1	1	無し	2
3	2	2	無し	2
4	3	3	無し	2
5	4	4	無し	2
2	2	2	召喚	2
2	3	3	召喚	2
1	1	1	循環	2
1	3	2	循環	2
2	1	2	速攻	2
3	1	3	速攻	2
1	2	2	攻撃	2
2	3	3	攻撃	2
1	1	1	治癒	2
2	1	3	治癒	2

表 2: 定義した状態空間

状態説明	次元数	最小値	最大値
自, 敵プレイヤーの HP	2	0	20
自, 敵プレイヤーのコスト	2	0	5
手札 1 ~9 の HP と攻撃力	18	0	20
手札 1 ~ 9 のコスト	9	0	5
手札 1 ~ 9 の特殊効果	9	0	5
自盤面 1 ~ 5 の HP と攻撃力	10	0	20
敵盤面 1 ~ 5 の HP と攻撃力	10	0	20
自盤面 1 ~ 5 がターン中行動可能かどうか	5	0	1
お互いのライブラリの残り枚数	2	0	15

また, 表 2 , 3 に定義し直した状態空間, 行動空間を示す.
報酬は以下のように定義した.

$$reward = 0.0, \quad 1\text{エピソード終了後}\, reward = \begin{cases} 1.0 & \text{(学習プレイヤーの勝利)} \\ -1.0 & \text{(敵プレイヤーの勝利)} \end{cases}$$

2.2 実験 1

先週は 先攻側で DQN を用いて 1000000 ステップ実験を行い, 10000 回ゲームを実行し算出された勝率は 0.1461 とかなり低い数値だった. そこで, ステップ数の問題かどうか調べるため学習ステップを 10 倍の 10000000 ステップに増やして実験した.

Algorithm 1 敵の行動

```
for 手札のカード do
  if 盤面にプレイできる then
    カードをプレイ
  else
    pass
  end if
end for
for 自盤面のカード do
  if 敵の盤面に 1 回の攻撃で倒せるカードがある then
    そのカードを選んで攻撃
  else
    敵プレイヤーを攻撃
  end if
end for
```

表 3: 定義した行動空間

行動説明	次元数
手札 1 ∼ 9 を自盤面に出す	9
手札 1 ∼ 9 を自盤面に出さない	9
自盤面 1 が敵盤面 1 ∼ 5 に攻撃 or 何もしない or 敵プレイヤーに攻撃	7
自盤面 2 が敵盤面 1 ∼ 5 に攻撃 or 何もしない or 敵プレイヤーに攻撃	7
自盤面 3 が敵盤面 1 ∼ 5 に攻撃 or 何もしない or 敵プレイヤーに攻撃	7
自盤面 4 が敵盤面 1 ∼ 5 に攻撃 or 何もしない or 敵プレイヤーに攻撃	7
自盤面 5 が敵盤面 1 ∼ 5 に攻撃 or 何もしない or 敵プレイヤーに攻撃	7

2.3 実験 1 結果

図 1 に実験 1 における学習時の獲得報酬の平均の推移を示す.

200 エピソードにおける平均獲得報酬が -0.9 付近で安定しており学習が進んでいないことが判明した. 勝率は記録していませんでした. 申し訳ありません.

3 実験 2

実験 1 の結果を受けてエージェントの行動空間に 2 種類変更を施して学習 → 勝率計算の流れで実験した. また, 共通して DQN における $\epsilon-$greedy に改良を施した.

- 変更前

$$\epsilon = \max(\epsilon_{\min} \quad , \quad -\frac{\epsilon_{\max}-\epsilon_{\min}}{\text{stepnum}}(stepcount)+\epsilon_{\max})$$

  学習時に, *stepcount* が stepnum ステップに達するまで $\epsilon_{\max}$ から $\epsilon_{\min}$ へと線形的に減少する.

