

ニューラルネットワークによる 2048 プレイヤの内部動作解析

松崎 公紀^{1,a)} 寺村 舞童華^{1,b)}

概要：ゲーム 2048 は、スライド&マージ型ゲームに分類される、確率的一人ゲームのひとつである。これまで、N タプルネットワークと強化学習を用いたコンピュータプレイヤが開発され、非常に強いプレイヤが作られている。ニューラルネットワークを用いたコンピュータプレイヤの開発について、近藤と松崎が 2018 年のプログラミング・シンポジウムにて発表し、畳み込み層の層数を 2 から 3 以上に増やすとプレイヤの強さが大きく向上する結果を得ていた。本研究では、畳み込み層が 2 層の場合と 5 層の場合に着目し、そのニューラルネットワークプレイヤの内部動作を解析する。その結果、いずれの場合でも第 1 層では特徴抽出がなされていること、畳み込み層の層数が 2 の場合でもある程度の汎化能力を得ていること、層数が 5 の場合にはより特徴を明確に捉えることができていること、などが確認された。

キーワード：ゲーム 2048, ニューラルネットワーク, 解析

1. はじめに

ゲームのコンピュータプレイヤの開発において、機械学習が広く利用されるようになってきている。その中でも、ニューラルネットワーク、特に畳み込みニューラルネットワークを用いたプレイヤについて活発に研究が進められており、人間のトップレベルを越えるコンピュータプレイヤも作られるようになってきた。そのような例として、囲碁 (AlphaGo [12] や AlphaGo Zero [14]), チェス (Giraffe [6] や DeepChess [2]), 将棋 (AlphaZero [13]), ポーカー (Poker-CNN [17] や DeepStack [10]), ビデオゲーム (Atari games) [9] などがある。

本研究において対象とするゲームは「2048」[1] である。2048 は確率的一人ゲームであり、平行移

動と結合によるゲーム (slide-and-merge games) に分類される。2048 の「ルールを学ぶのは簡単だが極めるのは難しい」という特徴により、非常に多くの人を惹きつけている*1。

これまで、2048 のコンピュータプレイヤが複数開発されてきた。その中で最も成功したアプローチは、N タプルネットワークを評価関数とし、そのパラメータを強化学習により調整するものである。Szubert と Jaśkowski [15] によって 2048 に導入されたこのアプローチに基づき、様々な研究が行われてきた。Jaśkowski による最先端のコンピュータプレイヤ [4] では、N タプルネットワークプレイヤを改良する手法が複数取り入れられており、1 手あたり 1 秒の制限時間のもとで平均得点 609,104 を達成した。

著者らは先行研究 [5] において、畳み込みニューラ

¹ 高知工科大学情報学群

^{a)} matsuzaki.kiminori@kochi-tech.ac.jp

^{b)} 190341z@ugs.kochi-tech.ac.jp

*1 著者によると、ゲームの公開から最初の 3 週間で延べ 3000 年以上もの時間プレイされたとのことである。

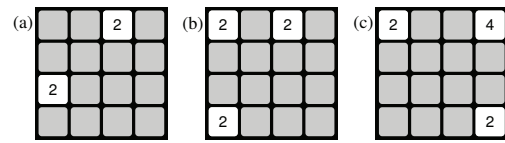
ルネットワークによる 2048 プレイヤを開発した。畳込みニューラルネットワークを 2048 プレイヤに用いる研究には Gueira [3] によるものがあったが、以下の 2 点で異なっている。1 つ目の違いは、畳込み層の数を 2 から最大 9 まで増やして評価を行ったことである。ただし、各層のフィルタ数を変えることで、パラメータ数はほぼ同一となるように調整した。2 つ目の違いは、既存の強いコンピュータプレイヤのプレイログをもとに教師あり学習を行ったことである。6 × 10⁸ 状態分の学習の後の評価から、畳込み層の数が 2 層の場合と 3 層以上の場合で得点に大きな差があることが分かった。また、畳込み層の数が 5 層のとき最も高い平均得点が得られ、探索なし (1-ply) での平均得点は 86,030 であった。

本研究では、畳込みニューラルネットワークの内部動作を解析することにより、畳込み層の数が 2 層の場合と 3 層以上の場合で大きな差が生じた理由を明らかにする。具体的には、まず、第 1 層のフィルタを可視化することで、特徴抽出がどのように行われているかを調べる。次に、特定の盤面についてネットワークを逆向きにたどる逆畳込み (deconvolution) を行い、類似の状態についてどのように手の選択がなされるかを調べる。

本論文における重要な発見を以下にまとめる。

- 畳込みニューラルネットワークの第 1 層は、ゲーム状態の特徴抽出の働きをしている。特に、畳込み層が 2 層のネットワークと畳込み層が 5 層のネットワークとで、対応するフィルタが見られる。
- 類似のゲーム状態について、ネットワークの後段で同様の計算を行っており、それらに共通する知識がエンコードされていると解釈される。畳込み層が 2 層のネットワークもそのような汎化能力をもつが、畳込み層が 5 層のネットワークではより強いことが確認できた。
- 人間のプレイヤであれば容易に良い手を選択できるゲーム状態において、先行研究のニューラルネットワークプレイヤがそれを選択できないことがあることを確認した。

本論文の構成は以下のとおりである。第 2 節で



- (a) 初期状態の例。初期状態では、2 つのタイルがランダムに置かれる。
 (b) 1 手目に「上」を選択した後の状態。左下の角に新しい 2 のタイルが出現した。
 (c) 2 手目に「右」を選択した後の状態。2 つの 2 のタイルが結合して 4 のタイルとなり、得点 4 を得る。左上の角に新しい 2 のタイルが出現した。

