

# テトリスを題材にしたスケールダウンを利用した学習手法の開発

青木 勢馬<sup>1,a)</sup> 橋本 剛<sup>1</sup>

**概要:** 人間が評価関数を作らないゲーム AI の汎用的な学習手法が成果を挙げているが、複雑なゲームではその手法が通用しない。テトリスもその複雑なゲームの一つで、学習で良い特徴を得るためにはランダム操作で横一列を揃える必要があり、それには隙間なく埋める操作を連続して行う必要がある。連続して隙間なく埋める操作を行える確率は低い上に、隙間を埋めない操作を途中で行った場合、横一列を揃えられる状態から遠ざかってしまう。その難しさのため、人間が評価関数を選定しない手法では AI の作成が上手く行っていない。本研究では、テトリスはゲームのサイズをスケールダウンしても同じルールで遊べるという性質に注目し、スケールダウンしたサイズでの学習を行い、その結果を元のサイズでの学習に利用するという手法を提案した。この手法はテトリスだけでなく、スケールダウンが可能なその他のゲームにも適用出来る。この手法を評価するための実験を行ったところ、提案手法を用いた学習は、それを用いない学習と比べて劣るという結果が得られた。また、提案手法の性能が劣った原因についての考察を行った。

## A Study on Learning Method Using Scale Down about the Tetris

SEIMA AOKI<sup>1,a)</sup> TSUYOSHI HASHIMOTO<sup>1</sup>

**Abstract:** A general learning method of game AI that human does not select evaluation function have succeeded, but this method can't apply in complicated games. Tetris is also one of the complicated games. To obtain good features in learning, it's necessary to align one row horizontally by random action, and it's necessary to continuously act operations without gaps. The probability that continuously act operations without gaps is low, and if it acts operations making gaps in the middle, it gets away from state that it can align row horizontally. Because of this difficulty, the game AI that human does not select evaluation function han't been created yet. In this research, we focused on the feature that Tetris can play with the same rule even if the size of game is scaled down, we proposed a method that learn with scale down size, and using the result for learning at the original size. This method can be applied not only to Tetris but also to other games that can be scaled down. We experimented to evaluate this method, we found that learning using the proposed method is inferior to learning without using it. Also, we investigated the cause of poor performance of the proposed method.

### 1. はじめに

近年、ゲーム AI の研究が盛んで、様々なゲーム AI 作成手法が提案されている。かつては人間が評価関数などを作成して行う学習が主流だったが、AI に評価関数を自動で学習させる汎用的な学習の研究が増えつつある。特に、DQN (Deep Q-Networks) と呼ばれる、既存の手法である強化学習に、深

層ニューラルネットワークを適用したゲーム AI の学習が成功を収めている [1].

しかし、この汎用的な学習で AI の作成がされているのは、ブロックくずしなどの簡単なゲームに限り、その他の複雑なゲームでは汎用的な学習が上手くいっていない。汎用的な学習が上手くいっていない理由として、DQN はランダム試行から学習を行うため、状態数が膨大な場合は良い特徴を得るのに時間がかかる、という点が考えられる。テトリスもその一つのゲームであり、汎用的な学習が上手くいっ

<sup>1</sup> 松江工業高等専門学校  
National Institute of Technology, Matsue College  
<sup>a)</sup> s1601@matsue-ct.jp

ていない。ヒューリスティックを用いたテトリス AI の作成は既に報告されているが、評価関数を人間が選定しているため、汎用性は低いと言える [2].

