3K1-OS-06a-4

深層学習と強化学習を用いたゲームAIへの時系列予測の導入

Introduction of Time Series Prediction to Game AI with Deep Learning and Reinforcement Learning

松尾　星吾　　岡　夏樹
Seigo MATSUO　　Natsuoki OKA

京都工芸繊維大学　大学院工芸科学研究科
Graduate School of Science and Technology,　Kyoto Institute of Technology

In this research, we attempt to develop an agent that plays Atari 2600 games using deep learning, reinforcement learning, and time series prediction. The agent predicts future states to be branched from the current state according to the operations of the player in a game. It selects the next action which will maximize the expected future rewards from the predicted states. We experimentally compared the performance of four types of agents: a Deep Q-Network that combines deep learning and reinforcement learning, the first proposed agent that is additionally integrated with time series prediction and maximizes the sum of the future rewards of the branched states, the second proposed agent that is similar to the first proposed agent but maximizes the maximum of the future rewards of the branched states, and an agent that randomly takes actions. We tested the agents by playing the Atari 2600's MsPacman and Breakout. The average score of the second proposed agent was higher than that of the Deep Q-Network. Analysis of the results is ongoing, and we plan to improve the reward evaluation by the Deep Q-Network, increase the accuracy of state prediction, and reexamine the algorithm of action selection.

1. はじめに

強化学習とは，エージェントが試行錯誤することによって学習を行う機械学習手法である．エージェントは観測した状態に基づき行動を選択する．環境の変化に伴って何らかの報酬が与えられ，徐々に環境に適した行動を学習していく．強化学習は様々な分野で成功を収めているが，状態を完全に表現できる低次元の問題にしか適応できず，より実世界に近い問題で学習を成功させるには複雑な高次元の入力を効率的に表現する必要があった [Tesauro 95].

V. Mnih, K. Kavukcuoglu, D. Silver ら [Volodymyr Mnih 15] は Deep Q-network と呼ばれる新たなエージェントを開発し，高次元の入力での学習を成功させた．これは，強化学習の一種である Q 学習の最適行動価値関数をディープニューラルネットワーク [Bengio 12] で近似した手法である．畳み込みニューラルネットワークを用いることで，より複雑な状態を観測することを可能にした．往年の家庭用ゲーム機，Atari 2600 [Bellemare 13] の 49 種類のゲームでテストを行い，そのうち 29 種類のゲームでプロゲーマーと同等，またはそれ以上のパフォーマンスを記録した．しかし，極端に学習が進まないゲームもあり，Ms. Pac-Man や Montezuma's Revenge などのゲームは，完全にランダムで操作した場合と記録したスコアとほとんど差がなかった．このエージェントは 4 フレームの画像のみを入力とするため，即時に成功するような単純な戦術しか学習できないという弱点が指摘されている．

本研究ではこの問題に対し時系列予測を用いることで，エージェントが観測する状態を網羅的に予測し，より複雑な戦術を実行できるエージェントの作成を試みる．一般的な家庭用ゲーム機では，方向を入力するキーと複数のボタンによって操作が行われる．プレイヤーの入力によってゲーム内の状態が分岐していくため，次の状態を予測するには，操作ごとに別の予測をする必要がある．本研究では，時系列予測を行うニューラルネットワークモデルである Prednet [Lotter 16] を用いて，エージェントが観測する次の状態を予測する．

本研究では深層学習と強化学習を用いたエージェントに時系列予測を導入した手法を提案する．古典的なビデオゲームを実行するというタスクおいて，時系列予測を用いない場合と比較し，エピソード終了時のスコアが向上するのかを検証する．

連絡先: 606-8585 京都市左京区松ヶ崎橋上町 1
京都工芸繊維大学 大学院工芸科学研究科 情報工学専攻
インタラクティブ知能研究室，matsuo@ii.is.kit.ac.jp

