深層強化学習を用いたガイスターAI の構築

木村勇太 [1]　伊藤毅志 [1]

概要：　本研究では不完全情報ゲームの一つであるガイスターに深層強化学習を導入し強いプログラムの実現を目指す．予備実験として，通常のガイスターよりも小さい盤面を作り，その環境下で強化学習の自己対戦実験を行った．その結果，ランダムエージェントよりも実力の高いエージェントが作成できることを確認したが，その学習には限界が見られた．相手の駒の推定を行っていないことが要因であると考えた．そこで，完全情報ガイスターを提案することで，相手の駒がわかっているときの学習を行い，相手の駒を推定する手法と組み合わせることで，より強いガイスターAI の構築を目指す．

キーワード：ガイスター，不完全情報ゲーム，深層強化学習

Construction of Geister AI using deep reinforcement learning

YUTA KIMURA[†1]　TAKESHI ITO[†1]

Abstract: In this research, we try to introduce deep reinforcement learning to Geister, one of the imperfect information games, to realize a strong program. As a preliminary experiment, we made a board smaller than a normal Guyster and conducted a reinforcement learning self-play experiment in that environment. As the result, it was confirmed that an agent with higher ability than a random agent could be created, but the learning was limited. We considered that the reason why is that the opponent's piece is not estimated. Therefore, we propose a perfect information Geister to learn when the opponent's piece is known. And we combine it with a method to estimate the opponent's pieces, aiming to build a stronger Geister AI.

Keywords: Geister(Ghost), imperfect information game, deep reinforcement learning

1. はじめに

ゲームは勝敗というわかりやすい形でその性能が評価できるため人工知能の研究の分野として様々な研究が行われてきた．囲碁や将棋などの二人完全情報確定ゲームは長い間ゲーム AI の研究の場として盛んに行われてきたが，最も難しいとされた囲碁の研究でアルファ碁がトッププロに勝ち越すレベルになり，二人完全情報確定ゲームの分野では一つの区切りを迎えている．このような背景から近年では，ゲーム AI の研究の場は多人数ゲームや不完全情報ゲーム，不確定ゲームなどへと広がりを見せている．

本研究で対象とするガイスターは相手の駒の種類がわからない二人不完全情報確定ゲームに分類される．ガイスターは数年前より AI 大会なども開かれるなど新しい研究題材として注目を集めつつあるが，まだ人間に勝てるほど強い AI は存在しない[1]．本研究では，ガイスターAI に深層強化学習を組み込むことでその強化を試みる．

2. ガイスター

ガイスターの盤面は 6×6 のマスで構成され，四隅にはそれぞれ出口が存在している．ゲームで使う駒には相手プレイヤに見えないよう後ろ側に色のついた印がついている．青色の印がついている駒(青駒)と赤色の印がついている駒(赤駒)の 2 種類の駒がある．両プレイヤはそれぞれの種類の駒を 4 つずつ，合計 8 つ所持している．プレイヤはゲームを始める前に，相手にわからないように盤面の手前側中央 2×4 のマスに自由に 8 つの駒を配置する．図 1 は，初期配置の一例である．相手の駒の種類は，見えないようになっている．ゲームは交互に駒を一つずつ動かすことで進行する．手番では自分の駒を上下左右どちらかに 1 マス動かすことが可能で，動かした先に相手の駒が存在する場合，相手の駒を取ることができ，それによってその駒の種類を知ることができる．それぞれのプレイヤは，以下の 3 つの勝利条件を満たすことを目標に交互に駒を動かしていく．

1. *自分の青駒を相手側の出口から脱出させる*
2. *相手の青駒をすべて取る*
3. *自分の赤駒をすべて取らせる*

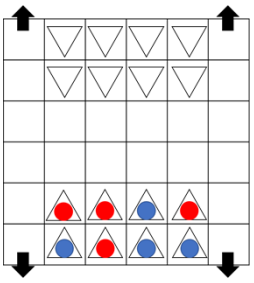

図 1．ガイスターの初期配置例

1 電気通信大学
University of　Electro-Communications, Chofu, Tokyo 182-8585, Japan

3. 関連研究

ガイスターAI の研究として，川上らはミニマックス探索をガイスターAI に導入する手法を提案している[2]. この研究では，考えられる相手の駒の配置の組み合わせすべてに対してミニマックス探索を行い，それぞれの組み合わせで最も評価値の低い手を求めている．各最低評価値の手の中から最も高い手を AI の着手としている．この AI はガイスターAI の大会で優勝するなど AI の中では高い実力を持っているが，初心者の人間相手に対して 2 連敗する程度のレベルであった．