また、難しい学習対象に対し、簡単な状態のみで学習を行い、次第に難しい状態を増やしていくといった、段階を設けて学習を行う手法が提案されており、良い性能を示している [3]. 設けられた段階のことをカリキュラムと呼び、それを用いた学習をカリキュラム学習という。テトリスなどのゲーム AI にも適用出来ると考えているが、カリキュラムの作成は人間が選定して行う必要があることが多く、テトリスも例外でない。このような手法でテトリスの AI を作成しつつ、他のゲームにも同じカリキュラムで学習が行えるような手法を作成したいと考えた。

そこで本研究では、テトリスがゲームの状態をブロック単位で扱える、という点に注目し、ゲームの状態をスケールダウンして学習を行う手法を考案する。テトリスはゲームの状態を縦 20 ブロック、横 10 ブロックといった配列の情報で表すことができ、これを縦 5 ブロック、横 5 ブロックといったスケールダウンを行ってもゲームとして成立する。スケールダウンを行った盤面なら、ランダム試行でも比較的良い特徴を得やすいと考えられるため、学習が上手くいくと考えた。そしてスケールダウンした状態での学習結果を、元の大きさの状態での学習に利用すれば、元の大きさでの学習も上手くいくと考えた。このような手法はテトリス以外の、ゲームの状態をブロック単位で扱えるようなゲームにも適用可能であると考えた。

本研究ではテトリスを題材にして、評価関数などを人間が選定せずに、盤面のスケールダウンを利用した学習手法を考案する。テトリスにおいて人間レベルのプレイを行える AI の作成をし、なおかつスケールダウンが可能な他のゲームにも適用出来る学習手法を考案するのが本研究の目的である。

## 2. テトリスのルール

### 2.1 一般的なテトリスのルール

テトリスは、4つのブロックで構成されたピースを盤面に隙間なく敷き詰めていくゲームである。盤面の横一列を全てブロックで敷き詰めると得点が入り、そのブロックが消滅する。ゲームの様子を図 1 に示す。

ゲームが始まると盤面の上部に 4つのブロックで構成されたピース (以下ピース) が出現する。ピースの種類には 7種類ある。図 2 にそれぞれの形状を示す。プレイヤーはこのピースを操作することが出来る。行える操作は右移動、左移動、下移動、右に 90° 回転、左に 90° 回転である。ピースがそれ以上、下に移動出来ない位置まで来たら、そのピースがその場所に設置され、また新たなピースが盤面上部出現する。出現するピースの種類はランダムに与えられる。ピースを設置した際、横一列が全てブロックで

埋まった列があった場合、その列のブロックが全て消去され、それより上にあるブロックが消した列の分だけ下に落ちる。また、消した列に応じて得点が入る。ピースが出現する位置がブロックで埋まった場合、それ以上ピースが出現出来ないため、ゲーム終了となる。ゲーム終了とならないようにピースを操作し列を消去していくことで、得点を増やしていくことがこのゲームの目的である。



図 1 テトリスゲーム

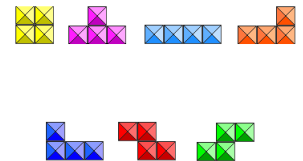


図 2 ピースの種類

### 2.2 本研究で用いるテトリスのルール

本研究で用いるテトリスのルールは、ほとんどが一般的なテトリスのルールと同じであるが、行える操作が、「現在のピースに対して、ピースの x 座標、および回転の角度を指定する」のみとなる。ピースの x 座標および回転の角度を指定したら、十分に高い位置からその状態でピースを下に移動させて、それ以上、下に移動出来ない位置まで来たら、ピースをその場所に設置する。

## 3. 関連研究

2012年に Alex Krizhevsky らが多層畳み込みニューラルネットワークを画像認識の分野に適用させ、既存の手法を上回る成果を挙げた [4] ことから、深層学習に関する研究が広まり始めた。本研究ではこの多層畳み込みニューラルネットワークをゲーム AI に適用させる。

Volodymyr Mnih らによる Deep Q-Networks を利用した汎用的ゲーム AI 作成 [1] で、人間が手を加えず、AI がゲーム画面を受け取るだけで特徴量や評価関数を自動で学習する手法が成功を取めた。いくつかのゲームにおいて、人間の熟練者のスコアを上回る結果が出たが、そのような結果が出たのは学習が行いやすいシンプルなゲームのみであった。この研究で対象となったのは Atari 2600 と呼ばれる家庭用ゲーム機でのいくつかのソフトで、人間の熟練者を上回るスコアが出たのは、例えば図 3 に示す Pong や図 4 に示す Breakout といったゲームである。これらのゲームはスコアを得るために、プレイヤーが操作する棒を玉に当てる必要があるが、ランダムなプレイでも比較的玉に当