2. 関連研究

本研究では，深層学習と強化学習を用いたエージェントに Deep Q-network を用いた．Q 学習は強化学習手法の一つである．Q 学習では，エージェントの目標は将来の累積報酬を最大化するような行動を選択することである．エージェントは，最適行動価値関数を用いて Q 値を推定することで強化学習を行う．最適行動価値関数は，各ステップ t において γ で割り引かれた報酬 r_t の合計が最大となる値を求める．Deep Q-network ではこの関数をディープニューラルネットワークを用いて近似している．

また，時系列予測には PredNet と呼ばれるモデルを用いた．Deep Predictive Coding Networks(PredNet) は，神経科学における Predictive Coding [Bastos 12] と呼ばれる仮説に基づいて考えられたニューラルネットワークモデルで，動画の時系列予測を行うことが可能である．仮説によれば，脳は外部からの感覚入力をうまく取り扱うために内部予測モデルを構築し，将来起こりうる現象に対し適切な対応ができるようにすると考えられている．脳は階層構造をとっており，学習により得られた内部予測モデルに従い上の層から下の層へとトップダウンに予測信号を伝達する．この予測信号が実際の感覚入力と違う場合，下の層から上の層へと予測エラー信号がボトムアップに伝達され，内部予測モデルを更新する．トップダウンな伝達とボトムアップな伝達の相互作用で予測モデルを最適化し，予測エラー信号を最小化する働きをすると考えられている．

3. 手法

本研究で用いる時系列予測を導入した 2 種類のエージェントについて説明する．

囲碁や将棋など，自分と相手の取り得る行動が完全に決まっているゲームでは，互いが選択する行動によってゲームの状態が木構造のように分岐していく．プレイヤーは自分の中に相手のモデルを持っており，相手が取る行動を予測して将来の状態を予測する．予測した将来の状態を自身の経験で評価し，有利であるような選択，自分が勝てるような選択をする．

一般的なビデオゲームでは，方向を入力するキーと複数のボタンを組み合わせたインターフェースで操作を行う．人間はゲームをプレイするとき，右キーを入力したらキャラクターが右側に移動する，×ボタンを押したらジャンプする，といった知識を持っており，それらの経験を元に次の行動を決定する．ビデオゲームでは，環境が確率的で次の状態がランダムに変化することがあるが，基本的にゲーム内の状態はプレイヤーの行動で分岐していく．このようなゲームであっても，プレイヤーは操作を行うとどのような状態になるのか，次の状態がよりよい状態になるにはどの操作を選ぶのが適切か，といった判断を過去の経験を元に下す．

Deep Q-network では，現在の状態を観測し，その状態に対して最も報酬の期待値の高い選択肢を取る．本研究では，未来の状態を予測し，その状態を利用して次の行動を決定することで，エージェントがより良い行動を選択できるのではないかというアイデアから，時系列予測を導入した．時系列予測には PredNet を用い，予測した状態を Deep Q-network で評価し行動の選択を行う．

本研究では，予測した状態の Q 値の合計を考慮する手法と Q 値の最大値を考慮する手法を提案する．

3.1 手法 1

エージェントの行動選択は次の式 1 に従う．エージェントが状態 s_t で行動 a を行ったときに予測される将来の状態の集合を S^a_{t+n} で表す．n は何段階先の将来の状態かを表す．a は状態 s_t で選択した行動を表し，S^a_{t+1} より後の状態で選択した行動ではない．エージェントは将来の報酬の期待値の合計が最も高くなる行動を選択する．

$$a^* = \arg\max_a \sum_{s \in S^a_{t+n}} \sum_{a' \in A} Q(s, a') \tag{1}$$

$\sum_{a' \in A} Q(s, a')$ で状態 s での各行動の Q 値の合計を求める．この計算を，$\sum_{s \in S^a_{t+n}} \sum_{a' \in A} Q(s, a')$ で各状態に対して行う．2 段階先までの予測を行う場合，式 2 は図 1 の部分の Q の総和を取る．

