強化学習を用いたゲームエージェントの評価

リスク工学グループ演習 1 班
小清水亮太　宮澤一矢　原和希　HUANG YUMENG
アドバイザー教員　高安亮紀　遠藤靖典

1. 研究背景

1.1. 第 3 次 AI ブームの到来

近年，AI の実用化が急速に進み，第 3 次 AI ブームが到来している[1].「ビックデータ」と呼ばれる大量のデータを用いることで人工知能（AI）自身が自ら知識を獲得する「機械学習」の実用化や，知識を定義する要素（特徴量）を人工知能（AI）が自ら習得するディープラーニングの登場が，ブームの背景にある. 第 3 次 AI ブームの到来により，機械学習を用いた研究が多くなされている.

1.2. 強化学習

機械学習を用いた手法の一つとして強化学習がある．強化学習とは，エージェントがある環境において経験から累積される報酬を最大にする行動を自主的に学習する機械学習の一種である．強化学習はシステム制御の分野で多く利用され，自動運転技術や自律ロボットの行動選択などに応用されている[2][3]．強化学習はゲーム開発の分野でも，ゲームのリリース前に動作に不具合がないか確認するテスト作業や，適切な難易度になっているかゲームバランスの確認を行う作業に利用され，ゲームの品質の向上につながっている[4].

近年では，強化学習を用いることで強いゲーム AI が提案されている．2015 年には Google DeepMind 社によってコンピュータ囲碁プログラム AlphaGo が開発され，プロ囲碁棋士に対して勝利を収めた[5]．また，2017 年には人工知能 Ponanza がプロの将棋棋士に勝利を収めている[6].

2. 研究目的

近年では強化学習を用いた強いゲーム AI が作成されるようになっているが，その強化学習手法間の結果や特徴の違いを把握し，評価することは現実問題において適切な手法を選択できるようにするために重要である.

そこで本研究では，Ms.Pacman というゲームを題材とし，そのエージェントを作成する．エージェントの作成に使う手法として，各種強化学習手法を用いる．そして各種手法によるエージェントの獲得する得点や特徴を評価する.

3. 手法

3.1. 強化学習モデル

強化学習では，対象のタスクに対して行動や状態を以下のようにモデル化する.

- 状態 S：Environment(環境)から取得される現在の状態を示す（操作するキャラクターや敵の位置など）.
- 行動 A：行動を示す（左右に移動，ジャンプするなどの行動）.
- 報酬 R：状態 S の時にある行動をした際に，得られた利益.

状態遷移は，マルコフ性に基づいて遷移していく，つまり t+1 ステップ目における状態は，t ステップ目での状態と行動のみに依存して，$p(S_{t+1} \mid S_t, A_t)$で決定する．以下の手順で学習が進行する.

1. Environment から時刻 t における状態S_tを観測する
2. Agent が方策 $\pi(a_t \mid s_t)$に従って行動a_tを決定する
3. 行動a_tを行うことによって，状態遷移確率 $p(s_{t+1} \mid s_t, a_t)$にしたがって未来の時刻 t+1 における状態であるs_{t+1}が決定する
4. 行動a_tによって得られた報酬を返す報酬関数 $R(s_t, a_t, s_{t+1})$が決定する
5. 1 に戻り，同様に繰り返す

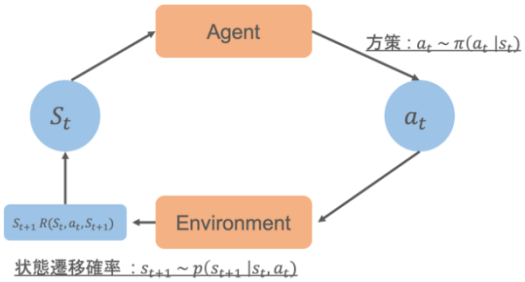


図 1 強化学習のフロー図

3.2. 方策

強化学習の目的は，可能な限り多くの報酬得られるように方策を最適化していくことである．つまり，Environment から取得される「報酬 R」と「状態 S」を元に方策を改善していく．方策の代表例として，greedy 法や局所解に陥るのを防ぐために確率 ε でランダムに行動し，確率 1-ε で greedy 法をとる ε -greedy 法などが存在する.

