Vol.2018-GI-39 No.4
2018/3/2

弾幕シューティングゲームを対象とした汎用的学習法

野村直也[1,a)]　橋本剛[1]

概要：近年のゲーム AI は AI 自身が評価関数を生成する汎用的な学習法が成果をあげており, 代表的なものとして Deep Q Network(DQN) などがある. これらの手法は様々なゲームに適用可能であるが, 画面の情報量が多いものや操作が複雑なゲームでは学習が進まないという問題があり, さらに広い範囲のゲームに適用できる手法が必要とされる. 本研究では複雑なゲームの一つとして弾幕シューティングゲームを取り上げ, このゲームに適用可能な学習手法を提案することで, 更に汎用的な手法について考察する. 弾幕シューティングゲームにおける人間のプレイの性質に着目すると, 人間は画面全体を見ておらず, 初心者は視野が狭く, 上達するに連れ視野が広くなっていくという性質があると考えた. ゲーム序盤は観測範囲が狭く, 徐々に観測範囲が拡大していくという性質を学習システム内に組み込むことで複雑なゲームに適応させる. 本研究では観測範囲を狭くして学習の効率化が図れるか実験を行い確認した。観測範囲を狭めて学習させたところ, 従来の手法よりも高いスコアを獲得した. また, 観測範囲の変化量の妥当性や, その他のゲームへの適用について考察を行った.

General machine learning for Bullet Hell Game

Nomura Naoya[1,a)]　Hashimoto Tusyoshi[1]

Abstract: Deep Q Network (DQN) is applicable to various games, but it has a problem that it tends to fail in complicated and difficult games. In this research, we pick up barrage STG even among complicated games, and tried to succeed in learning by incorporating the characteristics of human play in this game into DQN. In this game, human has a feature that are not seeing the entire screen and change the field of vision according to the amount of bullets. By incorporating the mechanism into DQN, we obtained better results than the conventional method.

1. はじめに

今日のゲーム AI は進化が目覚しく、さまざまな手法が提案されている。個別のゲームに個別の方策を取る特化型の AI に加え、様々なゲームに適用できる汎用的な AI の開発が主流になりつつある。ボードゲームでは囲碁の AlphaGo[1] が 2015 年に人間のプロに勝利するという成果を獲得した。AlphaGo は囲碁のみに特化した AI であったが、AlphaGo を改良した AlphaZero は人間の手を参考にせず、囲碁のルールしか知らない状態からプロを破ったものを上回るまでに成長した [2]。さらに AlphaZero は内部のアルゴリズムを変更せずに将棋とチェスにおいても強くなり、従来の AI を超える成長を見せた [2]。ボードゲームではある程度の汎用的な機械学習が成功していると言える。

ビデオゲームでは Deep Q Network(DQN)[3] と呼ばれる AI の作成手法が成功を収めている。2015 年には複数のゲームにおいて、ルールを一切知らない状態から人間よりも高いスコアを獲得するまでの成長を見せている [4]。これまでのゲーム AI は個々のゲームに固有の方策を用いていたが、DQN は同じ手法で複数のゲームを攻略するという高い汎用性を見せた。さらに DQN の中で用いられている手法を基にした Asynchronous Advantage Actor-Critic(A3C) はゲームによっては DQN を超えるスコアを獲得している [5]。

しかしゲームによっては人間を超えるほど強くなるもののブロック崩しやスペースインベーダーのような単純なものばかりであり、複雑なゲームでは長い時間をかけて学習

[1] 松江工業高等専門学
National Institute of Technology. Matsue College
[a)] s1719@matsue-ct.jp

させてもうまくいっていない。ビデオゲームにおける汎用的機械学習では、AI はゲーム画面をみてそこにランダムな操作を試すことで成長していく。しかしテトリスでは各ブロックの組み合わせが多すぎるためランダムな試行では学習に時間がかかりすぎてしまう [6]。弾幕 STG のようなゲームでも大量の弾とアクションの多さが原因で学習が上手くいかないことが報告されている [8]。しかし人間は情報量が多いゲームでもより短い時間でゲームの特徴を掴むことができる。弾幕 STG における人間のゲームプレイでは、人間は画面全体は見ておらず自機の周辺に視点が集中することが多いと考えた。自機の周辺が生死に直結するため、見る範囲も自機周辺になることが推測できる。この性質を用いて学習を行うことで状態数を減らして効率的に学習させることができると考えた。そこで本研究では学習における入力画像を自機周辺に切り取り学習を行う手法を提案する。自機周辺の重要な情報のみを切り出すことで、ランダム試行であっても学習が上手くいくと考えた。この手法はプレイヤーが画面上のキャラクターを操作して遊ぶゲームにおいて適用可能だと考える。本研究では汎用的学習法として、より簡単で弾幕 STG での実験事例が多い DQN を用いる。DQN の入力画像を自機周辺の小さいものにすることで学習が上手くいくか実験し、結果について考察した。