このようにガイスターAI の実力はまだ十分なレベルであるとは言えない．完全情報ゲームでは機械学習によって強いプログラムが実現されているが，ガイスターでは人間の強いプレイヤの棋譜が十分に手に入るわけではないので教師あり機械学習は難しい．そこで，自己対戦を用いた強化学習による手法が考えられる．

ガイスターに強化学習の組み込みを行った研究として佐藤の研究がある[3]．この研究では強化学習の行動価値関数をニューラルネットワークで近似し，自己対戦による学習を行った．勝敗の報酬を得やすくするために，勝敗条件を変更したり，着手に制限をかけたりすることで自己対戦を行った. その結果, 学習を行った AI は序盤の定跡のような手やブラフのような手を指すことができるようになった. しかし，AI 自体の実力は非常に低く，ランダムに手を指すプレイヤとの対戦にも勝てないレベルであった．

ガイスターでは駒の近傍の情報が着手の決定の上で重要になる．しかし，この研究では通常のニューラルネットワークを使用しており，駒の配置の入力は 1 次元による入力を用いていた. ニューラルネットワークを多層構造にし，入力情報を 2 次元にすることでエージェントを強くできる可能性がある．

実際に多層の入力情報を与えて，非常に高い実力を実現したプログラムとしては，Google 傘下の Deepmind 社の AlphaZero がある[4]．このプログラムでは盤面情報の入力から自分の着手の方策と盤面の価値の 2 つの出力を持つデュアルネットワークを作成し，モンテカルロ木探索によるシミュレーションに既存のプレイアウトではなくこれらの出力を用いている．この手法で囲碁，将棋，チェスにおいて既存の有力なプログラムよりも高い実力となることが示されている．ネットワークの更新には自己対戦で得た盤面とその出力結果を用いており，人間の棋譜やルール設定以外のパラメータの調整を必要としない．この点において学習データの入手手段の乏しいガイスターの学習にも適応できる可能性があるが，ガイスターは不完全情報ゲームであるため完全情報ゲームのように探索中に現れる状態に確実性はなく，同様の手法を行うにはそのまま使うことは難しい．

4. 予備実験

4.1 概要

佐藤の研究では行動価値関数をニューラルネットワークで近似し自己対戦を行ってガイスターAI の強化を図ったがあまり強くならなかった．そこで本研究では Deep Q-Network (DQN)を組み込んだガイスターAI を構築し自己対戦を行い, より AI の実力を向上させることを目的とする．今回は自己対戦を通して正常に学習が行われるのかを確認するために，予備実験として学習時間を短縮するために通常の 6×6 の盤面よりも小さい 5×4 の盤面の小路盤のガイスターを定義し，この環境での実験を行った．駒の初期配置は先手後手ともに前に赤駒を 2 つ，後ろに青駒を 2 つとなるように固定した．対戦が長引かないように，100 ターンを超えると引き分けとすることにした．

4.2 実験

この小路盤ガイスターにおいて自己対戦を行うためのエージェントを構築した. 実装には Python およびライブラリの Chainer，ChainerRL を使用した．ネットワークへの入力には 5×5 の 2 値画像を特徴数 3 枚分入力する．特徴は以下のものになる．