$$\sum_{s \in S^a_{t+n}} \sum_{a' \in A} Q(s, a') \tag{2}$$

エージェントの行動選択の手順を以下に示す．

1. 状態 s_t を観測
2. s_t から n 段階先の将来の状態を予測 $\{S^a_{t+n} | a \in A\}$
3. 式 1 に従い行動を選択

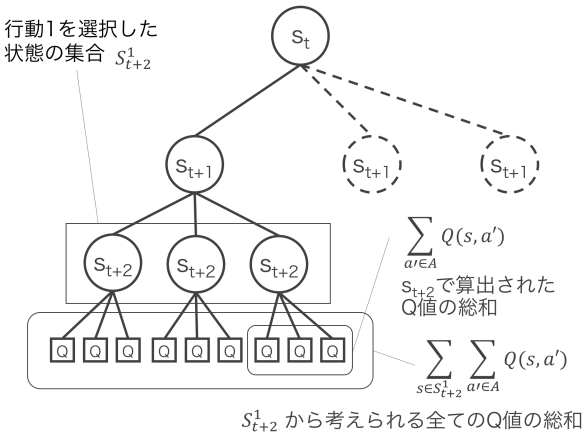

図 1: 式 2 で計算される Q 値の総和

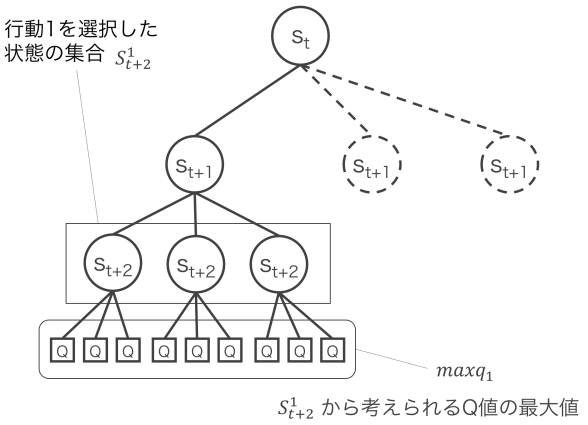

図 2: 式 3 で計算される Q 値の最大値

3.2 手法 2

エージェントは，予測した将来の状態を用いて Q 値を計算し，その Q 値が最も高い行動を選択する．S^a_{t+n} に対し Q 値を計算し，その最大値 $maxq_a$ を比較する．$maxq_a$ が最も高くなる行動 a が次の行動となる．

$$maxq_a = \max\{Q(s, a) | s \in S^a_{t+n}\} \tag{3}$$

$$a^* = \arg\max_a \{maxq_a | a \in A\} \tag{4}$$

式 3 は図 2 の部分の Q 値の最大値である．

4. 実験

4.1 実験方法

本研究では，3 つのエージェントを Atari 2600 エミュレータ上で実行し比較することで，深層学習と強化学習を用いたエージェントへの時系列予測の導入がどの程度ゲームのスコアに影響を与えるのかを評価する．比較するエージェントは以下の 4 種類である．

エージェント **1** 手法 1 を用いたエージェント

エージェント **2** 手法 2 を用いたエージェント

エージェント **3** 深層学習と強化学習を用いたエージェント

エージェント **4** ランダムで行動を選択するエージェント

深層学習と強化学習を用いたエージェントは Deep Q-network を用いて実装されたエージェントを用いる．時系列予測を導入した，深層学習と強化学習を用いたエージェントには，Deep Q-network と PredNet を実装したエージェントを用いて第 3 章で説明した手法により行動を決定する．