2. 弾幕シューティングゲーム

シューティングゲームのジャンルに弾幕シューティングゲーム (以降　弾幕 STG) がある。図 1 に弾幕 STG の一つである東方弾幕風の画面図を示す。シューティングゲームの二大要素の「撃つ」と「避ける」の中でも「避ける」ことを追求したゲームである。このジャンルのゲームではその他のシューティングゲームと比べて以下のような特徴が見られる。

- 弾の量が多い
- 自機の当たり判定領域は非常に小さい
- 自機の移動スピードを調節できる機構が備わっていることが多い

このゲームは有志による学習例が存在しているものの、いずれもステージの序盤を突破できず、ゲームを攻略できなかったという結果に終わっている。[7][8]。弾幕 STG はスペースインベーダーと同じシューティングゲームであるが、学習の結果には大きな差が出ている。本研究では、弾幕 STG を対象とした、さらに汎用的な学習法について考えていく。

3. 汎用的機械学習

現在世界で活用されている主流の AI は特化型 AI と呼ばれ、特定の分野や環境の知識を学習させ、その中のみで活用される。ボードゲームにおいては AlphaGo が囲碁に特化し人間を超える成果を見せている。しかし特化型の AI は特定の領域をまたいだ適用は難しく、様々なものに対応した汎用的な学習手法が求められている。

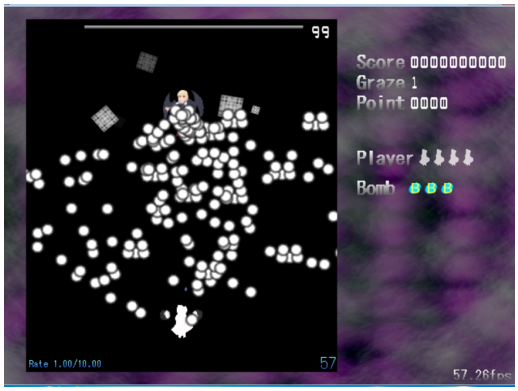

図 1　東方弾幕風

複数のゲームに適応した汎用的機械学習法のひとつに DQN がある。強化学習 (Q 学習) と多層畳み込みニューラルネットワークを組み合わせた深層強化学習とよばれるアルゴリズムで動作する。「Atari 2600」の 49 種類のゲームをプレイし、43 ものゲームで従来の人工知能のスコアを上回り、29 のゲームは人間を超えるスコアを獲得した [4]。中のアルゴリズムを変えずに複数のゲームに対応できるという特徴から注目を浴びた。現在では DQN の手法を活用した A3C と呼ばれる手法も発表され、DQN を超えるスコアを獲得している。

DQN の学習方式には以下のような特徴がある。

- ルールを教えられない
- ゲーム画面全体を入力とする
- 与えられた入力に対してランダムな操作を行う

ゲーム画面を見てランダムに操作を繰り返すことで徐々に成長していく。ゲームのルールにとらわれないためさまざまなものに適用可能であると注目を集めた。

4. 弾幕 STG における DQN の問題点

DQN は複数のゲームで人間以上のプレイを見せ、従来の学習手法以上の汎用性を見せた。しかしあらゆるゲームにも適用可能かというとそうではないことが知られている。DQN が上手くプレイできるゲームは Atari2600 における「BreakOut」や「Pong」などの単純なゲームに限られている。弾幕 STG のような情報量が多く複雑なゲームでは学習が上手くいかないことが過去の実験で報告されている [7][8]。

弾幕 STG において DQN を用いてもゲームを攻略できない原因として、主に 2 つ挙げられる。まず行動の次元が多いことである。DQN はランダムな試行を繰り返すことでゲームを学習していく。行動数が多いとその分学習時間も非常に長くなってしまう。図 2 に「BreakOut」のアクションを、図 3 に弾幕 STG のアクションを示す。「BreakOut」ではアクションの数は「←」、「→」、「stop」の 3 つである。