1. 自分の青駒の配置
2. 自分の赤駒の配置
3. 相手の駒の配置ととった駒

出力はゲーム中に可能なすべての駒の動き 64 手の中から一つになる．可能なすべての駒の動きは次のように求められる．ガイスターの駒は 4 方向に動くことができるので図 2 のように駒が盤面の角にあるなら 2 方向，辺上にあるなら 3 方向，それ以外の場所にあるなら 4 方向となる．したがって可能なすべての駒の動きは，5×4 のガイスターでは 4(角のマスの数)×2+10(辺のマスの数)×3+6(それ以外のマスの数)×4=62 となり，これに加えて奥の 2 マスから行える脱出の 2 つを加えて 64 手となる．中間層は畳み込み層 3 枚，全結合層 1 枚とした．

この条件で 50 万回の自己対戦を行った．自己対戦では勝敗がついたときに勝者に 1，敗者に-1 の報酬を与えた．このネットワークの出力はゲーム中に現れる手すべてであるため，ネットワークの出力が必ずしも合法手でない場合もある．その際はその時点でゲームを終了とし，合法手を指さなかったエージェントには敗北時と同様に-1 の報酬を与えた．

50 万回の自己対戦が終了した後，完全にランダムな手を指すエージェントとの対戦実験を行った．学習した局数 1 万回から 50 万回までのエージェントそれぞれとランダムエージェントを先後 5000 回ずつの 1 万回対戦させた．このとき構築したエージェントが合法手を選ばなかった場合はランダムな合法手を指すようにした．この条件で勝率と非合法手率（合法手を選べずランダムに指した割合）を記

録した.

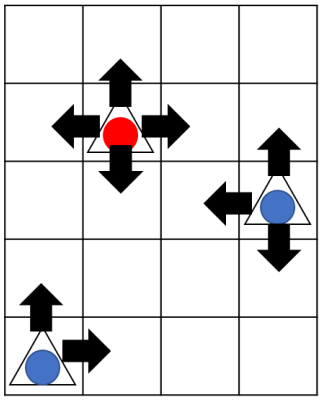

図 2. 可能な駒の動き

4.3 結果

図 3，および図 4 はランダムエージェントとの対戦結果である．それぞれ横軸が学習した局数を表している．図 3 は対局中に選んだ非合法手の割合を表している．図 4 は勝率(勝ち数/勝ち数+負け数)，および勝ち+引き分けの割合である．

学習回数が増えると徐々にランダムエージェントに勝ち越すようになることが確認できた.最終的には勝率は 6 割，勝ち+引き分け率は 7~8 割程度に収束している．これは学習回数 50 万回までの対戦結果を見ても変化がなかった．

4.4 考察

結果より今回の条件では 1 万回以上の対戦でランダムエージェント以上の実力になることが確認された．しかし非合法手率自体は収束しておらず一定の周期で増減している様子が観察されている．

両グラフを比較して考えると，非合法手率が低くなると対戦時の引き分けの数が増えている様子がわかる．これは報酬の設計が理由だと考えられる．報酬は負けた時と非合法手を指した時同じ-1 が与えられる．つまり非合法手率が低いモデルは報酬-1 を減らす手ばかりを学習してしまい，勝利時の報酬 1 につながる手を学習できていないと考えられる．このように非合法手を指してしまった際の結果がネットワークのパラメータへ影響を与えていることを考えると，非合法手を選ばない仕組みでのエージェントの設計を行う必要があると考えられる．

勝率が収束し増加する様子がなく，エージェントの学習は限界に達している可能性がある．この理由の 1 つに今回の実験では相手の駒の推測等を使わず，ネットワークの入力の際にはすべて同じものにしていることが考えられる．実際にネットワークの入力が同じであっても実際の盤面では相手の駒の色によって有利な手にも不利な手にもなりうる．こういった状況があるためにエージェントの強化が妨げられていると考えられる．駒の推測などをネットワークの入力前に行うことで相手の駒の種類も入力に加えるなどのことが必要であると考えられる．

