Vol.2021-GI-45 No.6
2021/3/5

ニューラルネットワークを用いた ガイスターの相手駒色推定とその拡張

寺村 舞童華[1]　松崎 公紀[2,a)]

概要：不完全情報ゲームにおいて隠れた情報を推定することは一般に重要である．二人不完全情報ゲーム「ガイスター」に対しては，これまでにモンテカルロシミュレーションを用いた相手駒色推定手法が提案されている．本研究では，ニューラルネットワークを用いた機械学習により駒色推定を行う手法を提案し，実験的に評価する．単一局面の手に対して駒色推定するネットワークに加えて，それまでに推定した相手駒の駒色を入力に加えたネットワークを実装した．原始モンテカルロプレイヤに対する駒色推定では，一致率 91.2% を達成した．駒色推定を用いるプレイヤと駒色推定を用いないプレイヤを対戦した結果，同一プレイヤから学習した場合に 50.0% から 64.0% へ，異なるプレイヤから学習した場合に 50.0% から 55.6% へ，それぞれ勝率が向上した．

1. はじめに

本研究の対象とするゲームは，二人零和確定不完全情報ゲームの一つである「ガイスター」[1] である．ガイスターの特徴は，各プレイヤが 2 種類の色の駒を持ち，相手駒の色が（その駒を取るまで）分からないことである．

不完全情報ゲームにおいて，隠れた情報を推定することは一般に重要である．栃川ら [4] は，相手駒の駒色の情報と勝率の関係について調べ，ガイスターにおいて駒色推定が重要であることを示した．また，モンテカルロシミュレーションにより駒色推定を行うアルゴリズムが，三塩ら [6] や鴛淵ら [8] により提案・評価されている．一方，ガイスター AI 大会 [2] のプレイヤでは，ヒューリスティックなアルゴリズムによる限定的な色推定が行われている [3], [5]．

本研究では，畳込みニューラルネットワークを用いた駒色推定法を提案し，その有効性を実験的に評価する．具体的には以下の手順で研究を進めた．

(1) 性質の異なる 4 種類のプレイヤを用意し，それらの相互対戦により訓練データ・評価データを生成した（第 3 節）．
(2) ある局面において選択された手から駒色を推定するネットワークを設計し，教師あり学習を行った（第 4 節）．
(3) それまでの駒色推定結果を入力としてとるようネットワークを拡張した．さらに，適切な学習方法により，拡張したネットワークが過去の推定間違いに対して頑健になるようにした（第 5 節）．
(4) このようにして得られた駒色推定ネットワークを用いるモンテカルロプレイヤを実装し，駒色推定の有効性を対戦により評価した（第 6 節）．

本研究で得られた重要な結果を以下にまとめる．

- 単純モンテカルロ法のプレイヤは，駒色の推定が容易であり，畳み込みニューラルネットワークにより 91.2% の一致率が達成された．
- （ランダムを除く）3 つのプレイヤで評価したとき，平均の一致率は訓練データに依らず 73% 程度となった．
- それまでの駒色推定結果を利用した駒色推定ネットワークにおいて，正しいデータのみで学習すると間違いを含む評価データで精度が下がることが確認された．学習の際に意図的に間違いを含めることでこの問題を解決することができ，間違いを 30% 入れて学習したネットワークは間違いを含む評価データに対しても性能向上した．
- 対戦相手と同じ訓練データで学習した駒色推定を用いたとき，勝率が 50.0% から 64.0% へ，もしくは，17.5% から 59.7% へと大きく向上した．また，対戦相手と異なる訓練データで学習した駒色推定を用いた場合でも，勝率が 50.0% から 55.7% へ，もしくは，17.5% から 38.3% へと向上した．これらの結果より，畳み込みニューラルネットワークによる駒色推定が有効であることが確認できた．