3.3. 行動価値関数

行動価値関数とは，状態 s での行動 a の価値を推定する関数である．この関数によって，状態 s を入力として，次に取るべき行動として何が最適な行動なのかを推定・評価することが出来る．この行動価値を得るための代表的な手法の 1 つが Q-Learning である．Q-Learning では Q 値と呼ばれる，ある状態 s において，ある行動 a を行ったときの価値を以下の式（1）に基づいて更新していく．ここで状態は s，行動は a,報酬を R，時間割引率を γ，学習率を α とする.

$$\mathbf{Q(s,a)} \leftarrow \mathbf{Q(s,a)} + \boldsymbol{\alpha}\left(\mathbf{r}_{t+1} + \boldsymbol{\gamma}\mathbf{max}\,\mathbf{Q}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) - \mathbf{Q(s,a)}\right) \quad (1)$$

$r_{t+1} + \gamma \max Q(s_{t+1}, a_{t+1}) - Q(s,a)$が報酬の期待値と現在の行動価値の誤差となっており，この値を 0 に近づけることを目的として式（1）を更新していく．この学習方法を TD 学習と呼ぶ．そして，Q 値を求めていき，状態と行動による Q 値を示された Q-Table と呼ばれる表を埋めていく．この Q-Table に基づいて次に取るべき行動として最適な行動を決定することが出来るようになる．図 2 の左側が Q-Table の例となる.

3.4. Deep Q Network (DQN)

Q-Table は取りうる状態空間を全て羅列する必要がある．状態数が少なければ，Q-Table $Q(s,a)$を構成することが可能であるが，複雑な実世界においては状態数が莫大になり，テーブル関数では表現することは

困難となる．そこで Q-Learning における行動価値関数$Q(s,a)$をディープニューラルネットワークにより近似する手法である Deep Q Network(DQN)[7] が使用される．DQN では Q 値そのものを推定するのにニューラルネットワークを使う．ニューラルネットワークの重みパラメータである θ を用いて，行動価値関数を近似する．実際には，$Q_\theta(s,a)$と$Q_\pi(s',a')$の 2 つのニューラルネットワークを用いている．$Q_\theta(s,a)$は最適な行動を選択して，Q 関数を更新する役割を担っており，一方で，$Q_\pi(s',a')$は行動の結果の s′ で取るべき行動 a′ の価値を推定，評価する役割を担っている．式 (2) が DQN の更新式となる．

$$\mathbf{Q_\theta} \leftarrow \mathbf{Q_\theta} + \boldsymbol{\alpha}\left(\mathbf{R(s,a)} + \boldsymbol{\gamma}\max_{\mathbf{a'}}\mathbf{Q_\pi(s',a')} - \mathbf{Q_\theta(s,a)}\right) \quad (2)$$

図 3 のように，状態を入力データとして，出力層の各ノードが各行動の行動価値つまり Q 値を推定するようになっている．また，通常のニューラルネットワークでは，Target となる正解データに基づいて θ の最適化を行うが，DQN では明確な正解データが用意されていないため，Target データ，つまり教師あり学習における正解データの代わりとして式 (3) を使用する．

$$\mathbf{target}_{DQN} = \mathbf{R(s,a)} + \boldsymbol{\gamma}\max_{\mathbf{a'}}\mathbf{Q_\pi(s',a')} \quad (3)$$

したがって，誤差関数は式 (4) のように定義され，これを TD 誤差と呼ぶ．この誤差を元に誤差逆伝播法を用いてニューラルネットワークのパラメータである θ を最適化することによって，最適な Q 値を推定出来るようになる．

$$\mathbf{L} = \mathbf{E}\left[\frac{\mathbf{1}}{\mathbf{2}}\left(\mathbf{R(s,a)} + \boldsymbol{\gamma}\max_{\mathbf{a'}}\mathbf{Q_{\theta-1}(s',a')} - \mathbf{Q_\theta(s,a)}\right)^{\mathbf{2}}\right] \quad (4)$$

3.5 Double DQN

Double DQN(DDQN)[8]とは，DQN における Target データ (式(3)) を改良した手法である．DDQN では，行動価値関数 Q に対して，価値と行動を選択するニューラルネットワークと，その行動を評価するニューラルネットワークの 2 つに役割を分ける．これによって，通常の DQN の計算では行動価値の推定が過大評価されてしまうという問題に対して対応することが出来る．Double DQN の計算では過大評価な推定を抑えて，より正確に推定することが可能になる．式 (5) が DDQN における Target となる式である．