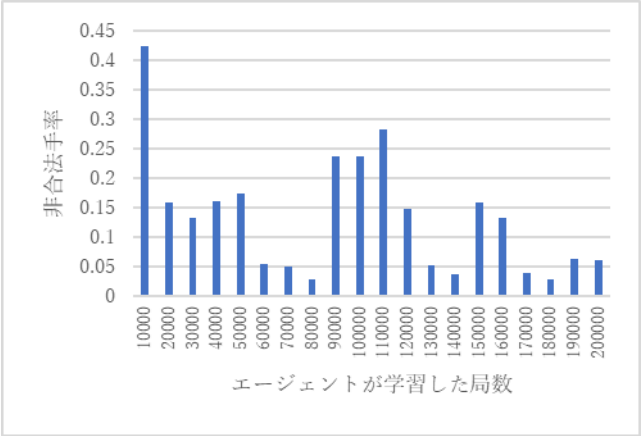

図 3. 非合法手率

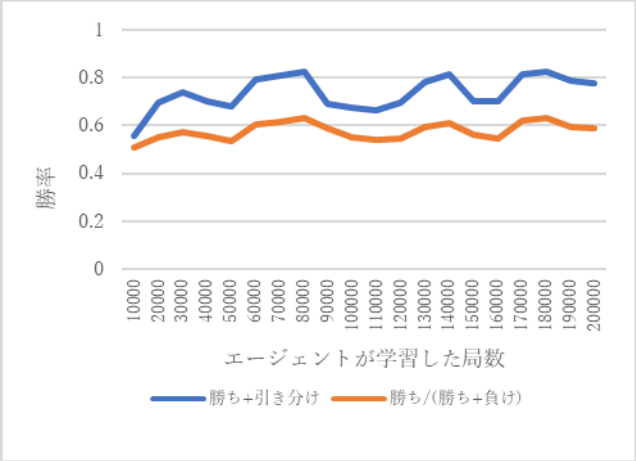

図 4. 勝率

5. 完全情報状態でのガイスターエージェント

5.1 概要

予備実験では深層強学習を用いてランダムエージェントよりも高い実力を持つエージェントを構築することができた.しかし,大幅に勝ち越すことができたわけではなく，学習局数を増やしても実力が上昇することはなかった．これは，非合法手を打たないような仕組みや相手の駒を全部同じものとして入力に与えてしまったことが原因と考えられる．

これらのことを踏まえて相手の駒の色がわかる完全情報状態のガイスターを定義して，その環境で非合法手を打たない仕組みを加えて学習を行うエージェントを作成した．

5.2 目的

予備実験で十分な実力をもったエージェントを構築できなかった理由として非合法手を選んでしまった際の学習まで行ってしまっていること，相手の駒の色を考慮せずに入力に与えている点が考えられた．2 つの解決策として完全情報ガイスターを定義する．完全情報ガイスターは通常のガイスターと同じ 6×6 の盤面であり，違いのある点はお互いの駒の色が公開されているという箇所のみである．

このガイスター上でなら既存の完全情報ゲームで成果を

出した手法を用いることができる．完全情報ゲームで最も有力な手法であるとされる AlphaZero のプログラムでは常に合法手のみを選ぶ仕組みになっているため，非合法手まで学習に加えてしまうことはなくなる．相手の駒色も把握できるので入力の問題も解決できる．

ここでは完全情報ガイスター上で学習を行うエージェントを設計してその実力を検証する．

5.3 エージェントの構築

完全情報ガイスター上で動作を行うエージェントを構築した．予備実験では初期配置は固定としたが，今回は初期配置はランダムに決定するように変更を加えた．この環境で学習を行うエージェントを構築した．

今回は完全情報ゲームとして学習を行うことができるので完全情報ゲームで非常に高い実力を持つ AlphaZero のアルゴリズムを参考にし，以下のように設計した．

ネットワークの設計

このエージェントはポリシー(次の 1 手)とバリュー(盤面の勝敗予測)の 2 つの出力を持つデュアルネットワークを保持している．ポリシーの出力は予備実験同様ありえる全ての手とした．6×6 の場合は 122 手となる．

入力は 6×6 の 2 次元配列が 4 つとなり詳細は以下のようになる

・自分の青駒の配置(6×6 の 2 次元配列)
・自分の赤駒の配置(6×6 の 2 次元配列)
・相手の青駒の配置(6×6 の 2 次元配列)
・相手の赤駒の配置(6×6 の 2 次元配列)