[1] 高知工科大学大学院工学研究科
[2] 高知工科大学情報学群
[a)] matsuzaki.kiminori@kochi-tech.ac.jp

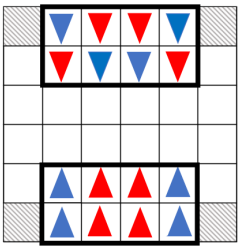

図 1　ガイスターの初期盤面

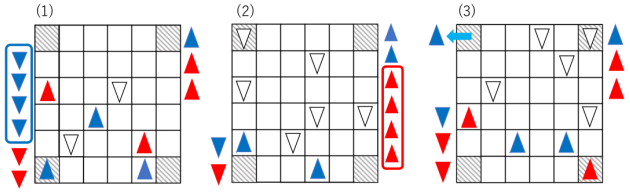

図 2　ガイスターの勝利時の盤面

本論文の構成は以下のとおりである．第 2 節では，ガイスターのルールと本研究で用いたプレイヤについて記述する．第 3 章では，単一局面の手に対する駒色推定のためのネットワークを設計し，その評価を行う．第 4 節では，それまでの駒色推定結果を入力とするようネットワークを拡張し，それに対する学習方法について議論・評価する．第 5 節では，駒色推定ネットワークを用いたモンテカルロ法プレイヤを実装し，駒色推定を用いないモンテカルロ法プレイヤと対戦させる．第 6 節で関連研究の述べ，第 7 章で本論文をまとめる．

2. ガイスターとそのプレイヤ

2.1 ガイスターのルール

ガイスターは，縦横 6×6 の盤面でゲームが行われ四隅に脱出口と呼ばれるものがある．一人 8 個の駒を持っていて，それぞれ青駒が 4 つ，赤駒が 4 つずつ持っている．上下左右のどちらか一方に駒を一つ動かすことができ，自分の脱出口を目指して駒を動かしていく．盤面の四隅にある脱出口のうち，相手の陣地側にあるものが自分の脱出口である．ゲーム開始の初期配置を図 1 に示す．図 1 において，囲まれている 2×4 の範囲で自分の駒をランダムに配置する．プレイヤは相手の駒色が分からない状態でゲームを行い，相手の駒を取ることで相手の駒色を知ることができる．

またプレイヤは，以下の条件一つを満たすことで勝利できる．図 2 はそれぞれ勝利条件を満たした場合の盤面を表している．

(1) 相手の青駒をすべて取る
(2) 相手に赤駒をすべて取らせる
(3) 相手の陣地にある自分の青駒を脱出口から脱出させる

このゲームにおいて，勝利条件の駒をすべて取ることや脱出口に青駒を脱出させることで勝敗が必ず決まるため，引き分けは基本的に存在しない．ガイスター AI 大会では 200 手目 (先手 100 手，後手 100 手) で勝敗が決まっていなければ引き分けとしている．本研究では，256 手目で勝敗が決まっていなければ引き分けとした．また，プレイヤの相互対戦で勝ち点を，勝ちを 1，引き分けを 0.5，負けを 0 とする．

2.2 データ生成に用いたプレイヤ

本研究では，性質の異なる 4 つのプレイヤを作成しそれらを相互対戦させることで，駒色推定ネットワークの学習に用いる訓練データと評価データを生成した．

作成した 4 つのプレイヤを以下に示す．

MC　原始モンテカルロプレイヤ．各プレイアウトにおいて，その開始時に相手駒の駒色をランダムに決定する．プレイアウトにおける手の選択は，脱出できる青駒がある場合にはその駒を脱出させる以外は，すべての合法手からランダムに決定する．256 手で引き分けとしてプレイアウトを打ち切る．プレイアウト数は，合計で 20000 とする．

PR　GAT2020 ガイスター AI 大会の優勝プログラム Naotti2020-3 を改変したプレイヤから学習した方策に従い手を選択するプレイヤ．詳細は付録 A.1 に示す．

PM　プレイアウトにおける手の選択に上記 PR プレイヤを用いるモンテカルロプレイヤ．各プレイアウトに時間を要するため，プレイアウト数は 50 とし，プレイアウト対象の手を UCB1 により選ぶ．プレイアウト数が少ないため，PR プレイヤの手の選択確率により勝率を初期化する．

RA　ランダムプレイヤ．脱出できる青駒がある場合にその駒を脱出させる以外は，すべての合法手からランダムに選択する．

これらのプレイヤの相互対戦の結果を**表 1** に示す[*1]．最も強いプレイヤは MC で，自己対戦を除く勝率 89.2% であり，次に強いプレイヤは PR で，自己対戦を除く勝率 58.2% であった．モンテカルロ法とポリシーネットワークを組み合わせた PM は自己対戦を除く勝率が 49.0% と強くなかった[*2]．それでも，これら 3 つのプレイヤはいずれも，ランダムプレイヤ RA に対しては 90% 以上の確率で勝っている．