図 1 ゲーム 2048 の過程

は、2048 のルールといくつかの重要なヒューリスティックスを説明する。第 3 節では、先行研究 [5] で設計した畳込みニューラルネットワークによる 2048 プレイヤについて説明する。第 4 節では、畳込みニューラルネットワークの第 1 層のフィルタを可視化し、特徴抽出について解析する。第 5 節では、特定のゲーム状態から逆畳込みを行うことで、汎化能力について解析する。第 6 節では、畳込みニューラルネットワークプレイヤの選択する手について、別のゲーム状態を用いて解析する。第 7 節で関連研究を述べ、第 8 節で本論文をまとめる。

2. ゲーム 2048

ゲーム 2048 は、平行移動と結合によるゲーム (slide-and-merge games) に分類される確率的一人ゲームであり、4 × 4 の盤面でプレイされる (図 1)。初期状態では、2 つのタイルがランダムに置かれ、そのタイルの値は 2 (確率 0.9) か 4 (確率 0.1) である。各ターンにおいて、プレイヤは上下左右のいずれかの方向を選択し、全てのタイルを選択した方向に平行移動する。同じ値をもつ 2 つのタイルが移動方向に衝突すると、それらのタイルは結合して和の値を持つタイルとなり、その和の値が得点に加算される。ここで、結合するタイルは奥側が優先され、一度結合したタイルは同じターンではそれ以上結合しない。したがって、222□、□422、2222 から右向きに平行移動した結果は、それぞれ □□24、□□44、□□44 となる。どのタイルも移動・結合しない方向を選ぶことはできない。平行移動と結合の後、新しいタイルが空きマスに出現する。新

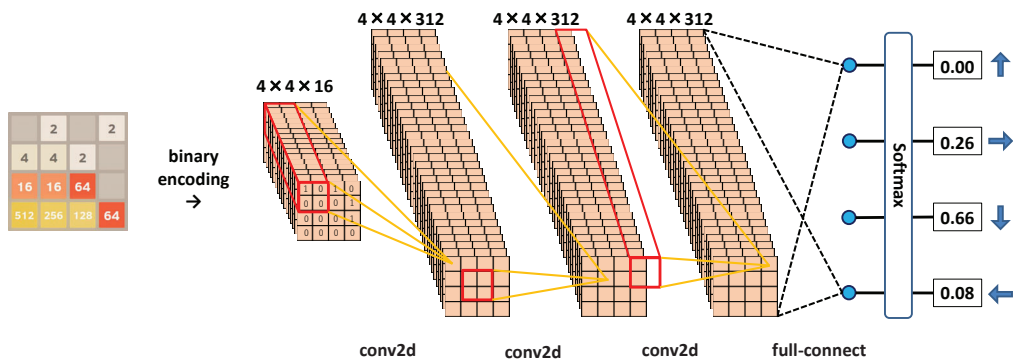


図 2 先行研究 [5] で用いた畳込みニューラルネットワークの全体図 (畳込み層 3 層のネットワークの例)

しく出現するタイルの値は初期状態と同様に 2 (確率 0.9) か 4 (確率 0.1) である。プレイヤーがどの方向にも移動・結合できなくなったら終局である。

もともとのゲーム 2048 における目標は、上記のルールのもとで移動と結合を繰り返して 2048 の値をもつタイルをつくることである。本研究では、2048 のタイルができた後もゲームを続け、できるだけ高得点をとることを目指す。1024 のタイルが最初にできたときの得点はおよそ 10,000 点である。また、2048 のタイルはおよそ 21,000 点、4096 のタイルはおよそ 46,000 点、8192 のタイルはおよそ 100,000 点、16384 のタイルはおよそ 220,000 点、32768 のタイルはおよそ 480,000 点でそれぞれつくることができる。

ゲーム 2048 におけるヒューリスティクスはいくつかあるが、本研究でニューラルネットワークの解析に重要となるヒューリスティクスを 2 つ挙げる。

Heuristic1 最大のタイルを角に置き、それを動かさない。

Heuristic2 値の大きなタイルは、最大のタイルから順に小さくなるよう盤面の辺に沿って並べる。

3. ニューラルネットワークによる 2048 プレイヤ

本節では、著者らによる先行研究 [5] で開発した畳込みニューラルネットワークによる 2048 プレイヤについて説明する。

3.1 ネットワークの構成

著者らが設計した畳込みニューラルネットワークは、畳込み層 k 層、全結合層 1 層、および、ソフトマックスからなる。例として、畳込み層が 3 層の場合のネットワークの構成を図 2 に示す。

入力の盤面は、大きさ 4×4 で 16 チャンネルからなる二値画像にエンコードする^{*2}。各チャンネルは、対応する値をもつタイルの位置を表す。例えば、第 1 チャンネルでは、空白マスが 1 で、それ以外が 0 で表される。

エンコードされた入力に対して、 k 層の畳込み層 (conv2d) を適用する。表 1 に示すとおり、各畳込み層のフィルタ数 $Ch(k)$ (これは中間画像のチャンネル数と等しい) を、全体のパラメータ数がおよそ等しくなるよう調整した。第 1 層は、 $2 \times 2 \times 16$ の大きさのフィルタ $Ch(k)$ 個からなり、続く畳込み層は $2 \times 2 \times Ch(k)$ の大きさのフィルタ $Ch(k)$ 個からなる。いずれの畳込みにおいても、ストライド幅は 1 とし、出力画像の大きさを 4×4 に維持するようパディングを入れている^{*3}。畳込みの後、チャンネルごとのバイアスを足し、活性化関数 ReLU を適用する。

畳込み層の適用後、全ての画素値は全結合層に渡され、全結合層から 4 つの値が出力される。これらの 4 つの値はソフトマックスによって確率に変換される。

^{*2} 入力盤面のエンコードは Guei ら [3] による方法と同じである。

^{*3} このパディングは右端と下端にのみ入るため、畳込みは非対称となる。

表 1 畳込み層の数, チャネル数 (フィルタ数) およびパラメータ数

畳込み層の数 k	チャネル数 $Ch(k)$	パラメータ数
2	436	817,068
3	312	819,628
4	256	820,228
5	224	832,612

表 2 ニューラルネットワークプレイヤーの平均得点と最大得点

畳込み層の数 k	平均得点	最大得点
2	25,669	175,628
3	69,840	332,868
4	80,284	343,496
5	86,030	385,560

3.2 教師あり学習

このように設計したネットワークに対して, 教師あり学習によってパラメータを調整した. 著者らは優れたコンピュータ 2048 プレイヤを開発済みであったため [8], それらのプレイヤーのプレイログを訓練データとして用いた.