盤面と入力の例を図 5 に示す．

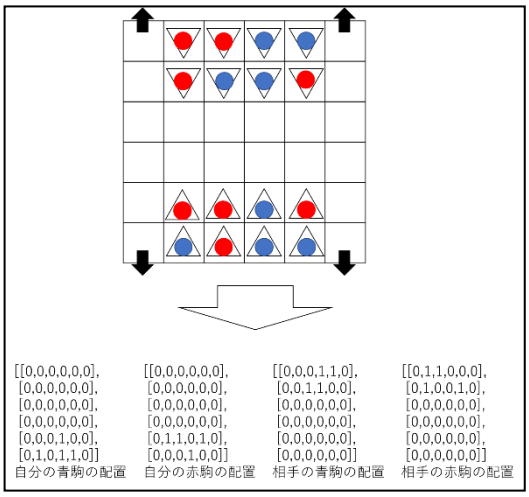

図 5．ネットワークへの入力例

デュアルネットワークの構成は以下のように設定した．

・畳み込み層(3×3 のカーネル 128 枚)
・残差ブロック 8 個
・プーリング層
・ポリシー出力，バリュー出力

モンテカルロ木探索

現在の局面から探索を行う．現在の盤面を根とし，一定回数シミュレーションを行う．シミュレーションでは通常のモンテカルロ木探索のようにプレイアウトを行うのではなく，デュアルネットワークの 2 つの出力を用いてノードの価値と着手確率を決定する．すべてのシミュレーションが終了したら，実際の手を決定する．最も試行回数の高い手がこの探索におけるもっとも有力とさせる手になるが，100%その手を選ぶと毎回同じ盤面で同じ手を打つことになってしまうため，バラつきを与えるためにボルツマン分布を用いて他の手も選ばれるようにした．

モンテカルロ木探索を導入することにより，合法手のみの評価値を取得し，手の出力も合法手のみとすることができるようになった．

5.4 学習サイクル

学習サイクルは次のように設定した．

Step1：デュアルネットワークの作成

重みがランダムなネットワークをベストプレイヤーモデルとして保存．未学習状態であるためポリシー，バリューはランダムな値を出力する．

Step2：自己対戦

ベストプレイヤーモデルを用いたエージェントで自己対戦を *N* 回繰り返し，盤面情報，着手，報酬を 1 セットとした学習データを局面分保存する．報酬は予備実験と同様に勝ち 1 負け-1 引き分け 0 とした．盤面が大きくなったので引き分け条件は 200 手とした．

Step3：パラメータ更新部

Step2 で得たデータを用いてエポック数 *E* 回分学習を行い，モデルを更新する．更新するモデルは Step2 で用いたベストプレイヤーモデルのコピーである．

Step4：新モデル評価部

Step3 でできた新しいモデルと *Step2* で使用していたベストプレイヤーモデルを数対戦させる．新しいモデルが勝ち越した場合はそのモデルを以降のベストプレイヤーモデルとする．*Step2* へと戻る

Step4 の新モデル評価部は AlphaZero では行っていないが，計算資源の都合上 AlphaZero のように多量の対戦を行うことができないので，有効であると判断した．

5.5 対戦実験

5.4 の *Step2* の自己対戦数 N を 1000，*Step3* のエポック数 E を 100 として学習サイクル 1 回行い作られたベストプレイヤーモデルを所持したエージェントを作成した．このエージェントを用いて対戦実験を行った．対戦相手としては，完全情報ガイスター上で動くランダムエージェントと，プレイアウトと UCB1 でモンテカルロ木探索のみで手を決定するエージェントを用意した．

それぞれのエージェントと先後 50 回ずつの計 100 回対戦を行った．

以下の表 1 が対戦結果である．対ランダムがランダムエ

ージェントとの対戦結果，対 MCTS がモンテカルロ木探索エージェントとの対戦結果である．

ランダムエージェントとの対戦ではほぼ勝利しており，ランダムエージェント以上の実力は持っていると考えられる．モンテカルロ木探索エージェントに対しては勝ち越してはいるものの，優位に実力が高いと言えるほどの差は馬われなかった．