これらのプレイヤを相互対戦させて訓練データと評価データを生成した．具体的には，4 つのプレイヤの先手・後手の各組み合わせについて，2560 ゲーム対戦させた．そ

[*1] この勝ち点は勝率とほぼ等価である．PR の自己対戦（3.4%），PR と RA の対戦（PR 先手 1.6%，RA 先手 1.5%），RA の自己対戦（5.9%）を除く組み合わせでは，引き分けとなったゲームは 0.1%以下であった．
[*2] モンテカルロ法において相手の駒色をランダムに決定したことで，実際は赤駒であるのに，青駒としてシミュレーションした結果から全ての赤駒と取って負けてしまうことが多く見られた．

表 1　先手の平均勝ち点（勝ち 1，引き分け 0.5，負け 0）．

		後手				自己対戦を除く平均
		MC	PR	PM	RA	
先手	MC	0.499	0.846	0.835	0.997	0.893
	PR	0.168	0.504	0.649	0.933	0.583
	PM	0.185	0.343	0.513	0.948	0.492
	RA	0.003	0.068	0.054	0.505	0.042
自己対戦を除く平均		0.109 (0.891)	0.419 (0.581)	0.513 (0.487)	0.959 (0.041)	

表 2　ニューラルネットワークの構成と学習方法

モデル	説明	パラメータ数
base	入力：盤面（6×6×4）．conv2d（2×2，フィルタ数 64）/ReLU 3 層，平坦化（3×3×64 ⇒ 576），全結合層（256 出力）/ReLU，全結合層（2 出力）/Softmax.	182 210
count	入力：盤面（6×6×4）＋残り駒数（16）．conv2d（2×2，フィルタ数 64）/ReLU 3 層，平坦化·結合（3×3×64+16 ⇒ 592），全結合層（248 出力）/ReLU，全結合層（2 出力）/Softmax.	181 546
countCH	入力：盤面（6×6×5）＋残り駒数（16）．conv2d（2×2，フィルタ数 64）/ReLU 3 層，平坦化·結合（3×3×64+16 ⇒ 592），全結合層（248 出力）/ReLU，全結合層（2 出力）/Softmax.	181 802

れより，各プレイヤについて 20480 ゲーム分対戦したことになる．訓練データは，そのうち 12800 ゲームからランダムに抽出した 40 万局面とした．評価データは，訓練データに用いていない 2240 ゲームからランダムに抽出した 2 万局面とした．

統計的な安定性のため，訓練データと評価データを以上の方法で 9 通り抽出した．以降の実験・評価にあたっては，9 通りのデータに対して実験を行い，得られた値の中間値を最終的な値とする．

3. 単一局面の手に対する駒色推定

まず，三塩ら [6] が提案した駒色推定と同様に，ある局面において選択された手で動いた駒の駒色を推定する．

3.1 ニューラルネットワークの構成

表 2 に本研究で使用したニューラルネットワークの構成を示す．本節では，そのうち，base と count を用いる．

base は畳込み層 3 層，全結合層 2 層からなる畳み込みニューラルネットワークである．入力は，6 × 6 の配列 4 つからなり，それぞれ盤面上の相手駒の位置，自分の青駒の位置，自分の赤駒の位置，相手の動かした駒の移動元と移動先を示す．それぞれの畳み込み層では，2 × 2 のフィルタが適用される．

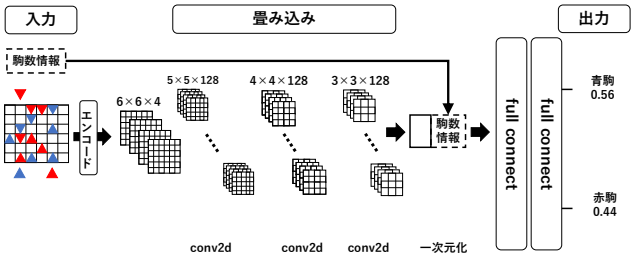

図 3　count のネットワーク構成

count は，base に対して残り駒数の情報を追加したネットワークである．入力は，base の 6 × 6 × 4 の配列に，自分／相手の青駒／赤駒の数がそれぞれ 1 ～ 4 のいずれであるかをビットで表した要素数 16 の 2 値配列を加えたものである．追加された 2 値配列は，盤面情報に対する 3 層の畳み込み・平坦化の後で結合される．base とパラメータ数を揃えるため，count では一つ目の全結合層の出力数を少し減らし，それ以外は同じとした．**図 3** に，count のネットワーク構成を図示する．

教師あり学習により base および count のネットワークを学習させた．訓練データにおける実際の駒色を正解とし，損失関数（Loss）は平均交差エントロピーとした．学習アルゴリズムは ADAM とし，そのハイパーパラメータには TensorFlow におけるデフォルト（$\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-7}$）を用いた．バッチサイズを 1000 とし，40 万局面の訓練データに対して最大 30 エポック学習させた．

