2022 年 12 月 27 日 B3 西村昭賢

進捗報告

1 今週やったこと

- 研究発表会で頂いた指摘, アドバイスについての考察
- ランダムに行動するプレイヤー同士の勝率計算

2 研究発表会で頂いた質問, アドバイスについての考察

- DQN で action がどのように決まるのか

  DQN では, 図 に示した疑似コード [1] における下から 3 行目で計算する loss を最小化するように学習し Q 値を更新する. DQN の実装で用いた keras-rl では, 方策に基づいて action を決定する際に, それぞれの action (離散値) に対して Q 値の推定値を算出する [2] . 実験で用いた ϵ - greedy では, 確率 ϵ でランダムに行動し, それ以外では算出された Q 値の推定値から最も値が大きい action を選択する.

  このため, 質問にあったような action が 8.5 とかになって 8 と 9 のどちらかを選ぶ状況にはならない. そもそも DQN は離散的な行動空間にしか対応していない.

Algorithm 1 Deep Q-learning with Experience Replay

Initialize replay memory $\mathcal{D}$ to capacity N
Initialize action-value function Q with random weights
for episode $= 1, M$ **do**
  Initialise sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**
    With probability ϵ select a random action a_t
    otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ according to equation 3
  end for
end for

図 1: DQN の疑似コード [1]

- 標準偏差が学習進むにつれて小さくならないのか

  これも, ϵ - greedy によるものと考えられる. 学習が安定してからも確率 ϵ でランダムに行動するため, 平均を取るエピソード数の勝敗にばらつきが発生すると考えられる.

- 考察に学習プレイヤーの学習序盤と学習後期の行動の比較, 人間が考える最善手との比較を含めるべき

  資料や発表でどのように示すかが難しそうと考えた. 現在は学習済みモデルにおいて 5 エピソードくらいのログを見てどのように行動しているか判断している. 例えば先週の資料では, 「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.」といったように記載した.

表 1: ランダムに行動するプレイヤーの行動空間

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

表 2: 10000 回対戦し計算した先攻の勝率 (戦略はミラー)

戦略	勝率
ランダム	0.5011
アグロ	0.6661

これだけだと主観的すぎると考えた. そのため勝利時に盤面にプレイした平均枚数をカードごとに算出できるようにした. これで学習プレイヤーがどのカードを重視して盤面にプレイしているかを示すことができる. しかし, これだけでは盤面での行動の様子が数字として示せない. 各カードについてどのカードを攻撃したか, 相手プレイヤーに直接攻撃したか回数を測定すれば盤面のカードについての傾向が示せるのではないかと現時点では考えている.

- どこに新規性があるのかはっきり言えるようにしておくべき

  森先生と相談した.

  バランス調整という抽象的な概念について, 定量的に評価する. それを構築環境で実験することに新規性がある.

3 ランダムに行動するプレイヤー同士の勝率計算

バランス調整に取り組むにあたり, まずはランダムに行動するプレイヤー同士の勝利を計算できるようにした. ランダムに行動するプレイヤーは, 表 1 に示す行動空間に沿って, 取りうる行動の中からランダムに行動を選択して行動する. なお, 記載ミスで前回の資料で示した実験は全てゲーム開始時に初期手札としてカードを 5 枚ドローしていた. 手札のマリガンといった操作がないため意図的に変更していたが変更を記録していなかった. 申し訳ありません.

今回ではゲーム開始時の初期手札枚数を 3 枚に戻し, ランダム同士の対戦, 先週の資料で示したアグロ的な戦略同士の対戦を 10000 回実行し先攻の勝率を計算した. 表 2 に結果を示す. 先週, アグロ同士の先攻の勝率は 0.6425 であったが , 初期手札の枚数が減り運要素が高まったためさらに先攻有利に傾いた.

4 今後やること (優先度高い順)

- 「何もしない」を行動空間に含めた際に,「何もしない」が損であると学習しているかどうか確認

  盤面, 手札のカードに対して「何もしない」という行動を加えた行動空間で 100000 ステップほど学習させた結果, 改良した ϵ - greedy が指数関数的に減少するため学習のかなり序盤で ϵ が最小値を取り局所解に入ってしまい上手く学習が進まなかった. 先週, 手札に対して「何もしない」行動を加えた行動空間では, 改良した ϵ - greedy で学習できたためその行動空間に変更し確認してみる.

- バランス調整

  ランダム同士でほぼ五分五分の勝率であったため, 実験として後攻の勝率が 55 % 以上になるまでパラメータを調整してみる. 調整するパラメータは, 現段階では 「後攻の初期手札の枚数」, 「後攻の HP 最大値」 , 「後攻のコスト初期値」の 3 つを考えている.

- 構築環境の改良

  アグロが最適解の環境になっているため, ブロッキングと全破壊の特殊効果を追加して DQN を適用して学習プレイヤーの行動を観測する.

- 各カードについてどのカードを攻撃したか, 相手プレイヤーに直接攻撃したか回数を測定

  DQN の適用実験の結果として必要.

参考文献

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning. *arXiv e-prints*, p. arXiv:1312.5602, December 2013.

[2] `https://github.com/keras-rl/keras-rl/blob/216c3145f3dc4d17877be26ca2185ce7db462bad/rl/agents/dqn.py`.