著者らの先行研究 [8] の成果物より 3 つのプレイヤーを選び, 訓練データを生成した. これらのプレイヤーは, マルチステージ拡張を行った N タプルネットワークを評価関数とした. その N タプルネットワークは 4 つの 6 タプルから構成され, ゲームは最大タイルの数に応じて 8 ステージへと分けた (ネットワークは 536,870,912 タプルからなる.). N タプルネットワークのパラメータは, ある強化学習手法 (リスタート方策ありの後退 Temporal Coherent 学習) によって 3 通り調整した. このようにして作成して評価関数を用いて, 深さ 3 の Expectimax 探索を行い手を決定する. 実験に用いた N タプルネットワークの平均得点はそれぞれ, 459,455, 463,660, 460,069 であった.

訓練データとして, 45,000 ゲームから 6×10^8 手分のプレイログを抽出した. 1 手分のプレイログには, ゲーム状態とプレイヤーが選択した手が含まれており, そのプレイヤーが選択した手を教師データとした. 訓練データは教師あり学習の前にシャッフルしてある. N タプルネットワークが対称であり, かつ, そのパラメータ調整も完全に対称となるよう行っているため, プレイログも対称性を有

することに注意せよ.

6×10^8 手分の学習を終えた後での畳込みニューラルネットワークプレイヤーの平均得点と最大得点 (10,000 ゲームのテストプレイの結果) を表 2 に示す. この表より, 畳込み層を 3 層以上もつプレイヤーは, 畳込み層を 2 層しかもたないプレイヤーよりずっと良い性能を示した.

4. 第 1 層の解析

畳込みニューラルネットワークでは, その第 1 層が入力の特徴抽出として働くことが多い. 本研究では, まず, 畳込み層の第 1 層のフィルタを可視化してどのように特徴抽出しているか解析する.

第 1 層に含まれる典型的なフィルタを表 3 に示す. 左の列は, 畳込み層が 2 層であるネットワークの第 1 層のフィルタ 6 つを示す. 右の列は, 畳込み層が 5 層であるネットワークの第 1 層のフィルタ 6 つを示す. パラメータの値が似ているフィルタを選んで表に掲載していることに注意せよ. 第 1 層のフィルタの大きさは $2 \times 2 \times 16$ であるが, 図では縦方向に対象範囲における位置 (上の行から順に左上, 右上, 左下, 右下) を, 横方向にタイルの値 (左の列から順に, 空白, 2, 4, ...) を並べている. パラメータの値が正であれば青色に, 負であれば赤色に色付けされており, 色の濃さが絶対値の大きさを表す. 各図の左のフィルタ番号は, 後の解析で参照するために付けてある.

この表より, 特徴抽出の働きをするフィルタがあることが確認できる.

- 畳込み層が 2 層であるネットワークにおけるフィルタ #74 と畳込み層が 5 層であるネットワークにおけるフィルタ #198 は, 対象範囲の左上が 4 のタイルであり, 左下が 4 のタイルでない (畳込み層 2 層のネットワークでは, 右上も 4 でない) という特徴を抽出する.
- 畳込み層が 2 層であるネットワークにおけるフィルタ #363 と畳込み層が 5 層であるネットワークにおけるフィルタ #213 は, 対象範囲 (右下を除く) に 8 のタイルがあるという特徴を抽出する.
- 畳込み層が 2 層であるネットワークにおける