3.2 結果と考察

表 3 に，4 つのプレイヤそれぞれについて，そのプレイヤから生成した訓練データと評価データを用いた場合の一致率と Loss の結果を示す．

訓練データ・評価データともにプレイヤ MC によるもののとき，駒色推定の精度が最も高く，base ネットワークで 89.9%，count ネットワークで 91.2% を達成した．既存手法であるモンテカルロシミュレーションによる駒色推定の一致率（64%程度 [6]，80% 程度 [8]）から考えると，畳み込みニューラルネットワークを用いることでより高い精度が得られていると言える．

base と count の比較では，入力に与える情報が count のほうが多いこともあり，count のほうが一致率が高くなった．ただし，プレイヤ PR のデータで学習・評価した場合には，その差は非常に小さい．これは，プレイヤ PR の方策の元にした Naotti2020-3 が残り駒数にあまり依存せずに手を選択していたためだと考える．

プレイヤ MC と同様にモンテカルロ法を用いるプレイヤ

表 3　同一プレイヤにより生成した評価データの場合の駒色推定結果

プレイヤ	base		count	
	一致率	Loss	一致率	Loss
MC	89.87%	0.1233	91.18%	0.1070
PR	83.02%	0.1802	83.22%	0.1741
PM	61.52%	0.3250	62.29%	0.3175
RA	49.97%	0.3567	55.94%	0.3380

表 4　異なるプレイヤから生成された訓練データと評価データを用いた場合の駒色推定結果

評価データ	訓練データ					
	MC		PR		PM	
	一致率	Loss	一致率	Loss	一致率	Loss
MC	91.18%	0.1070	77.27%	0.2844	82.46%	0.2247
PR	69.16%	0.3897	83.22%	0.1741	75.73%	0.2597
PM	59.02%	0.5199	59.18%	0.4751	62.29%	0.3175
平均	73.12%	0.3389	73.22%	0.3112	73.49%	0.2673

PM では，一致率が約 62% となった．プレイヤ MC と比べて一致率がかなり小さくなった理由の解析については今後の課題である．プレイヤ RA では，特に base において一致率が 50.0%となっているが，ランダムプレイヤでは本質的に駒色を推定することができないので，この結果は妥当である．

本論文のこれ以降では，ニューラルネットワークの構成として count（とその拡張である countCH）に限定して報告する．

次に，訓練データを生成したプレイヤと異なるプレイヤによる評価データを用いた場合の結果を**表 4** に示す．ただし，この表においてはランダムプレイヤ RA を除いている．評価データのプレイヤによる違いについて見ると，訓練データのプレイヤにほぼ依らず，プレイヤ MC が最も推定しやすく，続いてプレイヤ PR，プレイヤ PM の順であった[*3]．一方，訓練データのプレイヤの違いについて見ると，3 つの評価データに対する結果の平均一致率は 73% 程度で同程度となった．しかし，Loss の値を見ると，プレイヤ PM による訓練データから学習した結果が最も Loss が小さくなった．これより，一致率の観点ではプレイヤ MC が最も高いものの，複数のプレイヤに対する評価においてはプレイヤ PM による学習が最も優れることになる．

訓練データの違いによる学習されたネットワークの動作の違いについて解析するため，ネットワークが評価データにおける正解に対してどのような値を返したかをグラフにした．4 つのプレイヤに対し，同一プレイヤによる訓練データと評価データに対する結果を**図 4** に示す．また，プレイヤ MC，PR，PM の 3 つの評価データ全体に対して，4 つのプレイヤによる訓練データで学習したネットワークの結果を**図 5** に示す．プレイヤ MC による学習で得られたネットワークでは，極端な（1 に近い，または，0 に近い）値が返ることがより多い．そのため，3 つの評価データ全体に対して一致率がほぼ同じであるのに Loss が大きくなっていることが分かる．