以下の 2 つのゲームで実験を行う．

1. Ms Pacman
2. Breakout

V. Mnih らが行った実験では，Deep Q-network は Breakout では非常に高い性能を示したが，Ms Pacman においてはランダムに実行した際とそれほど変わらないスコアだった．本実験では，Deep Q-network が得意とするタスク，苦手とするタスクを選択した．エピソード終了時のゲーム内のスコアを記録し，それぞれのエージェントで 30 回ゲームを行い，スコアの平均値を比較する．また，Breakout ではエージェント 1，3，4 のみで実験を行った．

エージェント 1，2，3 は実装に Deep Q-network を用いており，比較のため同様のパラメータ，学習回数を用いる．

4.2 ネットワーク構造

本実験で用いたモデルのネットワーク構造について説明する．

Deep Q-network の構造は以下の通りである．ネットワークへの入力は $84 \times 84 \times 4$ で，これはゲーム画面に対しリサイズ，グレースケール化などの前処理を施した直近の 4 画面である．第 1 層はフィルタサイズ 8×8，フィルタ数 32，ストライド 4 の畳み込み層．第 2 層はフィルタサイズ 4×4，フィルタ数 64，ストライド 2 の畳み込み層．第 3 層はフィルタサイズ 3×3，フィルタ数 64，ストライド 1 の畳み込み層．第 4 層はユニット数 512 の全結合層．第 5 層は，選択可能な行動と同数の出力を持つ．第 1 層から第 5 層までの活性化関数は全て ReLU を用いている．また，ネットワークの最適化アルゴリズムには RMSprop を用いた．

PredNet は Deep Q-network への入力に合わせて，84×84 を入力とした．Deep Q-network はグレースケール画像を入力とするが，Atari 2600 のゲームでは MsPacman など同じ形のオブジェクトでも行動パターンに違いがある場合を考慮しカラー画像を学習させた．PredNet の階層数は 4 とした．

4.3 学習データ

PredNet は各ゲームで選択可能な行動と同数のネットワークを学習させる必要があり，操作ごとにそれぞれ別のデータセットを用意した．Atari エミュレータ上で学習済みの Deep Q-network を実行し，選択した操作ごとにゲーム画面を保存する．MsPacman ではそれぞれ 6000 枚，Breakout ではそれぞれ 4500 枚の画像を学習に使用した．

5. 結果

まず，学習した PredNet でゲーム画面を予測した結果を示す．下キーの動作を学習させた PredNet に，図 3 を入力し，予測結果として図 4 を得た．

MsPacman の実験結果を表 1 に示す．それぞれのエージェントで 100 エピソード実行し，終了時点でのスコアの平均を比較した．手法 2 を用いたエージェントのスコアが，深層学習と強化学習のみを用いたエージェントを上回ったが，手法 1 はそれを下回る結果となった．

Breakout の実験結果を表 2 に示す．MsPacman と同様に，深層学習と強化学習のみを用いたエージェントが手法 1 を用いた結果を上回った．

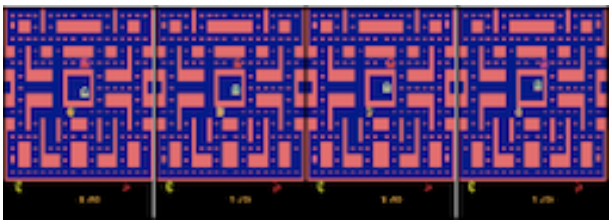

図 3: PredNet への入力

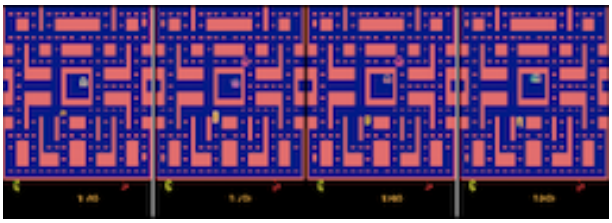

図 4: PredNet の予測結果

表 1: MsPacman を 100 エピソード実行した際の平均スコアの比較

エージェント	スコア
手法 1 を用いたエージェント	370.2
手法 2 を用いたエージェント	605.8
深層学習と強化学習を用いたエージェント	574.4
ランダムで行動するエージェント	270.5