- 変更後

$$\epsilon = \max(\epsilon_{\min} \quad , \quad \epsilon_{\min}+(\epsilon_{\max}-\epsilon_{\min})\exp(-\frac{stepcount}{\epsilon_{\text{decay}}}))$$

  学習時に, ϵ_{decay} に応じて, $\epsilon_{\max}$ から $\epsilon_{\min}$ へと指数的に減少する.

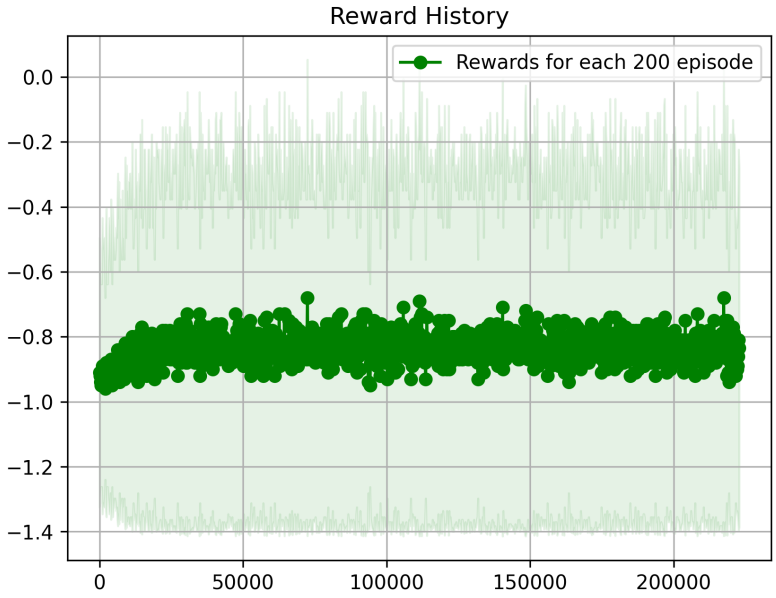

図 1: 実験 1 における獲得報酬平均の推移

表 4 にこの方策に基づいた実験 2 における DQN のパラメータを示す.

表 4: DQN のパラメータ

方策	ε -greedy
$\epsilon_{\max}$	1.0
$\epsilon_{\min}$	0.05
ϵ_{decay}	学習ステップ数 / 10.0
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
Target Network 更新重み	0.5
Exprience Memory への書き込み開始 step	10000
Experience Replay のメモリ量	50000

3.1 実験 2 - 1

盤面にあるカードに対して「相手プレイヤーに攻撃」という選択肢があるにもかかわらず「何もしない」という選択肢を選ぶのは特にメリットがないと感じたため表 5 のように行動空間を変更した.

この条件下で先攻側を 3000000 ステップ学習し, 10000 回ゲームを実行し勝率を計算した.

3.2 実験 2 - 1 結果

図 2 に実験 2 - 1 における学習時の 500 エピソードの平均獲得報酬の推移を示す.

また, 表 6 に勝率を示す.

表 5: 実験 2 - 1 で定義した行動空間 (太字は変更した箇所)

行動説明	次元数
手札 1 ～ 9 を自盤面に出す	9
手札 1 ～ 9 を自盤面に出さない	9
自盤面 1 が敵盤面 1 ～ 5 に攻撃 or 敵プレイヤーに攻撃	**6**
自盤面 2 が敵盤面 1 ～ 5 に攻撃 or 敵プレイヤーに攻撃	**6**
自盤面 3 が敵盤面 1 ～ 5 に攻撃 or 敵プレイヤーに攻撃	**6**
自盤面 4 が敵盤面 1 ～ 5 に攻撃 or 敵プレイヤーに攻撃	**6**
自盤面 5 が敵盤面 1 ～ 5 に攻撃 or 敵プレイヤーに攻撃	**6**