[*3] 例外は，訓練データがプレイヤ PR であった場合のみである．

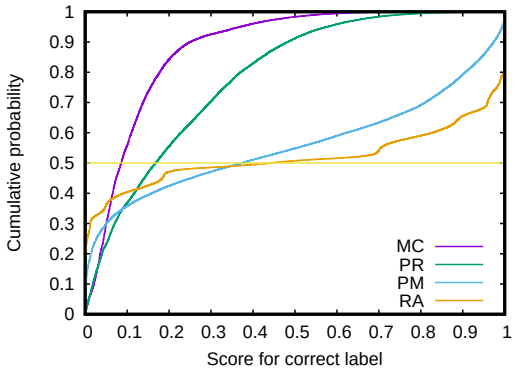

図 4　同一プレイヤによる評価データにおける，正解に対するニューラルネットワークの出力の累積分布

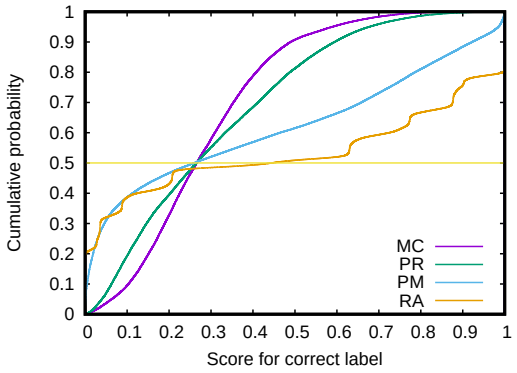

図 5　MC と PR と PM による評価データ全体における，正解に対するニューラルネットワークの出力の累積分布

以上の結果を受けて，本論文のこれ以降では，学習データ・評価データとしてプレイヤ MC とプレイヤ PM の 2 つに限定して述べることとする．

4. 推定対象の駒以外の推定駒色を利用した駒色推定

前節では，ある局面において，相手駒の駒色については何も情報がないという状況で駒色を推定する問題を扱った．ガイスターでは，相手駒の駒色は見えないものの，どの駒をどのように移動させたかの履歴は完全に把握することができる．したがって，1 ゲームにおけるそれまでの履歴から，相手駒の駒色をある程度推定することができる．そこで本節では，相手駒の駒色について，ある程度の情報があるという状況で駒色を推定する問題を扱う．

4.1 実験設定

本節で用いるネットワークは，表 2 の countCH である．学習時の入力のエンコードにおいて，相手駒の色情報を盤面情報の第 5 層として追加した．青駒と想定する駒の場所には区間 $[+0.25, +1.00]$ の値をランダムに与え，赤駒と想定する駒の場所には区間 $[-1.00, -0.25]$ の値をランダムに与えた．countCH ネットワークは，最初の畳み込み層の入力が $6 \times 6 \times 5$ の配列となること以外は，count ネットワークと同じである．

ネットワークの学習に用いる訓練データには，プレイヤ PM によるものを用いた．学習にあたっては，以下の 2 つのパラメータを指定できるようにした．

- p : 駒色推定の対象でない各相手駒について，色情報を与える確率．$p = 0$ のとき，単一局面における駒色推定に相当する．
- e : 駒色情報を与える際に，間違った駒色とする確率．$e = 0$ のとき，与えられる全ての色情報は正しい．

評価についても同様に，以下の 2 つのパラメータを指定できるようにした．

- P : 駒色推定の対象でない各相手駒について，駒色情報を与える確率．
- E : 駒色情報を与える際に，間違った駒色とする確率．

本研究では，駒色を与える確率 p を 0.1 から 1.0 まで 0.1 刻みで，また，間違った駒色とする確率 e を 0 から 0.3 まで 0.1 刻みで指定し，それぞれ学習を実行した．以下の解析においては，学習におけるパラメータとして，$p = 0.2, 0.4, 0.6$ および $e = 0, 0.3$ の場合についてのみ示す．評価におけるパラメータについては，間違った駒色の確率 E を，評価データ MC に対しては 0 または 0.3 に，評価データ PM に対しては 0 または 0.3 とした．駒色情報を与える確率 P については，0 から 1.0 まで 0.1 刻みで評価した．

4.2 実験結果と考察

プレイヤ MC による評価データに対して，$E = 0$ としたときの駒色情報の確率と一致率の関係を**図 6** に，$E = 20$ としたときの関係を**図 7** に示す．また，プレイヤ PM による評価データに対して，$E = 0$ としたときの駒色情報の確率と一致率の関係を**図 8** に，$E = 30$ としたときの関係を**図 9** に示す．

学習において正確な駒色情報を与えた場合（$e = 0$），評価における駒色情報が正確であればより高い精度が達成でき，駒色情報を与える確率が増えるほど精度が高くなっている．しかしながら，入力にそれまでの駒色推定結果を利用する場合，その駒色推定には間違いが入ることからこの仮定はあまり現実的ではない．また，図 6 のように異なるプレイヤに対しては，正確な駒色情報を用いて学習した結果，駒色情報のない場合（$P = 0$）において推定精度が下がってしまっている．