表 3 第 1 層のフィルタの可視化

畳込み層 2 層のネットワーク														畳込み層 5 層のネットワーク																				
	-	2 ¹	2 ²	2 ³	2 ⁴	2 ⁵	2 ⁶	2 ⁷	2 ⁸	2 ⁹	2 ¹⁰	2 ¹¹	2 ¹²	2 ¹³	2 ¹⁴	2 ¹⁵		-	2 ¹	2 ²	2 ³	2 ⁴	2 ⁵	2 ⁶	2 ⁷	2 ⁸	2 ⁹	2 ¹⁰	2 ¹¹	2 ¹²	2 ¹³	2 ¹⁴	2 ¹⁵	
# 74	NW	0.19	-0.09	0.57	-0.68	-0.66	-0.68	-0.66	-0.67	-0.51	-0.05	-0.02	-0.04	-0.07	-0.13	-0.65	-0.67	NW	-0.11	-0.20	0.61	-0.47	-0.54	-0.56	-0.56	-0.43	-0.27	-0.38	-0.33	-0.33	-0.45	-0.21	-0.08	-0.32
	NE	0.06	-0.18	-0.93	0.16	0.17	0.12	-0.06	0.08	0.01	0.03	0.14	0.10	0.11	0.08	0.20	0.00	NE	-0.00	-0.04	0.01	0.02	0.00	-0.04	-0.08	-0.04	-0.01	0.00	-0.01	-0.02	-0.07	-0.13	-0.11	-0.10
	SW	0.25	-0.06	-0.97	0.23	0.07	0.20	-0.03	-0.08	0.03	0.14	0.15	-0.06	-0.48	-0.87	0.06	-0.96	SW	-0.02	0.15	-0.68	0.13	0.02	0.09	0.02	0.02	0.03	0.01	0.04	0.00	-0.02	-0.06	-0.02	0.01
	SE	0.40	-0.30	0.42	-0.80	-0.15	0.08	0.06	-0.02	-0.04	0.05	0.06	0.13	-0.11	-0.10	-0.09	-0.01	SE	0.11	0.12	0.10	0.07	0.15	0.08	0.13	0.08	0.08	0.17	0.11	0.11	0.14	0.14	0.13	0.17
#363	NW	0.50	-0.17	-0.37	0.59	-1.11	-0.92	-0.43	-0.21	-0.20	-0.11	-0.09	-0.10	-0.16	-0.11	-0.08	-0.76	NW	0.31	-0.33	-0.66	0.40	-0.73	-0.63	-0.61	-0.62	-0.73	-0.66	-0.62	-0.61	-0.62	-0.66	-0.68	-0.55
	NE	-0.12	-0.66	-0.56	0.59	-0.41	0.05	-0.01	0.02	0.05	0.06	-0.12	0.05	-0.04	-0.06	-0.04	-0.08	NE	-0.30	-0.21	-0.21	0.36	-0.27	-0.04	0.00	-0.01	0.01	-0.02	-0.03	-0.05	0.01	-0.03	0.02	-0.02
	SW	-0.18	-0.60	0.07	0.46	-0.63	0.07	-0.10	-0.17	-0.05	-0.07	0.03	0.16	0.11	0.14	0.04	0.18	SW	-0.27	-0.19	-0.30	0.28	-0.03	0.02	-0.01	-0.01	0.00	-0.05	-0.01	-0.01	-0.01	-0.08	-0.07	-0.06
	SE	0.27	-0.21	0.41	-1.07	-0.04	-0.18	-0.16	-0.06	-0.06	-0.05	-0.01	0.10	0.10	0.05	0.06	-0.03	SE	0.07	0.07	0.05	0.01	0.03	0.03	-0.01	0.04	0.06	0.08	0.00	0.01	0.07	0.07	0.08	0.09
# 17	NW	-1.18	-0.41	-0.10	0.05	-0.06	-0.19	-0.58	0.00	0.32	0.51	0.45	0.50	0.52	0.39	0.36	0.42	NW	-0.54	-0.78	-0.36	-0.49	-0.78	-0.78	0.01	0.25	0.33	0.37	0.33	0.32	0.29	0.31	0.33	0.30
	NE	1.37	-0.68	-0.83	-0.16	-0.28	-0.12	0.08	0.16	0.48	0.42	0.50	0.42	0.41	0.49	0.39	0.44	NE	0.16	0.05	0.00	0.07	-0.00	-0.80	0.16	0.28	0.40	0.34	0.33	0.30	0.29	0.29	0.33	0.30
	SW	0.60	0.41	0.24	-0.06	-0.12	-0.06	-0.55	-0.68	-0.76	0.21	0.04	-0.01	0.06	0.11	0.01	0.05	SW	0.05	0.04	0.01	0.10	0.26	0.36	-0.11	-0.23	-0.02	-0.08	-0.06	-0.01	-0.08	-0.03	-0.03	-0.05
	SE	0.51	0.47	0.42	-0.28	-0.17	-0.13	0.07	-0.14	-0.14	-1.12	-0.37	0.01	-0.13	-0.13	-0.10	-0.19	SE	0.28	0.25	0.24	0.28	0.33	0.25	0.11	-0.06	0.06	0.17	0.20	0.18	0.19	0.20	0.19	0.15
#432	NW	-0.18	-0.47	-0.03	-0.07	-0.03	0.05	0.15	-0.15	-0.11	-0.31	-0.05	0.25	0.25	0.44	0.45	0.50	NW	0.66	-0.43	-0.64	-0.70	0.26	0.42	0.27	0.22	0.18	0.16	0.11	0.08	-0.12	0.17	0.16	0.04
	NE	-0.54	-0.43	-0.30	0.14	0.14	-0.06	0.33	0.34	0.58	0.70	0.71	0.49	0.51	0.17	0.20	0.30	NE	-0.27	-0.11	0.00	0.20	0.08	0.31	0.27	0.23	0.20	0.21	0.17	0.05	0.18	0.10	0.07	0.13
	SW	0.56	0.13	-0.13	-0.46	-0.39	-0.21	0.03	0.02	0.23	0.24	0.32	0.29	0.47	-1.62	-0.94	-0.40	SW	0.33	0.28	0.27	-0.00	-0.71	0.22	0.02	0.04	-0.04	-0.19	-0.46	-0.16	-0.44	-0.54	-0.63	-0.45
	SE	0.85	0.14	0.15	0.34	0.09	0.03	-0.21	-1.27	-1.29	-1.30	-0.57	-0.63	-0.31	-0.12	-1.07	-0.99	SE	0.25	0.16	0.14	0.01	-0.56	-0.07	-0.11	-0.21	-0.17	-0.21	-0.23	-0.15	-0.14	-0.55	-0.69	-0.53
#120	NW	0.43	0.59	-0.70	0.03	-0.49	-0.51	-0.56	-0.45	-0.40	-0.34	0.08	0.18	0.08	-0.11	-0.21	-0.13	NW	0.29	-0.52	-0.50	0.21	0.30	-0.42	-0.48	-0.59	-0.60	-0.26	-0.44	0.01	0.06	-0.50	-0.04	0.51
	NE	-0.15	-0.25	-0.30	-0.43	-0.01	0.10	0.33	0.32	0.47	0.57	0.48	0.40	0.45	0.34	0.29	0.09	NE	-0.45	-0.32	0.51	0.37	-0.00	-0.05	0.46	0.50	0.55	0.52	0.44	0.39	0.32	-0.45	-0.45	-0.13
	SW	0.83	0.37	-0.07	-0.09	-0.37	-0.42	-1.32	-1.35	-1.23	-1.34	-1.28	-0.30	-0.43	-0.38	-0.41	-0.41	SW	-0.31	-0.59	0.08	0.22	-0.06	-0.55	-0.16	-0.57	-0.57	-0.54	-0.58	-0.54	-0.24	-0.60	-0.57	-0.43
	SE	0.56	0.19	-0.40	-0.29	-0.13	-0.54	-0.58	-0.33	0.00	0.04	0.14	0.27	0.05	0.13	0.15	-0.02	SE	-0.62	0.08	0.18	-0.17	-0.32	-0.66	0.03	-0.01	0.06	-0.01	0.07	0.01	-0.04	0.10	0.11	-0.08
#154	NW	1.00	0.04	0.03	0.16	0.08	-0.49	-0.44	-1.05	-1.02	-0.66	-0.01	0.66	0.35	-0.52	-1.05	-0.74	NW	0.10	-0.08	-0.15	-0.20	-0.16	-0.86	-0.93	-0.81	-0.75	-0.56	0.08	-0.74	0.63	-0.45	-0.61	-0.53
	NE	0.00	0.38	0.33	-0.27	-0.09	-0.16	-1.17	-0.65	-1.15	0.07	0.42	0.60	0.56	-1.14	-1.14	-1.09	NE	0.38	0.18	0.11	0.02	0.01	-0.29	-0.84	-0.73	-0.63	-0.76	0.14	0.05	0.64	-0.98	-1.05	-0.92
	SW	0.28	0.41	-0.04	-0.02	-0.06	-0.44	-1.23	-1.12	-1.21	-1.17	-1.19	0.21	0.51	-0.07	0.41	-0.02	SW	0.32	0.22	-0.15	-0.05	-0.54	-0.25	-0.71	-0.23	-0.71	-0.85	-0.20	0.15	0.66	-0.34	-1.00	-1.05
	SE	0.03	-0.34	0.10	0.20	0.50	0.23	0.24	0.16	0.16	0.29	0.43	0.08	-1.13	-0.22	1.08	0.04	SE	0.44	0.35	0.29	0.26	0.28	-0.31	-0.57	-0.54	-0.39	-0.21	-0.83	-0.60	0.48	-0.05	-0.94	-0.78