対して弾幕 STG では、一般的にアクション数が 18 個であることが多い。2016 年の能登による DQN の実験でもアクション数は 18 個であった [8]。１つの状態に対し 18 個ものアクションをランダムで試さなければならないため学習が進みにくくなる。

次にゲーム内の状態数が非常に多いことが挙げられる。図 4 に示すのは「BreakOut」での状態変化の例である。「BreakOut」では画面上での変化は、ブロックの消失、弾１つの移動、バーの移動の３つだけであり、場面数はこれらの変化の組み合わせの数である。対して弾幕 STG では図 5 に示すように画面内に移動する弾が大量に存在し、組み合わせの数が膨大になってしまう。DQN は観測した情報を CNN により分類することでその場面での行動を学ぶ。そのため画面内に球が増えるほど場面数が増えてしまい、現実的な時間内に学習が収束しない。また自機のすぐ近くの生死に関わる弾の状況が違っていてもその他の状況が似ているために同じような評価値が帰ってきてしまう問題もある。弾幕 STG で DQN を用いて学習させるにはこのような問題を解決しなければならない。

能登による DQN の実験では、DQN を改良した DRQN と DoubleDQN を組み合わせたものを用いて実験を行っていた [8]。表に示す学習環境で、6 ステージからなる弾幕 STG をおよそ 30 時間かけて学習させた。その結果、6 つあるステージの中で 1 ステージの序盤をクリアできる程度であり、ゲームクリアとは程遠い結果であったとある。

図 2 BreakOut のアクション

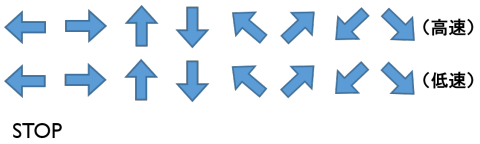

図 3 弾幕 STG のアクション

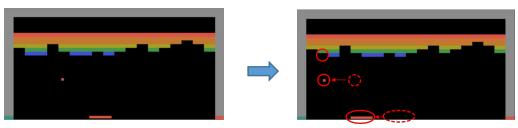

図 4 BreakOut の状態変化

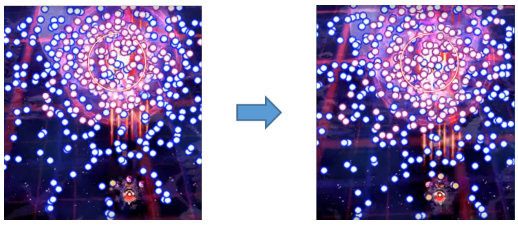

図 5 弾幕 STG の状態変化

表 1 能登の実験環境

OS	Windows10
GPU	GeForce GTX 970
言語	Python2.7.11
学習ライブラリ	Chainer1.6.1
ゲーム	東方甘珠伝

5. 人間はどのように弾を避けているか

本研究では前述した 2 つの問題点のうち、後者である観測情報における問題の解決について考える。

観測情報とは、人間における目からの情報である。そこで弾幕 STG プレイ時の視点を考える。人間は大量の弾が迫ってきたとき、おそらく自機周辺に視点を集中させていると思われる。自機近くの弾が一番生死に直結しているため、視点も自機周辺に視点が集中することが多いと考える。図 6 は人間が見ていると思われるおよその範囲のイメージ図である。

DQN は画面全体を入力として弾を避けていた。つまり画面端の弾まで全て見ながら目の前の弾を避けているということである。しかし人間は目の前に弾があるとき自分から遠い位置にある関係ない弾まで見て避けることはないと思われる。初心者はもちろん、上級者も初心者に比べて視野が広くなってはいるが、関係ない弾は見ずに自機周辺の必要な情報のみを観測して行動を決定していると考えられる。この性質は他のゲームにも同様であると思われ、人間は画面の情報量に対応しきれない場合は自分が十分に処理できる範囲まで視野を狭めると推測できる。