表 5 色推定を行うモンテカルロプレイヤの平均勝ち点

プレイヤ MC による訓練データの場合

プレイヤ		count		countCH	色推定
		単一	継続	継続	なし
MC	先手	0.575	0.656	0.603	0.499
	後手	0.628	0.625	0.653	0.501
PM	先手	0.289	0.581	0.569	0.185
	後手	0.303	0.614	0.547	0.165

プレイヤ PM による訓練データの場合

プレイヤ		count		countCH	色推定
		単一	累積 A	累積 B	なし
MC	先手	0.500	0.547	0.544	0.499
	後手	0.525	0.539	0.569	0.501
PM	先手	0.269	0.314	0.400	0.185
	後手	0.197	0.389	0.367	0.165

より現実的な設定として，評価において不正確な駒色情報を与えた場合について結果を考察する．学習において正確な駒色情報を与えた場合（$e = 0$）では，駒色情報を与える確率を増やすと駒色推定精度が下がってしまう結果となった．一方，不正確な駒色情報を用いて学習した場合（$e = 30$）では，PM による評価データにおいて精度が向上した．MC による評価データではベースラインの精度が高いことからそれを越える結果は得られなかったが，低下量は小さかった．

以上の結果より，続く実験では，駒色情報を与える確率を 40%，そのうち間違った駒色情報を与える確率を 30% と指定して学習した countCH を用いる．

5. 色推定を用いるプレイヤの対戦実験

作成したニューラルネットワークによる駒色推定を用いるモンテカルロプレイヤを作成し，その強さを評価した．モンテカルロプレイヤ（プレイヤ MC，PM）では，各プレイアウト前に相手駒の色をシャッフルする．本節で用いるプログラムでは，駒色推定により計算される各駒の青駒確率を保持し，相手駒の色をシャッフルする際にその確率に沿うように色を決定する．

5.1 色推定を用いるプレイヤの実装

使用する駒色推定ネットワークと各駒の青駒確率の更新方法について，3 通り実装した．

count 単一 駒色推定のネットワークに count を用い，相手の直前の手で動かした駒の青駒確率をネットワークの出力とする．それ以外の相手の駒の青駒確率は均等にする．すなわち，それ以前の相手の駒色推定結果を利用しない．例えば，相手赤駒が 2 個，青駒が 3 個であり，相手の直前の手の駒 a の青駒確率が 0.8 だとする．そのとき，相手駒 a の青駒確率 p_a は 0.8，残りの駒の青駒確率は $p_i = (3 - 0.8)/(3 + 2 - 1) = 0.55$