フィルタ#17と畳込み層が5層であるネットワークにおけるフィルタ#93では、上の行に大きな値をもつタイルがあり下の行に小さな値をもつタイル（または空きマス）があるという特徴を抽出する。

- 畳込み層が2層であるネットワークにおけるフィルタ#432と畳込み層が5層であるネットワークにおけるフィルタ#98も上記と同様の特徴を抽出するが、これらは下行に空きマスがあるとより強く反応する。

一方で、抽出される特徴を説明することが困難なフィルタもある。畳込み層が2層であるネットワークにおけるフィルタ#154は、対象範囲の左上、右上、左下のマスに、大きな値をもつタイル（2048または4096）か小さな値をもつタイル（2または4）があるという特徴を抽出する。しかしながら、ゲーム2048においては、大きな値のタイルと小さな値のタイルはまったく異なるものであ

る。著者らは、畳込み層が2層であるネットワークのフィルタ#154は、独立な2つ（もしくはそれ以上）のフィルタが重ね合わされているのではないかと考える。畳込み層が5層であるネットワークでは、そのような重ね合わされたようなフィルタは少ない。

畳込み層が2層のネットワークの第1層と畳込み層が5層のネットワークの第1層を比較して気付いたことがもう1点ある。それは、畳込み層が5層のネットワークのほうが、第1層のパラメータの値の絶対値が小さい（可視化したものがより白く見える）ことである。定量的に解析するため、フィルタのパラメータ数を横軸にとり、絶対値が閾値を超えているパラメータ数が横軸の値より多いフィルタ数を図3にプロットした。このグラフから、畳込み層が5層のネットワークでは絶対値が小さなパラメータを多く持つフィルタが多いことが確認できる。そのようなフィルタは、より局所

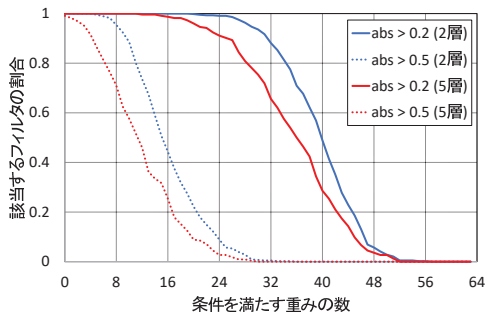


図 3 重みの値の分布に基づく第 1 層のフィルタの比較

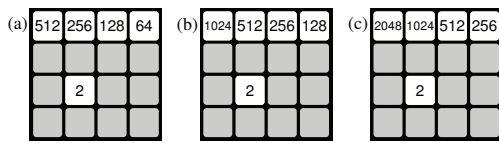


図 4 汎化能力の解析に用いるゲーム状態

的な特徴をうまくとらえていると考える。同様のグラフを畳込み層の数が 3 と 4 の場合にもプロットしたが、それらは層の数が 5 の場合にとっても近かった。したがって、この特徴抽出が畳込み層 2 層で性能がよくない理由のひとつになると考える。

5. 汎化能力の解析

ゲーム 2048 において、N タプルネットワークを用いたアプローチによって優れたコンピュータプレイヤーが得られることはすでに示されている。しかしながら、ニューラルネットワークを用いることで、その汎化能力によって良いプレイヤーができるのではないかと著者らは考えている。本節では、先行研究で作成した畳込みニューラルネットワークが汎化能力をもっているか解析する。

ゲーム 2048 における汎化といっても複数の観点がありうる。本研究で調査する汎化能力は、連続するタイルの値を 2 倍（もしくは 0.5 倍）にした場合の計算のロバスト性についてである。図 4 (a)–(c) に示す 3 つのゲーム状態を考える。人間のプレイヤーであれば、これらの 3 状態に「最大のタイルが角にあり、それ以外の大きなタイルが降順に辺に沿って並んでいる」というパターンが共通していることを容易に認識できる。しかし、コンピュータプレイヤーはその特徴を特別に実装してい

ない限り、そのようなパターンを捉えてプレイすることは容易ではなく、実際、N タプルネットワークではタイルの値が異なるゲーム状態は独立に扱われる。そこで本研究での質問は、「畳込みニューラルネットワークを用いたプレイヤーにおいて、これらのゲーム状態の計算に共通する部分が存在するか」である。