そこで入力を人間のように重要な部分のみに減らすことで、場面の分類を上手く進めることができるのではないかと考えた。

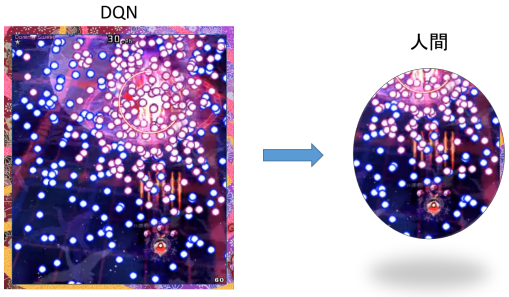

図 6 視点の集中する範囲

6. 提案手法

そこで提案するのは、学習範囲を自分の周りの狭い範囲のみに絞り学習を進め、その学習で得た評価関数を徐々に広い範囲に適用させる手法である。この手法では主に 2 つのステップがあり、画面の中から重要な部分を選び入力画像を小さくして学習させるものと、その場所から徐々に入力とする範囲を広げていくものに分かれる。

まず学習時の入力画像を画面の中の需要な場所に限定させる。図 7 は入力の局所化例である。観測範囲を自機を中心とした座標で考え、図 7 の赤枠のように自機周辺に絞る。弾幕 STG では周辺の弾に当たるかどうかで生死が決まるため人間の視点も集中することが多いと考えられる。また一般に弾幕 STG やマリオのような画面に表示されているキャラクターをプレイヤーが動かして遊ぶゲームは視点がそのキャラクター周辺に集まると推測でき、自機周辺の情報が重要であると考えられる。観測範囲を自機を中心とした座標で考え、自機周辺の情報のみで学習させることで、より少ない時間での効果的な行動の獲得が期待できる。

また重要な部分での学習だけでなく図 8 のように入力を赤枠で示す範囲に拡大することでさらに成長すると考える。弾幕 STG では近くの弾を避けてもその先で周辺全てを弾に囲まれて死んでしまうこともあり、すぐ近くの弾だけではなくある程度広い範囲の弾も把握しなければならない。そのため次のステップとして入力画像を自機周辺に絞り学習させたのちに徐々に学習範囲を広げていく。学習範囲を広げた歳には、狭い学習範囲で得た評価を基に行動させ、小さい入力画像と広い範囲の入力画像を考慮した行動を学ばせる。

本研究では自機周辺の画像での学習に焦点を当てる。ゲーム画面から自機を検出し入力を小さくすることで学習の効率化を図ることができるかを実験により確認する。

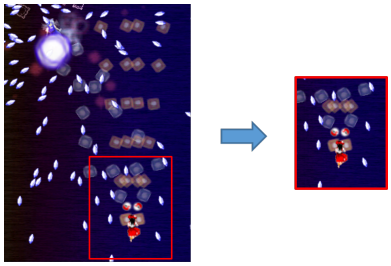

図 7 入力範囲の縮小

7. 実装および実験

7.1 実装

本実験での実験環境を表 2 に示す。

図 9 に入力画像を小さくした DQN による学習の流れを示す。本研究では弾幕 STG 作成用スクリプトと動作ソフトを公開している「東方弾幕風」を用いた。図 1 は東方弾幕風のプレイ画像であり、敵は図 1 のように画面上方からの単発の弾と 5 つの弾が重なった弾をランダムに発射する。エージェントは 2 つのシステムに分割し、それぞれを入力を取得およびゲームを操作するもの (図 9A)、CNN と Q 学習を用いて学習するもの (図 9B) とした。A はまずゲームの画像をスクリーンショットで取得し、Opencv を用いて自機を検出する。そして自機を中心とした 84 ピクセル四方に画像を切り取って学習側である B に送る。B で CNN と Q 学習を組み合わせたアルゴリズムで学習を行い、その評価結果をエージェントである A に送る。A はその評価に対応した操作をゲームに返す。図 9 内の自機検出・画像加工のステップを省略しそのまま B への入力画像としたものが従来の DQN となる。実際に学習を行う B は変更せずに、エージェントである A を変更して実験を行う。3 に CNN や Q 学習におけるパラメータや使用手法を示す。内部で使用したパラメータは先行実験 [7][8] を参考にした。