表 1．対戦結果

	対ランダム	対 MCTS
勝利	97	53
敗北	3	44
引き分け	0	3

5,5 考察

対戦実験の結果より，ランダムエージェント以上の実力を持つエージェントを構築できたと言える．モンテカルロ木探索エージェントに対しては同等程度の実力となった．この点については学習局数が 1000 と多いとは言えず今後より自己対戦の数を増やしていけば実力が向上する可能性がある．

6. 提案手法

5 章では完全情報ガイスターを定義し，その環境で学習を行うエージェントを構築した．しかし，実際のガイスターの対戦では相手の駒の色がわからないので，このモデルをそのまま使用することはできない．しかし，相手の駒を推定し，エージェント内で完全情報ガイスターと同じ盤面に変更すれば，5 章で構築したエージェントの入力として用いることが可能である．そこで，本章では完全情報ガイスターで作ったモデルを実際の対戦で使用するために駒推定を用いるエージェントを提案する．

相手の駒の推定には末續らの研究で用いた手法を使用する[5]．この推定ではそれぞれの敵駒が'青らしさ'を初期値 0 で保持しており，手番ごとに駒の動きに応じて以下の表 2 にある点数を更新していく．この推測の特徴としては手番が来るたびに新たに推測を行うのではなく，前の盤面までの推測の結果を保持しておき，それを更新していくことで駒を推定していることである．この手法を導入した末續のガイスターAI は GAT2018 ガイスター大会で優勝を収めており，非常に精度の高い推測手法であると言える．
完全情報ガイスターの学習と並行して今後は，この推定手法を用いて通常ガイスターの盤面を完全情報ガイスターと同じ盤面へ変換し，5 章で作成したモデルとそれを用いた探索への入力として渡すことで，着手を決定するエージェントの構築を進めていく．

表 2．'青らしさ'の更新値

駒のふるまい	元の値に対する更新値
前進し，接敵数が増えたが，自分から見て橋の 1 段目または 2 段目に来た	+2.5
前進し，接敵数が増えた	-1.5
横に動き，接敵数が増えた	-1.0
前に移動して，接敵数が 0 になった	+4.0
それ以外の移動で，接敵数が 0 になった	+1.5
接敵数が 0 のまま変わらず，自分から見て 1 段目または 2 段目に来た	+10.0
接敵数が 1 以上であるのに，動かなかった	-1.2

7. 終わりに

予備実験では通常のガイスターより小さい盤面のガイスターを定義して，その環境での深層強化学習エージェントを構築した．自己対戦の結果，エージェントはランダムエージェント以上の実力になることが確認できた．しかし，引き分けの数が多くなるなど，十分な学習結果が得られていない．これは非合法手を打ってしまった時のデータも学習してしまっていること，盤面情報の入力で相手の駒をすべて同じものとして扱ってしまっていることなどが原因と考えた．

その問題を解決するべく，完全情報でのガイスターを定義し，その環境で学習を行った．現状では 1000 局程度の学習でランダムエージェントより遥かに強く，モンテカルロ木探索エージェントと同等の実力を持ったエージェントになった．こちらは以後も学習を続けていき，どの程度実力が向上するかを検証していく．この環境で十分な実力を持っていると判断できれば，提案手法のエージェントに組み込み，実際のガイスターAI との対戦実験やガイスターAI 大会への参加を行う予定である．

謝辞

本研究は JSPS 科研費 18H03347 の助成を受けたものです．

参考文献

[1] ガイスターAI 大会@GAT2019(2019)、<http://www2.matsue-ct.ac.jp/home/hashimoto/geister/GAT/
[2] 川上直人,橋本剛,ガイスターAI の研究,ゲームプログラミングワークショップ論文集,vol2018,pp.35-42
[3] 佐藤佑史,ガイスターにおける自己対戦による行動価値関数の学習,電気通信大学　修士論文,2015.
[4] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis,A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science,Science,362(6419):1140–1144, 2018
[5] 末續鴻輝,織田祐輔,機械学習を用いないガイスターの行動ア

ルゴリズム開発,(GAT2018)