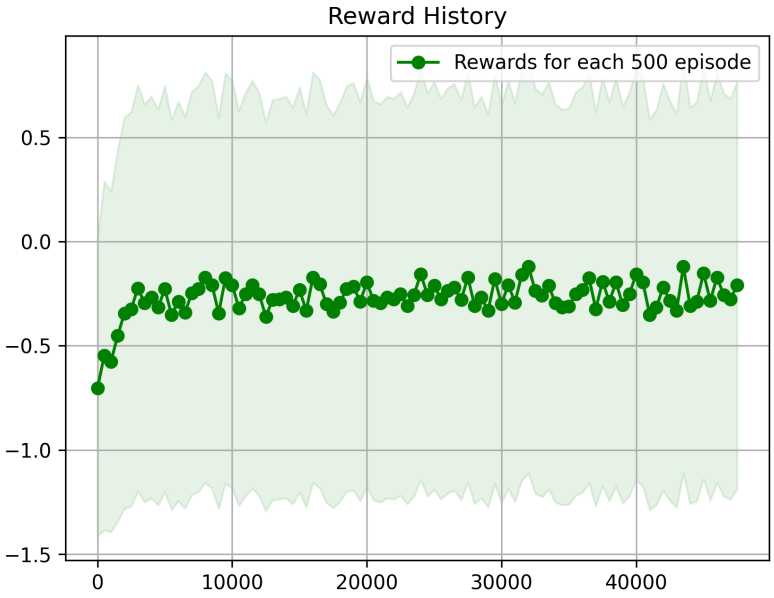

図 2: 実験 2 - 1 における平均獲得報酬の推移

実験 1 に比べ , 学習は進んでいくにつれ平均の報酬が高くなり, 約 6 割の勝率を記録した. しかし, 先攻側にアルゴリズム 1 で示した行動ルーチンを持つプレイヤーを配置した場合の勝敗に比べると小さく, ルールベースな敵 AI よりも強化学習を用いる意味がない.

3.3 実験 2 - 2

手札において「盤面に出さない」という選択肢が学習が進まなくなる要因であると当たりをつけ, プレイヤーの選択肢に「ターンエンド」を追加した. 表 7 に定義した行動空間を示す.

この条件下で先攻側を 3000000 ステップ学習し, 10000 回ゲームを実行し勝率を計算した.

3.4 実験 2 - 2 結果

図 3 に実験 2 - 2 における学習時の 500 エピソードの平均獲得報酬の推移を示す.

また, 表 8 に勝率を示す.

表 6: 実験 2 - 1 結果

手法	勝率
DQN	0.6022
対戦相手と同じ戦略	**0.6425**

表 7: 実験 2 - 2 で定義した行動空間 (太字は変更した箇所)

行動説明	次元数
手札 1 ～ 9 を自盤面に出す	9
自盤面 1 が敵盤面 1 ～ 5 に攻撃 or 敵プレイヤーに攻撃	6
自盤面 2 が敵盤面 1 ～ 5 に攻撃 or 敵プレイヤーに攻撃	6
自盤面 3 が敵盤面 1 ～ 5 に攻撃 or 敵プレイヤーに攻撃	6
自盤面 4 が敵盤面 1 ～ 5 に攻撃 or 敵プレイヤーに攻撃	6
自盤面 5 が敵盤面 1 ～ 5 に攻撃 or 敵プレイヤーに攻撃	6
ターンエンド	**1**

学習時の獲得平均報酬が大きく上昇し, ルールベースで作成した敵に対して 9 割 5 分以上の勝率を記録することができた. 学習したエージェントの行動を見てみると先攻プレイヤーらしく, 積極的に相手プレイヤーに攻撃し, 手札からも攻撃の特殊効果を持つカードを優先的にプレイしていた.

4 後攻側の学習

高い勝率を記録した実験 2 - 2 の条件で学習プレイヤーを後攻に配置して, 1000000 ステップ学習後 10000 回ゲームを実行して勝率を計算した. 図 4 に学習時の 500 エピソードの平均獲得報酬の推移を示す.

また, 表 9 に勝率を示す.

後攻側の学習においてもルールベースで作成した敵に対して高い勝率を記録した.