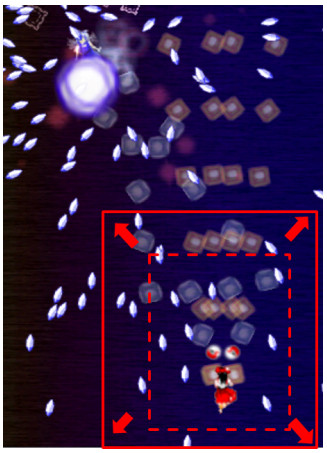

図 8 入力範囲の拡大

表 2 実験環境

OS	Windows7,Ubuntu16.04LTS
GPU	G-Force GTX 1070
言語	Python2.7
学習ライブラリ	Chainer
ゲーム	東方弾幕風

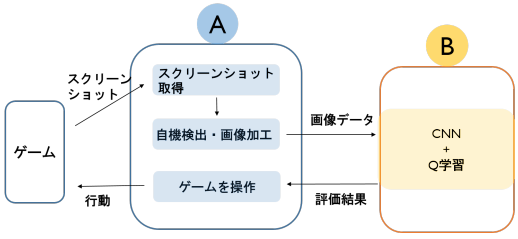

図 9 システム図

表 3　パラメータおよび使用手法

割引率	0.99
学習係数	0.1
行動決定	ε-Greedy 法
活性化関数	ReLU
optimizer	Adam

7.2　実験方法

図 9 に示すループを繰り返すことで DQN の入力画像を小さくして実験を行った。従来の DQN は図 9 内の自機検出・画像加工のステップを省略しそのまま B への入力画像として実験した。従来の DQN および小さい入力画像のどちらもゲーム開始から死ぬまでを 1 回とし、これをそれぞれ 5000 回繰り返した。報酬としてゲーム開始からの生存時間を与え、学習中の報酬遷移を記録・比較した。そして学習後のデータを用いて従来手法・小さい入力での学習それぞれ 10 回プレイさせ生存時間を記録した。また弾幕 STG の初心者 10 人に 10 回ずつプレイさせその生存時間を記録した。なお本研究では観測範囲は拡大せず、入力をゲーム画面全体とせずに重要な部分だけを抽出し情報量を減らすことで場面分類を進めることができるかを検証した。

8.　結果

図 10,11 に従来手法と小さい入力での手法の報酬グラフを示す。またそれぞれの移動平均を図 12、13 示す。横軸がゲームをプレイした回数であり、縦軸がそのときの生存時間である。小さい入力での学習と従来の学習の学習にはどちらも約 40 時間かかった。本研究で対象としたゲームはランダム性が高いため実際の生存時間は誤差が非常に大きいものであった。報酬の移動平均を取り生存時間の推移を確認した。また、図 14、15、16 に従来の DQN・小さい入力の DQN・人間にそれぞれ 10 回プレイさせたときの生存時間のグラフを示す。図 14、15 は従来の DQN、小さい入力の DQN それぞれ 5000 回プレイさせたときの AI にゲームをプレイさせた時の生存時間である。図 16 の横軸は被験者の番号であり、縦軸はそれぞれの被験者の平均の生存時間である。表 4 は図 14、15、16 におけるそれぞれの平均時間である。