この質問に答えるため、畳込みニューラルネットワークの結果から逆方向に解析を行う。第 2 節の Heuristics 1 と 2 より、図 4 (a)–(c) に示すゲーム状態において「下」を選択するのは最も悪い手である。したがって、ニューラルネットワークの「下」に対応する計算結果から始めて、ニューラルネットワークの入力に向かって逆畳込みを行う。本研究における逆畳込みは以下の方法で行う。

- (1) 畳込み計算において、入力の値とパラメータとの積をそれぞれ計算する。
- (2) それらの積の絶対値和を計算する。
- (3) 各入力の「貢献率」を、入力の値とパラメータとの積の絶対値の、絶対値和に対する比と定義する。その貢献率の上位を抜き出して解析する。

例えば、畳込み層が 2 層であるニューラルネットワークに対して図 4 (b) の盤面を入力として与えたとする。「下」の手に対応する全結合層の計算結果（ソフトマックスの前）は、 -5.89 であった。全結合層の直前の値 ($4 \times 4 \times 436$ 通りある)のうち、パラメータとの積の絶対値の大きなものから順に並べたものが表 4 の入力要素の列である。またその貢献率を表 4 の図 4 (b) の列に示している。最も貢献度の高い入力要素は (0,0) の位置のフィルタ 163 の結果であり、その貢献率は 15.4% である。次に、貢献度の高い入力要素を 28 個とり、同様にして畳込み第 2 層の逆畳込みを行った結果が表 5 の入力要素および図 4 (b) の列である。ここで、貢献度の計算において、全結合層の貢献度との積をとり入力要素ごとに和をとっている。これにより、図 4 (b) の盤面に対する計算において、畳込み第 1 層の結果のうちフィルタ 17, フィルタ 120, フィルタ 432 が重要な役割を果たしていることが分かる。

表 4 畳込み層 2 層のネットワークにおいて、「下」から始めて全結合層まで逆畳込みを行った結果。括弧内は累積。

入力要素	図 4 (a)	図 4 (b)	図 4 (c)
(0, 0)#163	9.1% (9.1%)	15.4% (15.4%)	7.7%(7.7%)
(1, 1)#285	12.5% (21.6%)	13.3% (28.7%)	11.6%(19.3%)
(1, 0)#163	15.4% (37.0%)	12.4% (41.1%)	15.4%(34.7%)
(2, 0)#340	4.8% (41.8%)	9.6% (50.7%)	24.7%(59.4%)
(1, 0)#275	4.7% (46.5%)	8.1% (58.8%)	4.7%(64.1%)
(0, 1)#285	4.3% (50.8%)	4.6% (63.4%)	4.0%(68.1%)

表 5 畳込み層 2 層のネットワークにおいて、「下」から始めて畳込み第 2 層まで逆畳込みを行った結果。括弧内は累積。

入力要素	図 4 (a)	図 4 (b)	図 4 (c)
(1, 0)#17	2.7% (2.7%)	4.3% (4.3%)	2.8% (2.8%)
(1, 1)#120	2.6% (5.3%)	2.5% (6.8%)	1.9% (4.7%)
(0, 0)#17	2.2% (7.5%)	2.2% (9.0%)	1.5% (6.2%)
(1, 0)#432	1.5% (9.0%)	2.0% (11.0%)	1.8% (8.0%)
(2, 0)#17	1.4% (10.4%)	2.0% (13.0%)	5.1% (13.1%)
(2, 0)#432	1.6% (12.0%)	1.9% (14.9%)	3.2% (16.3%)
(1, 1)#418	1.7% (13.7%)	1.8% (16.7%)	1.3% (17.6%)
(2, 1)#209	1.7% (15.4%)	1.6% (18.3%)	1.3% (18.9%)
(1, 1)#252	1.8% (17.2%)	1.5% (19.8%)	1.3% (20.2%)
(0, 0)#432	1.1% (18.3%)	1.3% (21.1%)	1.0% (21.2%)
(2, 1)#137	1.1% (19.4%)	1.2% (22.3%)	1.6% (22.8%)

表 6 畳込み層 5 層のネットワークにおいて、「下」から始めて全結合層まで逆畳込みを行った結果。括弧内は累積。

入力要素	図 4 (a)	図 4 (b)	図 4 (c)
(0, 1)#10	34.5% (34.5%)	34.4% (34.4%)	30.8%(30.8%)
(0, 0)#116	20.9% (55.4%)	23.2% (57.6%)	25.1%(55.9%)
(2, 0)#116	23.0% (78.4%)	22.5% (80.1%)	22.6%(78.5%)
(1, 0)#116	18.4% (96.8%)	18.7% (98.8%)	19.5%(98.0%)

表 7 畳込み層 5 層のネットワークにおいて、「下」から始めて畳込み第 5 層まで逆畳込みを行った結果。括弧内は累積。

入力要素	図 4 (a)	図 4 (b)	図 4 (c)
(1, 2)#173	7.6% (7.6%)	7.4% (7.4%)	6.8%(6.8%)
(1, 1)#126	4.6% (12.2%)	4.5% (11.9%)	4.2%(11.0%)
(0, 1)#126	4.4% (16.6%)	4.3% (16.2%)	4.0%(15.0%)
(1, 1)#134	3.8% (20.4%)	3.7% (19.9%)	3.5%(18.5%)
(0, 1)#181	3.1% (23.5%)	3.0% (22.9%)	2.8%(21.3%)
(2, 1)#173	3.0% (26.5%)	2.9% (25.8%)	3.0%(24.3%)
(0, 1)#134	2.9% (29.4%)	2.8% (28.6%)	2.7%(27.0%)
(0, 0)#54	1.9% (31.3%)	2.6% (31.2%)	3.1%(30.1%)
(0, 2)#106	2.3% (33.6%)	2.3% (33.5%)	2.1%(32.2%)
(2, 0)#54	2.4% (36.0%)	2.2% (35.7%)	2.4%(34.6%)
(0, 1)#79	2.2% (38.2%)	2.2% (37.9%)	2.1%(36.7%)