5 今後の課題

- 対戦相手の行動の改善

  今回の実験で, エージェントの行動空間の定義を改善することができ学習によりルールベースで作成した敵よりも高い勝率を残すエージェントを作成することができた. しかし現在は学習の際, 対戦相手は好戦的な行動ルーチンに基づいて行動している. バランス調整のシミュレーション回す際に学習したエージェントを用いる予定なので防戦的な行動ルーチンを作成し, エピソードごとに敵のルーチンを変えるなどしてどちらにも勝てるようなエージェントを作成する必要がある.

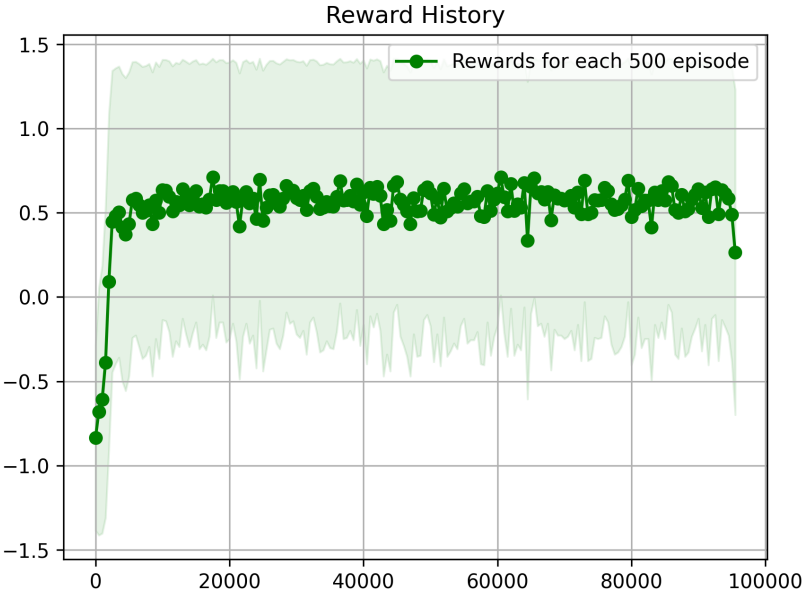

図 3: 実験 2 - 2 における平均獲得報酬の推移

表 8: 実験 2 - 2 結果

手法	勝率
DQN	**0.9708**
対戦相手と同じ戦略	0.6425

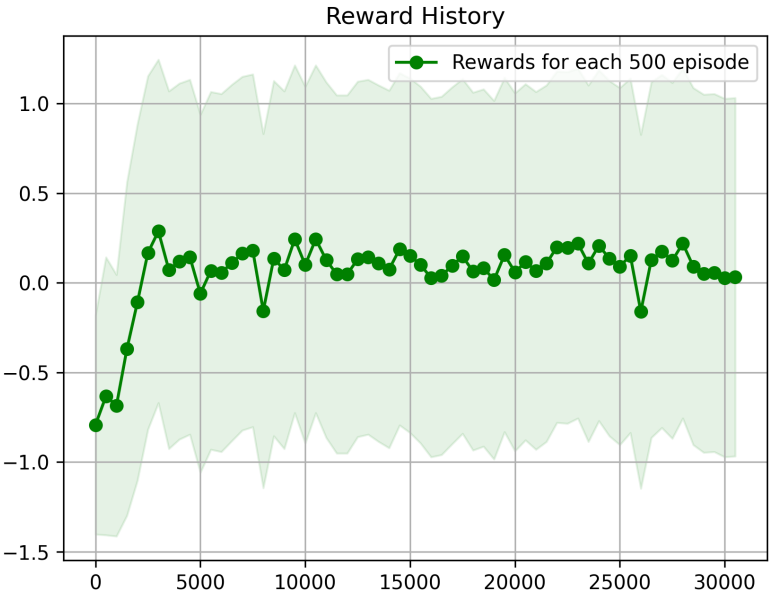

図 4: 学習時における平均獲得報酬の推移

表 9: 実験結果

手法	勝率
DQN	**0.7969**
対戦相手と同じ戦略	0.3575