たりやすく、学習は比較的行きやすいと言える。しかし、それ以外のゲームではスコアを得るためのステップ数が多いなどの要因で、ゲームの複雑度が上がり、この手法での学習が成功していない。

テトリスの AI 作成については、様々な手法が先行で研究されている。Istvan Szita らによる研究 [5] では、ノイズを導入したクロスエントロピー法を用いてテトリス AI の学習を行い、その AI は 1 ゲームあたり 30 万ライン消去するという性能を示した。しかし、縦列の高さの最大値、隣り合う縦列同士の高さの差などの、盤面の評価関数に用いる特徴を人間が選定し組み込んでいたため、汎用性は低いと言える。また、Victor Gabillon らによる研究 [2] では、近似動的計画法をテトリスに適用して AI の作成を行い、その AI は 1 ゲームあたり 5100 万ライン消去するという性能を示した。しかしこちらも、盤面の評価関数に用いる特徴を人間が選定し組み込んでいたため、汎用性は低いと言える。

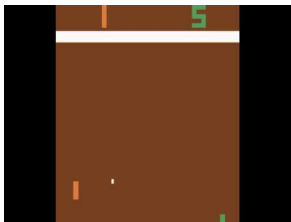


図 3 Pong

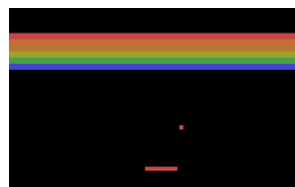


図 4 Breakout

#### 4. 学習でテトリス AI を作る困難さ

ゲーム AI での強化学習は最初はランダムに操作して、上手くいったパターンを学習していくというものである。ブロックくずしなどの簡単なゲームでは、ランダムに操作を行っても比較的少ない試行回数でスコアが増える。つまり上手くいくパターンへ楽にたどり着くことが出来る。しかしテトリスにおいては、隙間を埋める操作を連続して行う必要がある。図 5 に例を示す。テトリスでスコアを得るためには横一列をブロックで埋める必要があるが、そのためには図のように複数のピースを連続して隙間が埋まるような位置に置く必要がある。ランダム試行だとこのように上手くいくことはほとんどない。その上、図 6 のように一つでも隙間を埋めない操作をすると、隙間の上が塞がってしまい、スコアを得る一歩手前の状態だったものが、スコアを得るには程遠い状態になってしまう。これらのことから、汎用的学習でテトリス AI を作るのは困難であるといえる。

#### 5. スケールダウンを行うテトリスでの学習

通常のテトリスは盤面サイズが縦 20 ブロック、横 10 ブロックであるが、この盤面サイズを例えば縦 5 ブロック、



図 5 隙間を埋める操作を連続して行った状態



図 6 隙間を埋めない操作を行った状態

横 5 ブロックと変更しても、同じゲームのルールを適用出来る。そしてスケールダウンを行い、小さい盤面サイズのテトリスで学習を行うことを考える。盤面サイズを小さくした場合、横一列を揃えるために必要なピースの数が少なくなる。そのためランダム操作でも比較的横一列を揃えやすくなるため、学習が上手く行くと考えられる。しかし、ある一定以上の盤面サイズになると、ランダム操作で横一列が揃いにくくなり、学習が上手く行かなくなる。そこで、図 7 のように、学習が上手く行かない大きい盤面サイズについて、学習が上手く行った小さい盤面サイズでの結果を利用する手法を考える。例えば、縦 5 ブロック、横 5 ブロック (5\*5) の盤面サイズで学習が成功し、縦 7 ブロック、横 7 ブロック (7\*7) の盤面サイズで学習が失敗した場合、5\*5 での学習結果を 7\*7 での学習に利用する、という手法である。5\*5 の学習で隙間を埋める操作を学習出来ているので、それを上手く利用出来れば、7\*7 の盤面サイズでも隙間を埋める操作が発生しやすくなり、学習が上手く行くと考える。そして学習が成功した 7\*7 の結果を次はさらに大きい盤面サイズに利用して、段々と大きい盤面サイズでの学習を行っていけば、最終的には通常サイズでの学習が出来ると考える。