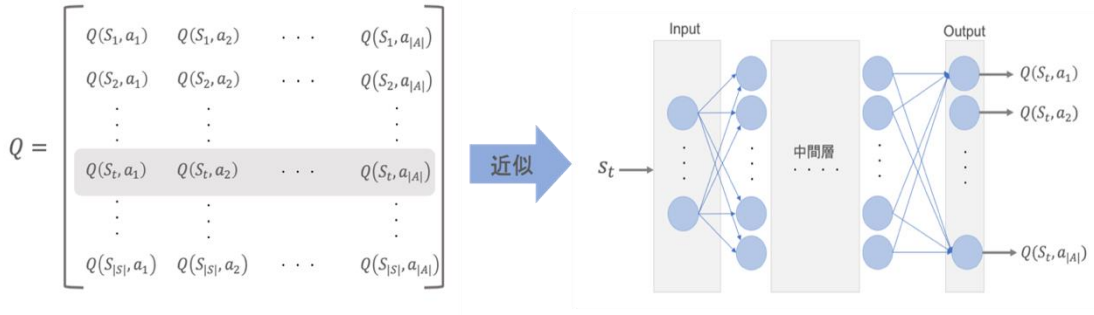


図 2 Deep Q-Network の例

$$target_{DDQN} = R(s,a) + \gamma Q_{\pi}(s', argmax_a Q_{\theta}(s',a)) \quad (5)$$

まずDQNでは，次状態s′のもとで取るべき最善の行動の評価値$\max_{a'} Q_{\pi}(s',a')$用いてQ_{θ}を更新する．つまり，次に取るべき行動の選択とその評価を1つのニューラルネットワークQ_{π}を用いて行っていることになる．一方でDouble DQNは，次の状態s′で取るべき行動aをニューラルネットワークQ_{θ}を用いて，$\mathrm{argmax}_a Q_{\theta}(s',a)$により決定する．そして，その行動aの評価にはDQNと同様にQ_{π}を用いて$Q_{\pi}(s',a)$を求め，Q_{θ}を更新する．これにより，次の行動選択をQ_{θ}で行い，その選択の評価をQ_{π}で行うことが可能になるため，行動価値の推定が過大評価されるのを防ぐことが可能になる．

3.6 Dueling Network

Dueling NetworkとはDQNのアーキテクチャの1つである．Q関数は，状態sのみで決定することが可能な情報と行動aによって決定する情報に分離することが可能であることに着目し，これら2つの情報を分けて学習する手法である．これによって，ある状態s自体の価値を行動aに対する評価を介することなく直接的に表現することが可能になる．Dueling Networkは，DQN, DDQNなどの手法と組み合わせて，収束を早める効果やパフォーマンスを向上する効果が期待出来る．

4. 実験

4.1. 実験環境

本研究では，OpenAI[10]が開発した強化学習のシミュレーション用環境であるOpenAI Gym[11]を使用した．OpenAI Gymにはアメリカのアタリ社が開発した家庭用ゲームであるAtari2600に対応したゲームが数多く含まれている．今回はその中でもMs.PacManというゲームを対象に実験を行う．図3がゲームの操作画面となる．ゲームルールは，PacManを操作してゴーストから逃げつつ，画面上にあるpillを食べ，クリア及び高得点を目指すゲームとなっている．通常のPacManとの大きな違いはゴーストの動きにランダム性が導入されており，絶対に安全であるルートが存在しないことである．

図 3 Ms. PacManの画面

今回は強化学習の手法として，DQN[7]，DDQN[8]，Duel. DQN[9]，Duel. DDQN[9]の4つの手法をディープラーニング実装ライブラリの1つであるKerasおよびKeras-RL[12]を用いて実装し学習，評価を行った．入力データとして図3のようなゲーム画面を畳み込みニューラルネットワーク（CNN）を使用して読み込むことにより，Q値を求め，学習を行った．

4.2. 実験結果

表1が各手法の実験結果となる．横軸が学習を行ったステップ数となっており，縦

軸が学習後のモデルによって 10 回テストを行った平均獲得報酬となっている．

4.3. DQN vs DDQN

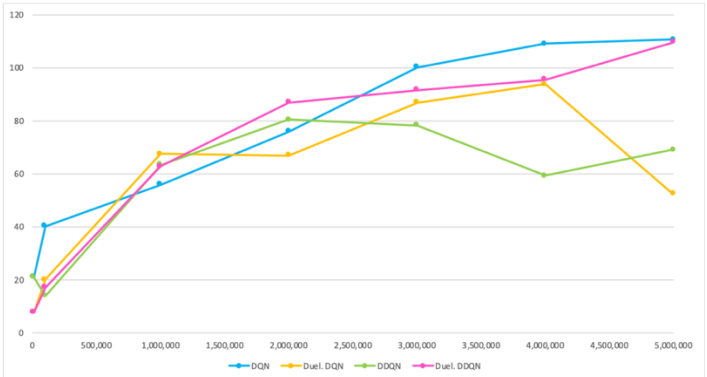
DQN と DDQN を比較するとほとんどのステップ数にて, DQN の方が報酬を多く獲得することが出来ていることが見て取れる．DDQN の提案されている論文[8]でも DQN と DDQN の性能は全体では DDQN の方が優れている傾向にあるが，一部のゲームにおいては性能が悪化すると報告されており，今回対象にした Ms. Pacman においても悪化するという傾向を確認することが出来たと言えるだろう．しかしながら，今回は比較のためにハイパーパラメータやニューラルネットワークのアーキテクチャは同様のものを使用したが, DDQN の提案されている論文では各々最適なパラメータの探索も行っており，DDQN において Target network の更新間隔を調整することによってパフォーマンスが向上するということも報告されている．そのため，DQN の方が優れていると断言は出来ないと考える．今後 DDQN のパラメータ調整を行っていく必要があると考える．