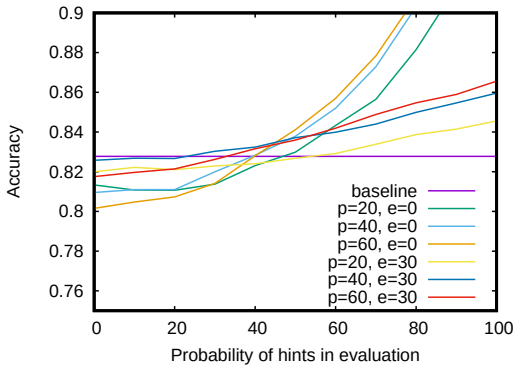

図 6　評価データ MC，$E=0$ の場合の駒色情報確率に対する一致率

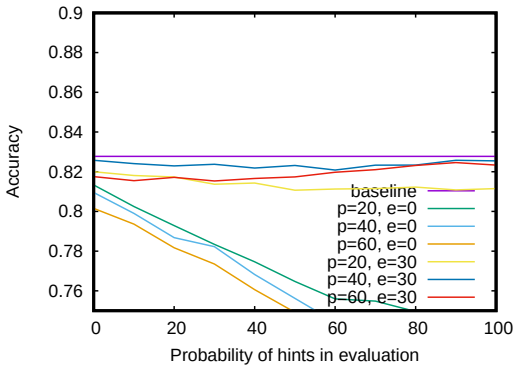

図 7　評価データ MC，$E=20$ の場合の駒色情報確率に対する一致率

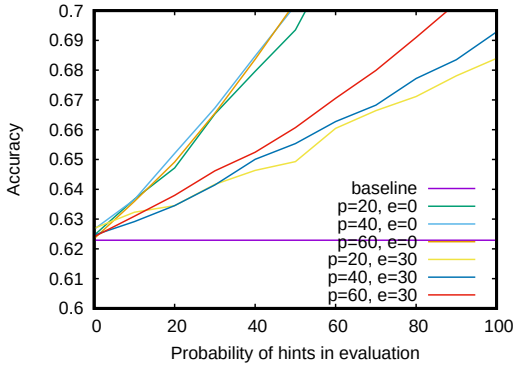

図 8　評価データ PM，$E=0$ の場合の駒色情報確率に対する一致率

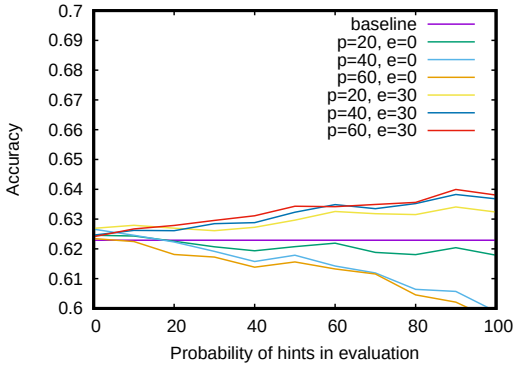

図 9　評価データ PM，$E=30$ の場合の駒色情報確率に対する一致率

となる．

count 継続　駒色推定のネットワークに count を用い，駒色推定結果を累積していく．ゲーム開始時には，相手駒の青駒確率は全て 0.5 であるとする．相手が手を選んだ際に，ネットワークから得られた値をもとに青駒確率を以下のように更新する．ただし，更新前の青駒確率を p_i，駒 a の駒色推定の結果が p'_a，$\delta = p'_a - p_a$ とする．[*4]

$$p'_i = \begin{cases} p_a & (\text{駒 } a) \\ p_i + \dfrac{\min(1-p_i, p_i)}{\sum_j \min(1-p_j, p_j)}\delta & (\text{駒 } a \text{ 以外}) \end{cases}$$

また，相手駒を取って色が判明した場合には，$p'_a = 0$ または $p'_a = 1$ として上記の更新を実行する．

countCH 継続　駒色推定のネットワークに countCH を用いること以外は count 継続と同じである．確率 0.5 に近いところでの揺らぎの影響を小さくするため，相手駒の駒色に関する countCH ネットワークの入力 v_i は，相手駒の青駒確率をもとに

$$v_i = \begin{cases} 2p_i - 1.0 & (p_i < 0.35) \\ 0 & (0.35 \le p_i \le 0.65) \\ 2p_i - 1.0 & (0.65 < p_i) \end{cases}$$

とした．

5.2　実験結果と考察

以下の条件で対戦実験を行った．対戦に用いるプレイヤは，プレイヤ MC とプレイヤ PM における相手駒のシャッフルを置き換えたものである．対戦相手のプレイヤは色推定のないモンテカルロプレイヤ（プレイヤ MC）とし，先手後手を入れ替えて計 720 戦（40 戦 × 2 × 9 通り）ずつ対戦させた．駒色推定ネットワークには，プレイヤ MC とプレイヤ PM から生成した訓練データを用いた．**表 5** に対戦結果として，色推定を行うモンテカルロプレイヤの平均勝ち点を示す[*5]．

訓練データと駒色推定方法のいずれの組み合わせにおいても，ベースラインである色推定なしのプレイヤよりも勝率が向上した．

単一局面の駒色推定のみを行う MC プレイヤでは，平均

[*4] $\sum_j \min(1-p_j, p_j) < \delta$ のとき，更新後の確率が負または 1 より大きくなる可能性があるため，そのときには δ を制限した．

[*5] 試合数が多くなかったため，この評価のみ中間値ではなく平均値とした．

勝率が 60.1% であった．これは，鴛淵ら [8] によるモンテカルロ法による駒色推定プレイヤの勝率 56% よりも良い結果である．ニューラルネットワークにより駒色推定の精度が上がったことがその理由であると考える．

また，駒色推定の結果を継続して用いるプレイヤは，単一局面の駒色推定に比べて大きな精度向上を果たしている．最も良い結果は，プレイヤ MC のシャッフルにおいて，プレイヤ MC による訓練データを用いた count 継続の方法をとったもので，その勝率は 64.0% であった．ランダムなシャッフルでは勝率の低かったプレイヤ PM においても，プレイヤ MC による訓練データを用いた count 継続の方法をとったものは，その勝率が 17.5% から 59.7% に大幅に向上した．プレイヤ MC による訓練データを用いた場合，count を用いたほうが countCH を用いたものより勝率が高かった．これは，プレイヤ MC による訓練データを用いた学習では一致率がとても高く，countCH の学習で意図的に間違いを含めたことが悪影響したためではないかと考える．