(a)		4096	2048	128
		4	16	32
	2	16	32	16
	8	2	2	4

(b)	2	2	2	
	16	16	4	4
	32	64	128	256
	2048	4096	2048	1024

図 5 人手で設定したゲーム状態と実際にプレイされたゲーム状態

同様の逆畳込みを図 4 (a) と (c) に対して行い、その結果を表 4 と 5 に合わせて示している。ただし、それらの結果は図 4 (b) に対する結果の貢献度順に並べられていることに注意せよ。また、畳込み層が 5 層であるネットワークについて同様に逆畳込みを全結合層と畳込み第 5 層について行った結果を表 6 と 7 に示す。これらの表においても、結果は図 4 (b) に対する結果の貢献度順に並べられている。

表 4 の結果より、3 つのゲーム状態のいずれにおいても、貢献度上位の 4 つの入力要素は同じであり、それらによって値の 40% 以上が決まっている。このことから、畳込み層 2 層のニューラルネットワークにおいても汎化能力をある程度もっていることが確認できた。また、畳込み第 2 層の入力要素 (表 5) をみると、フィルタ 17 と 432 の結果の貢献度が高い。これらは盤面の第 1 列が大きな値であり第 2 列が空白という特徴を捉えており、それらのゲーム状態に対する我々の理解に一致する。

畳込み層 5 層のニューラルネットワークの場合 (表 6)、全結合層の入力要素 4 つによって 96% 以上の値が決まっている。これは、畳込み層 5 層のニューラルネットワークがゲーム状態に共通するパターンをより正確に捉えていることを意味していると考えられる。畳込み層 5 層の場合についても同様に畳込み第 1 層まで逆畳込みを行うことを試みたが、表 7 に示すように多くの入力要素が結果に貢献していたため本研究では畳込み第 5 層までで断念した。

6. 畳込みニューラルネットワークによる不適切な手の選択

6.1 人手で設定したゲーム状態に対して

図 5 (a) に示す人手で設定したゲーム状態を考

える。第 2 節の Heuristic 1 より、人間プレイヤーは「左」を選択してはどうかとまず考えるはずである。しかしながら、畳込み層 2 層のネットワークは

[上: 0.243, 右: 0.405, 下: 0.104, 左: 0.248]

と右が最も良いと返した。畳込み層 5 層のネットワークにおいても、

[上: 0.470, 右: 0.470, 下: 0.000, 左: 0.060]

と上または右が良いと返した。

いずれの畳込みニューラルネットワークにおいても、「左」を良い手として返さなかった理由として 2 つ考えた。1 つ目の理由は、畳込みニューラルネットワークが捉えることのできる特徴が制限されていることである。Heuristic 1 を適用するには、4096 が最大の値であることを知らなくてはならない。しかし、畳込み層が 2 層もしくは 5 層ではその特徴を捉えることができず、したがって Heuristic 1 を直接実行することができなかったのではないかと考える。2 つ目の理由は、訓練データに含まれるバイアスによるものである。訓練データに含まれるプレイログにおいて、空白マスに向けて大きな値のタイルを動かすことは非常にまれであり、したがって畳込みニューラルネットワークがそのような手がそもそも良い手であると考えなかったのではないかと考える。

6.2 実際にプレイされたゲーム状態に対して

畳込み層 5 層のネットワークを用いて 20 ゲームを行い、ゲーム状態を 77,253 状態集めた (各ゲームの最初の 100 状態は除いた)。それらの状態について、畳込み層の数が 2, 3, 4, 5 のネットワークを用いて手の選択を行い、選択された手がどれだけ一致するかを調べた。その結果、22,807 状態 (29.5%) では 4 つのプレイヤーが全て同じ手を選択したものの、5,915 状態 (7.7%) では畳込み層 2 層のネットワークのみが異なる手を選択していた。

図 5(b) に示すのは、畳込み層 2 層のプレイヤーのみが異なる手を選択したゲーム状態のひとつである。このゲーム状態からの最善手は「左」であり、畳込み層 3, 4, 5 層のネットワークでは最善手を返した。しかしながら、畳込み層 2 層のネッ

トワークは最悪手である「上」を返した。

畳込み層2層のネットワークが「上」を選択した理由を、前節と同様に逆畳込みによって調査した。その結果、畳込み第1層の結果のうち位置(0,3)においてフィルタ154が返した値が最も貢献度が高いことが分かった。表3に示すとおり、畳込み層2層のネットワークの畳込み第1層のフィルタ154では、次の場合に正の値を返す。

- 左上, 右上, または左下が小さなタイル(2, 4)である。
- 左上, 右上, または左下が大きなタイル(2048, 4096)である。
- 右下が4~1024のタイルである。

もし、図5(b)のゲーム状態において、第4行の2048と4096が小さな値であれば確かに「上」を選択するのは合理的な手である。しかし、この場合では2048と4096のタイルによってこのフィルタの結果が正となり、それによって不適切な手が返されたのではないかと考える。畳込み層2層のネットワークのみがこのような不適切な手を選択した理由として、ネットワークが浅いため高次元なパターンを捉えることができず、結果として複数の(本来別に扱わなければならない)特徴を混合して認識するようになったのではないかと考える。

7. 関連研究

2048のコンピュータプレイヤーにおいて、Nタプルネットワークを評価関数として、そのパラメータを強化学習により調整するというアプローチが最も成功している。Nタプルネットワークはデザインと実装が単純であるため、パラメータを増やすことでプレイヤーの性能を向上することが容易である[7], [11]。また、複数ステージ化と呼ばれる技術を使ってNタプルネットワークを拡張することも可能である[16]。