4.4. DQN vs Duel. DQN

次に DQN と Duel. DQN を比較する．DQN は報酬の伸びがなだらかであり．安定していることが見て取れる．一方で Duel. DQN は各ステップにおける報酬のばらつきが大きい．このことから，学習が不安定になっていると考えられる．これは，Duel. DQN は DQN と比較して学習において調整すべきハイパーパラメータの数が多く，今回の検討においては，ハイパーパラメータの調整が不足しているためであると考えられる．

また，Duel. DQN では，4,000,000 ステップ程度で最高獲得報酬を記録し，それ以降は過学習によって，報酬が低下している特徴もみられる．実際に提案論文[9]において，Duel. DQN は収束を早める効果があると報告されている．本研究でも同様に DQN と比べて収束を早めることが出来ていることを確認することが出来た結果であると言えるだろう．以上のことから，Dueling Network を使用することによって収束を早めることが出来るが，学習をより安定させるためには，ハイパーパラメータの探索を

表 1 比較結果(DQN, Duel.DQN, DDQN, Duel.DDQN)



Step数	10,000	100,000	1,000,000	2,000,000	3,000,000	4,000,000	5,000,000
DQN	21	40.3	56	75.9	100.1	109	110.6
Duel. DQN	7.8	20.1	67.5	66.8	86.9	93.8	52.4
DDQN	21	14.1	63.3	80.4	78.2	59.2	69.1
Duel. DDQN	7.8	17.2	62.8	86.9	91.6	95.5	109.5

行うことが重要であるという知見が今回の検討によって得られた．

5. まとめ

本研究では Ms.PacMan というゲームを対象に各種強化学習手法を用いて実験を行った．また，各種手法の実験結果の比較，考察を行った．結果として，今回の実験では手法として DQN を用いて 5,000,000 ステップ学習させたものが平均獲得報酬が最も大きいという結果となった．また，今回の問題における各手法の結果や特徴について評価することができた．

今後の課題としては，まず今回実験していない手法についての実験，評価を行うことがあげられる．例えば高精度を出す手法として Rainbow[13]という複数の手法を統合した手法が提案されている．このような今回実験できなかった手法についても評価を行う必要がある．

また，ハイパーパラメータや層の構成により精度が変わることが考えられるため，それらの適切なチューニングを行うことも検討する必要がある．

参考文献

[1]総務省．平成 28 年版 情報通信白書｜人工知能（AI）の研究の歴史, http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h28/html/nc142120.html .

[2]WAYVE, leaning to drive in a day. https://wayve.ai/blog/learning-to-drive-in-a-day-with-reinforcement-learning

[3] Andres El-Fakdi, Marc Carreras “Policy gradient based Reinforcement Learning for real autonomous underwater cable tracking”, 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems(2008)

[4] 株式会社 BrainPad，深層強化学習によるゲーム AI の開発支援, https://ai.brainpad.co.jp/case-study/2631/

[5] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”, Nature529, 484-489(2016).

[6]HEROZ 株式会社, Ponanza における強化学習とディープラーニングの応用, https://www.slideshare.net/HEROZ-JAPAN/ponanza-83900718

[7] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015)

[8] Van Hasselt, Hado, Arthur Guez, and David Silver. "Deep reinforcement learning with double q-learning." Thirtieth AAAI conference on artificial intelligence. 2016.

[9]Wang, Ziyu, et al. "Dueling network architectures for deep reinforcement learning." arXiv preprint arXiv:1511.06581 (2015).

[10]OpenAI，https://openai.com

[11]OpenAI Gym, https://gym.openai.com/

[12] Matthias Plappert, keras-rl, GitHub repository, ,https://github.com/keras-rl/keras-rl,

[13] Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, David Silver. “Rainbow: Combining Improvements in Deep Reinforcement Learning ” arXiv:1710.02298v1 [cs.AI] 6 Oct 2017