表 2: Breakout を 100 エピソード実行した際の平均スコアの比較

エージェント	スコア
手法 1 を用いたエージェント	4.5
深層学習と強化学習を用いたエージェント	14.4
ランダムで行動するエージェント	1.9

6. 考察

MsPacman と Breakout で実験を行い，手法 1 では時系列予測の導入によるスコアの向上は確認できなかったが，手法 2 では従来の手法を上回った．

PredNet から出力された予測画像を再帰的に入力する処理を繰り返すと，出力される画像は徐々に荒くなる．本実験では，上記の処理により Deep Q-network へ入力する画像がぼやけてしまうのを制限するために，予測する状態は 3 段階先までとしていた．また，PredNet で扱う画像は，Deep Q-network に入力するサイズに合わせて画像を縮小しており，画像があらくなる段階をより早めていたと考えられる．PredNet で予測する画像サイズをゲーム画面と同じサイズで学習させ，予測す

る画像の精度を向上させることで，より先の状態を予測できると考えられる．

本論文で提案した手法 1 は，将来の状態を予測し，その状態での報酬の期待値の平均値が最も高いものを選択するようなアルゴリズムである．上述の通り，本実験では最大で 3 段階先までしか予測していないため，その状態で平均値を計算するとほとんどの場合において非常に近い値となった．

予測を 3 段階先までに抑えており，ゲーム画面が大きく変化することはないため，Deep Q-network での Q 値の予測が大きく変わることは考えにくい．それにも拘らず，手法 1 のスコアが Deep Q-network を大きく下回ったのは，式 1 で示した行動選択のアルゴリズムが不適切であったからだと考えられる．Q 値の平均値ではなく最大値を考慮する手法 2 は，従来の手法を上回った．

また，実装したエージェントの問題点としては，一つの行動を選択するのに多くの時間を必要とする点があげられる．従来の Deep Q-network では，1 つのネットワークに対し順伝播計算を 1 度行い，行動を決定する．しかし，本実験で実装したエージェントは，取り得る行動の数を A のゲームで n 段階先の状態を予測を行う場合，式 5 で表される回数の順伝播計算を行う必要がある．

$$\sum_{i=1}^{n} A^i \tag{5}$$

加えて，予測した全ての状態に対して Q 値を算出する必要がある．演算に必要なコストに見合ったスコアの上昇は確認できておらず，あまりにも非効率であるといえる．

参考文献

[Bastos 12] Bastos, A. M., Usrey, W. M., Adams, R. A., Mangun, G. R., Fries, P., and Friston., K. J.: *Canonical microcircuits for predictive coding.*, Neuron (2012)

[Bellemare 13] Bellemare, , G., M., Naddaf, Y., Y., , Veness, , J., , and Bowling, : The arcade learning environment: An evaluation platform for general agents, *J. Artif.*, pp. 253–279 (2013)

[Bengio 12] Bengio, Y.: Learning deep architectures for AI., *Foundations and Trends in Machine Learning*, Vol. 2, pp. 1–127 (2012)

[Chalasani 13] Chalasani, R. and Principe., J. C.: *Deep predictive coding networks.*, CoRR (2013)

[Lotter 16] Lotter, W., Kreiman, G., and Cox, D.: Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning, *arXiv*, p. 1605.08104 (2016)

[Riedmiller 09] Riedmiller, M., Gabel, T., Hafner, R., and Lange, S.: Reinforcement learning for robot soccer., *Auton. Robots*, Vol. 27, pp. 55–73 (2009)

[Tesauro 95] Tesauro, G.: Temporal difference learning and TD-Gammon., *ACM*, Vol. 38, pp. 58–68 (1995)

[Volodymyr Mnih 15] Volodymyr Mnih, e. a.: Human-level control through deep reinforcement learning, *Nature*, Vol. 14236, pp. 529–533 (2015)