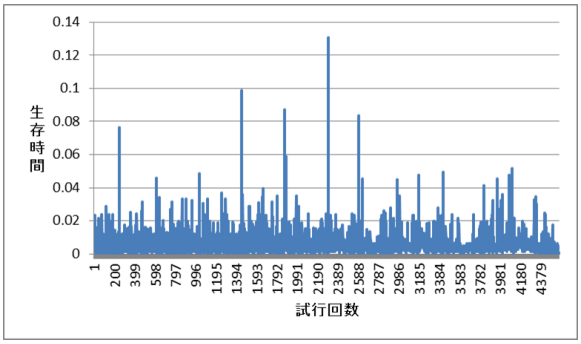

図 10　従来手法 生存時間遷移

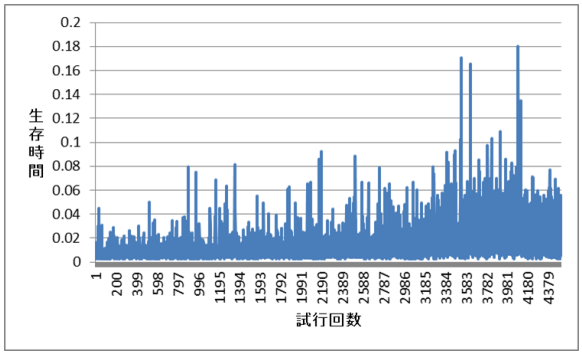

図 11　入力を狭めた学習 生存時間遷移

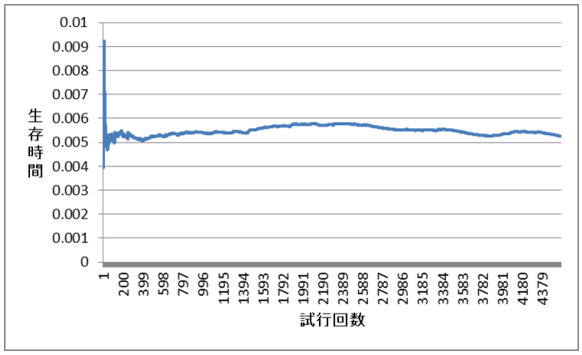

図 12　従来手法 移動平均

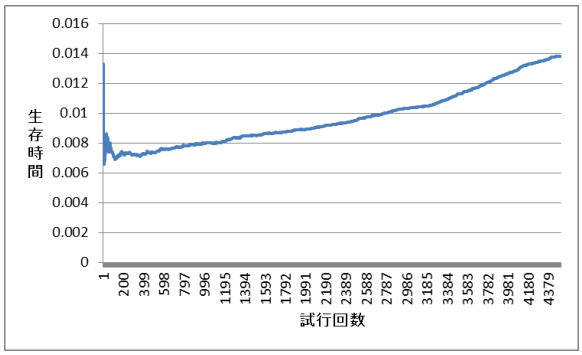

図 13　入力を狭めた学習 移動平均

表 4　各平均生存時間

プレイヤー	平均生存時間 (秒)
従来手法 AI	1.336
入力を狭めた AI	1.996
人間	3.822

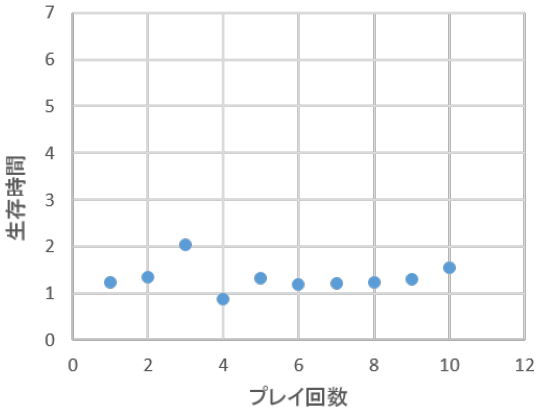

図 14 従来手法 AI の生存時間

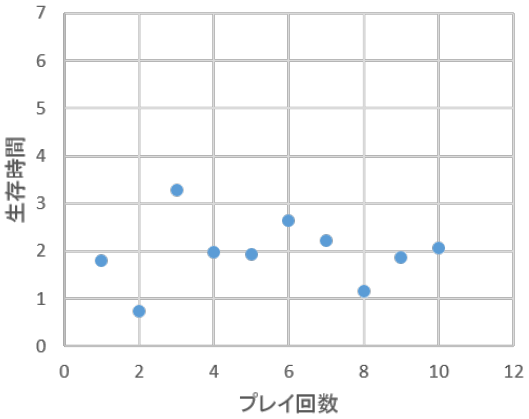

図 15 入力を狭めた AI の生存時間

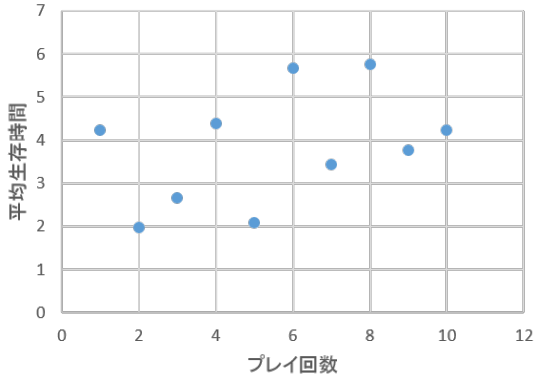

図 16 人間 (初心者) の生存時間

9. 考察

図 10 と図 11 を見てわかる通り、従来手法に比べ提案手法では生存時間が後半になるに従い増えているのが分かる。従来の手法の生存時間の推移はほぼ横ばいであり、成長しているとは言えない。対して提案手法ではわずかながら右上がりになっており、ゲームに上手くなっている。弾のランダム性が非常に高いため安定したスコアは獲得できていないが、学習後半は序盤の 2,3 回の行動は安定して避けられるようになっている。従来手法と提案手法では提案手法が良いプレイを獲得しているといえる。従来手法では画像の分類が上手くいかず、様々な状態に対し同じ行動を繰り返す場面が多くみられた。入力を縮小し情報量を減らすことで状態の分類が上手く進み、より正しい行動を選択できたと考える。