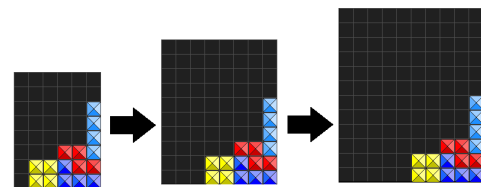


図 7 小さいサイズの学習結果を大きいサイズでの学習に利用していく

小さい盤面サイズでの学習結果を大きい盤面サイズに利用する具体的な手法として、設置場所を小さい盤面サイズでの学習結果から決定する、という手法を提案する。図 8 に例を示す。図 8 上部が盤面にオレンジ色のピースが落ちてきたという状態を示している。この盤面から、小さい盤面を複数個抜き出す。抜き出した複数の領域について、オ

レンジ色のピースが降ってきたと見なし、その状態に対する行動および行動の評価値を、小さい盤面での学習結果を利用してそれぞれ算出する。その中で最も評価の高い行動を、元の盤面サイズでの行動として選択する。抜き出した盤面での良い操作は、元の盤面でも良い操作となると考えた。このようにして盤面サイズを少しずつ大きくして学習していけば、通常サイズで良いプレイをする AI が作れると考えた。

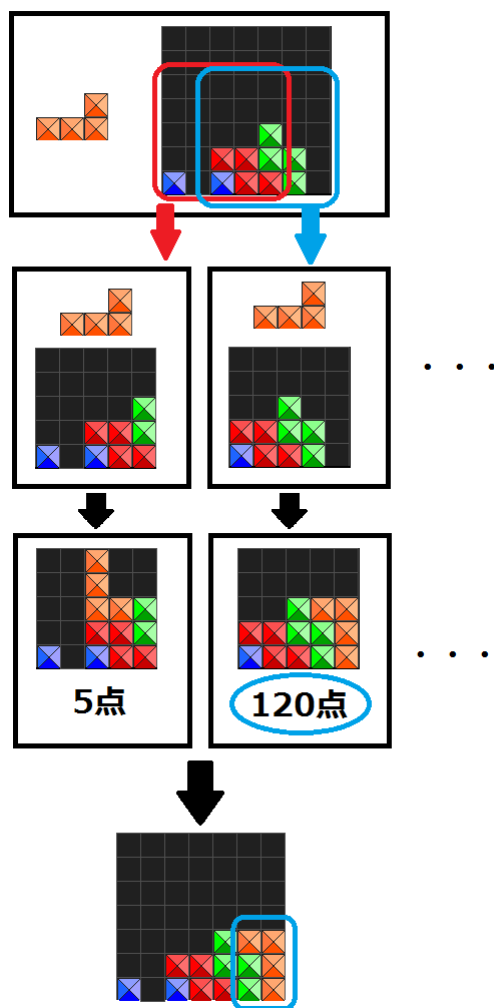


図 8 学習結果利用の例

## 6. 学習実験

(1) 盤面サイズ 5\*5, (2) 盤面サイズ 7\*7, (3) 盤面サイズ 5\*5 の学習結果を盤面サイズ 7\*7 に利用, の 3 種類の条件で Q 学習を行った。(3) の学習実験は、(1) の学習結果を利用して行った。学習で与える報酬は、横列を揃えた時に列の数×1点、ゲームオーバーになった時に-10点を設定した。学習の入力として与えたのは、盤面の状態と現在操作しているピースの種類と次に操作するピースの種類である。出力は、ピースの x 座標と回転の向きの全ての組み合わせについての Q 値であり、その中の操作可能なものの中

から Q 値が最大のものをアクションとして行う。アクションの選び方は実験 (1) と (2) については  $\epsilon$ -greedy 法で行い、 $\epsilon$ には 0.05 を設定した。実験 (3) については、Q 値が最大のものをアクションとして行う確率を 0.90、ランダムでアクションを選択する確率を 0.05、学習結果を利用してアクションを選択する確率を 0.05 となるように設定した。