しかしながら、Nタプルネットワークには弱点もあり、そのひとつが汎化能力をもたないことである。Nタプルネットワークにおけるパラメータは独立であるため、Nタプルネットワークがゲームの高次元の特徴を捉えることができない。この弱点を補う方法として、重み促進(weight promotion)

[4], [8]と呼ばれる手法を導入することが提案されている。

2048のコンピュータプレイヤーにおいて、ニューラルネットワークを用いたコンピュータプレイヤーはそれほどうまくいっていない。2048のコンピュータプレイヤーにニューラルネットワークを用いる研究のうち、出版された結果はGueiら[3]によるものが最初である。そこでは、大きさ 2×2 のフィルタを用いる畳込みニューラルネットワークと大きさ 3×3 のフィルタを用いるものが試されている。強化学習によってパラメータ調整を行った結果、 2×2 のフィルタを用いるもので平均得点約11,400点、 3×3 のフィルタを用いるもので平均得点約5,300点であった。畳込みニューラルネットワークによる別のコンピュータプレイヤーとしてtjweiによるものがある。その畳込みニューラルネットワークは、多数のフィルタを持つ2層の畳込みニューラルネットワークである。既存の強いプレイヤーのプレイログを用いる教師あり学習の場合、平均得点が85,351であったことが報告されている。

8. まとめ

本論文では、著者らの先行研究[5]において開発した畳込みニューラルネットワークによる2048プレイヤーの内部動作を解析した。まず、畳込み第1層のフィルタを可視化し、ゲーム状態の特徴抽出がなされていることを確認した。次に、特定の盤面から逆畳込みによってニューラルネットワークを逆方向に解析した。

本研究で得られた発見のうち、最も重要なものはニューラルネットワークプレイヤーの汎化能力についてである。畳込み層2層のネットワークと畳込み層5層のネットワークのいずれの場合にも、類似の盤面に対して共通する計算経路が存在することが確認され、したがってある程度の汎化能力をもつことが分かった。また、畳込み層が2層のネットワークに対する解析結果より、そのプレイヤーの性能があまりよくなかった理由についても明らかにできた。

今後の課題として、畳込みニューラルネットワー

クの構造と大きさを変えることによるプレイヤーの性能への影響についてより詳細に調べることがある。本研究の解析によって得られた知見が、今後のニューラルネットワークプレイヤーの開発に貢献できることを期待する。

参考文献

- [1] Cirulli, G.: 2048, <http://gabrielecirulli.github.io/2048/> (2014).
- [2] David, O. E., Netanyahu, N. S. and Wolf, L.: DeepChess: End-to-End Deep Neural Network for Automatic Learning in Chess, *International Conference on Artificial Neural Networks and Machine Learning (ICANN 2016)*, pp. 88–96 (2016).
- [3] Guei, H., Wei, T., Huang, J.-B. and Wu, I.-C.: An Early Attempt at Applying Deep Reinforcement Learning to the Game 2048, *Workshop on Neural Networks in Games* (2016).
- [4] Jaśkowski, W.: Mastering 2048 with Delayed Temporal Coherence Learning, Multi-Stage Weight Promotion, Redundant Encoding and Carousel Shaping, *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 10, No. 1, pp. 3–14 (2018).
- [5] Kondo, N. and Matsuzaki, K.: Playing Game 2048 with Deep Convolutional Neural Networks Trained by Supervised Learning, *Journal of Information Processing* (2018). Under revision after conditional accept.
- [6] Lai, M.: Giraffe: Using Deep Reinforcement Learning to Play Chess, Master’s thesis, Imperial College London (2015). *arXiv*, vol. 1509.01549v1.
- [7] Matsuzaki, K.: Systematic Selection of N-tuple Networks with Consideration of Interinfluence for Game 2048, *Proceedings of the 2016 Conference on Technologies and Applications of Artificial Intelligence (TAAI 2016)*, pp. 186–193 (2016).
- [8] Matsuzaki, K.: Developing 2048 Player with Backward Temporal Coherence Learning and Restart, *Proceedings of Fifteenth International Conference on Advances in Computer Games (ACG2017)*, pp. 176–187 (2017).
- [9] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M.: Playing Atari With Deep Reinforcement Learning, *NIPS Deep Learning Workshop* (2013).
- [10] Moravčík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., Davis, T., Waugh, K., Johanson, M. and Bowling, M. H.: DeepStack: Expert-level artificial intelligence in heads-up no-limit poker, *Science*, Vol. 356, No. 6337, pp. 508–513 (2017).
- [11] Oka, K. and Matsuzaki, K.: Systematic Selection of N-tuple Networks for 2048, *Proceedings of 9th International Conference on Computers and Games (CG2016)*, Lecture Notes in Computer Science, Vol. 10068, Springer, pp. 81–92 (2016).
- [12] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T. and Hassabis, D.: Mastering the game of Go with deep neural networks and tree search, *Nature*, Vol. 529, No. 7587, pp. 484–489 (2016).
- [13] Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K. and Hassabis, D.: Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm, *arXiv*, Vol. 1712.01815 (2017).
- [14] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T. and Hassabis, D.: Mastering the game of Go without human knowledge, *Nature*, Vol. 550, pp. 354–359 (2017).
- [15] Szubert, M. and Jaśkowski, W.: Temporal Difference Learning of N-Tuple Networks for the Game 2048, *2014 IEEE Conference on Computational Intelligence and Games*, pp. 1–8 (2014).
- [16] Wu, I.-C., Yeh, K.-H., Liang, C.-C., Chang, C.-C. and Chiang, H.: Multi-Stage Temporal Difference Learning for 2048, *Technologies and Applications of Artificial Intelligence*, Lecture Notes in Computer Science, Vol. 8916, pp. 366–378 (2014).
- [17] Yakovenko, N., Cao, L., Raffel, C. and Fan, J.: Poker-CNN: A Pattern Learning Strategy for Making Draws and Bets in Poker Games, *arXiv*, Vol. 1509.06731 (2015).