しかし、提案手法の報酬の増え方は非常に緩やかである。図 4 より、人間の初心者にこのゲームをプレイさせたところ平均の生存時間は約 3.8 秒であった。従来手法に比べプレイがうまくなったものの人間と同等のプレイをさせるには、いまだ現実的な時間内に学習を終えることができないといえる。

原因として重要な部分ではあっても狭い範囲の入力だけでは限界があるためと思われる。本研究では自機周辺に入力を絞っただけで、その他の情報は切り捨てている。弾幕 STG は自機のすぐそばの弾を避けてもその先で弾に囲まれて死ぬこともあり、自機周辺だけでなくその先も見て避ける方向を予測しなければいけない。本研究では切り捨てた広い範囲の情報を自機周辺の学習結果と組み合わることでさらに成長すると考える。

10. まとめ

DQN はこれまで以上の汎用的な学習手法であるとして注目を集めている。しかし DQN にはブロック崩しやスペースインベーダーのような単純なゲームのみ学習が上手くいき、ゲームが複雑になり情報量が多くなると上手くいかなくなるという問題点がある。本研究では複雑なゲームとして弾幕 STG を対象とし、この問題点の解決を試みた。この問題点を人間のゲームプレイ時の視点の範囲を参考にして、解決できるのではないかと考えた。具体的には、DQN の入力を縮小し学習を進め、そして徐々に入力を広げていくという方法である。本研究ではこの提案手法の、入力を縮小し学習を進める部分の実装・実験を行った。

結果として、提案手法は従来手法に比べ学習時の報酬は増加した。学習済みのデータを基にゲームをプレイさせたところ、提案手法の AI がより高いスコアを獲得した。しかし学習中の報酬の増え方は非常に緩やかであった。従来手法に比べ上手くプレイする AI を作成できたが、人間のプレイと比較すると未だ十分にゲームを学習できたとはい

えない状態である。観測範囲を徐々に広げて学習させることでさらなる成長が期待できる。

11. おわりに

本研究では DQN の入力を重要な部分的な箇所に絞り、学習の収束速度を速めるシステムを実装した。今後はさらなる広い範囲に狭い範囲の方策を適用させる具体的手法について考察し、実装・実験を行いたい。また弾幕 STG 以外のゲームについても、本研究での手法が有効かどうか検証したい。

参考文献

[1] David Silver et al, Mastering the game of Go with deep neural networks and tree search, Nature 529, 484489 (2016)
[2] David Silver et al, Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm,arXiv:1712.01815 (2017)
[3] Volodymyr Mnih et al, Playing Atari with Deep Reinforcement Learning, NIPS 2013 Deep Learning Workshop (2013)
[4] Volodymyr Mnih et al, Human-level control through deep reinforcement learning, Nature 518, 529533 (2015)
[5] Volodymyr Mnih et al, Asynchronous Methods for Deep Reinforcement Learning, arXiv:1602.01783 (2016)
[6] 青木勢馬,「テトリスを題材にした深層学習によるゲーム AI 強化手法の提案」, 組合せゲーム・パズルプロジェクト第 11 回 研究集会 (2016)
[7] imenurok, 「東方 Project を DeepLearning で攻略した…かった。」, http://qiita.com/imenurok/items/c6aa868107091cfa509c, (2018/1/10 アクセス)
[8] 能登,「深層強化学習による東方 AI」, 第 13 回 博麗神社例大祭 (2016)
[9] Richard S and AndrewG.Barto, MIT Press (1998), 「強化学習」(三上 貞芳・皆川 雅章共訳), 森北出版
[10] Watkins, C.J.C.H, Learning from Delayed Rewards. PhD thesis, Cambridge University (1989)
[11] D.Hebb, The Organization of Behavior, Wiley (1949)
[12] Homma et al, An Artificial Neural Network for Spatio-Temporal Bipolar Patters: Application to Phoneme Classification ,Advances in Neural Information Processing Systems 1: 3140.