## 7. 実験結果

5\*5 の盤面サイズで学習を約 4000 万ステップ行ったところ、1 ゲームあたり 4.66 ライン消去出来る AI が出来た。全てのピースについてランダムで設置場所を決めた場合の性能は 1 ゲームあたり 0.098 ラインであったため、強化学習による性能の向上が確認出来た。7\*7 の盤面サイズでの学習実験の結果を図 9 に示す。グラフの横軸が学習ステップ数、縦軸が 1 ゲーム当たりの消去ライン数を示す。図の実線が 5\*5 の学習結果を 7\*7 に利用したもので、破線が何も用いずに 7\*7 の学習を行ったものである。図 9 より、3000 万ステップ時点では、何も用いずに 7\*7 の盤面サイズで学習を行った方が性能が高くなった。

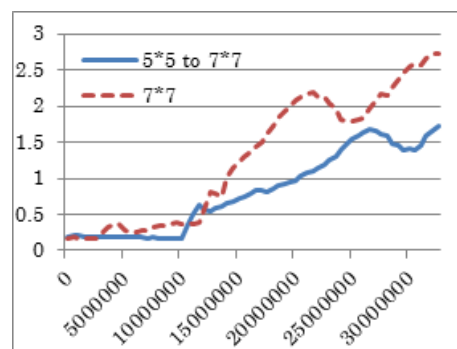


図 9 盤面サイズ 7\*7 での学習結果

## 8. 考察

提案手法である、小さい盤面の学習結果を大きい盤面での学習に利用したものの方が、3000 万ステップ時点での性能が低かった。このようになった原因として、図 10 の緑のピースのように、抜き出した盤面での置き方で評価値が高くても、元の盤面では良い置き方ではないことなどが考えられる。しかし、これとは逆に、抜き出した盤面での評価値が低い置き方は、元の盤面でも悪い置き方であることが言えると考えた。図 10 では、抜き出した盤面で列が揃っているため評価値が高くなるが、元の盤面では列が揃っていないため良い置き方ではない。一方、抜き出した盤面で列が揃っていないならば、必ず元の盤面でも列が揃わない。このことから、小さい盤面の学習結果を大きい盤面での学習に利用するとき、良い(可能性がある)置き方を小さい盤面から得るのではなく、悪い置き方を小さい盤面から得て、その際に評価値が低くなるように学習を行えば良いと

考えた。

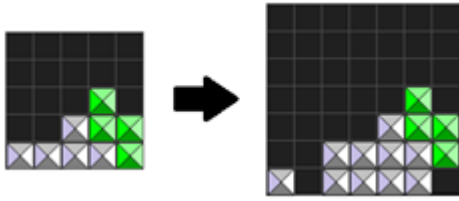


図 10 良い置き方ではない例

## 9. おわりに

前章で考察したように提案手法には問題点があり、提案手法を用いないものに比べて性能が劣った。この問題点を回避するために、小さい盤面から良い特徴を得るのではなく、悪い特徴を得て、それを回避するように学習するという手法を考案した。現在、その手法での学習実験を行っている。この手法でテトリスについて良い性能が示せれば、テトリス以外のスケールダウンが可能なゲームについても学習実験を行い、同様の手法で良い性能が出せるかを確認する。

### 参考文献

- [1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller: Playing Atari with Deep Reinforcement Learning, In NIPS Deep Learning Workshop, 2013.
- [2] Victor Gabillon, Mohammad Ghavamzadeh, Bruno Scherrer: Approximate Dynamic Programming Finally Performs Well in the Game of Tetris, In Advances in Neural Information Processing Systems, 2013.
- [3] Y. Bengio, J. Louradour, R. Collobert, and J. Weston: Curriculum learning, In ICML, 2009.
- [4] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton: ImageNet Classification with Deep Convolutional Neural Networks, In Advances in Neural Information Processing Systems 25, pp. 1106-1114, 2012.
- [5] István Szita, Andras Lorincz: Learning Tetris Using the Noisy Cross-Entropy Method, Neural computation, 18(12): 2936-2941, 2006.