対戦相手と異なるプレイヤ PM により学習した色推定ネットワークを用いた場合でも勝率が向上していることは重要な結果である．特に，プレイヤ PM により学習した場合，他の相手駒の推定結果を利用する countCH を用いたプレイヤのほうが勝率が高い．countCH による色推定を追加することで，プレイヤ MC の勝率が 50% から 55.7% へ，プレイヤ PM の勝率が 17.5% から 38.3% へとそれぞれ向上している．

6. 関連研究

2017 年よりゲーム AI トーナメント（GAT）においてガイスター AI 大会 [2] が開催されており，ガイスターのプレイヤ作成を中心に多くの研究が行われている．ここでは，相手駒の駒色の推定の観点で関連研究を示す．

栃川ら [4] は，相手駒の駒色推定が有効であるかどうかを実験的に評価している．具体的には，ゲーム開始時に相手駒のいくつかの駒色が判明していると仮定し，モンテカルロプレイヤの勝率を調査した．その結果，4 つの相手駒の駒色が分かっていれば勝率 86%，6 つの相手駒の駒色が分かっていれば勝率 90% とのことであった．

相手駒の駒色の推定法には，モンテカルロ法によるものとヒューリスティックスによるものが提案されている．三塩ら [6] は，モンテカルロ法に基づく相手駒色推定アルゴリズム UPP を提案している．これは，推定対象の駒色を赤または青と仮定してモンテカルロ法を行い，推定対象の手の良さにより色を推定するものである．UPP アルゴリズムの効果について鴛淵ら [8] により詳細な評価実験が行われている．原始モンテカルロプレイヤに対して色推定の一致度が 80%であり，また色推定を行うプレイヤの勝率が 56%であったと報告している．

GAT で優勝したプレイヤでは，ヒューリスティックスによる簡易的な駒色推定が実装されていることが多い．GAT2018 で優勝したプログラム TSP-2018A [3] では，各相手駒の「青らしさ」をその振る舞いに関する 7 つの条件と駒の位置により更新している．そのヒューリスティックな駒色推定は，木村らによる深層強化学習を用いたガイスター AI [7] でも利用されている．また，本研究におけるプレイヤ PR のためのベースとして利用した GAT2020 優勝プログラム Naotti2020-3 [5] でも，比較的単純なルールで相手駒が赤であると推定できるかを評価している．

7. まとめ

本研究では，畳込みニューラルネットワーク（CNN）を用いた相手駒の色推定法を提案し，その有効性を実験的に評価した．具体的には，単一局面における手の駒色を推定するネットワークと，それまでの駒色推定結果を利用して推定するネットワークを構成し，性質の異なる 4 つのプレイヤを用いて学習・評価を行った．また，そのようにして作成した駒色推定ネットワークを利用して駒色推定を行うモンテカルロプレイヤを作成し，駒色推定をしないモンテカルロプレイヤと対戦させた．

単純モンテカルロ法のプレイヤは，駒色の推定が容易であり，同一プレイヤにより生成した訓練データを用いた場合に 91.2% の一致率が達成された．一方，（ランダムを除く）3 つのプレイヤで評価したときには，平均の一致率は訓練データに依らず 73% 程度であった．対戦相手と同じプレイヤ MC により生成した訓練データで学習した駒色推定を用いたとき，プレイヤ MC の勝率が 50.0 % から 64.0% へ，プレイヤ PM の勝率が 17.5% から 59.7% と大きく向上した．また，対戦相手と異なるプレイヤ PM により生成した訓練データで学習した駒色推定を用いた場合でも，プレイヤ MC の勝率が 50.0 % から 55.7% へ，プレイヤ PM の勝率が 17.5% から 38.3% と大きく向上した．したがって，本研究で提案した畳み込みニューラルネットワークによる駒色推定は有用であると考える．

今後の課題の一つは，GAT ガイスター AI 大会優勝プログラムを含む複数のプレイヤに対して，本研究で得たニューラルネットワークによる駒色推定が有効であるかを評価することである．また，この駒色推定の結果をもとに，相手に駒色を悟らせない，または，駒色を誤解させるような手の生成方法を考えることも今後の課題である．

謝辞 本稿の実験の多くは，高知工科大学の IACP クラスタによって実施されたものである．

参考文献

[1] Geister, http://www.luding.org/cgi-bin/GameData.py/ENgameid/19686 (2013).
[2] ガイスター AI 大会，http://www2.matsue-ct.ac.

jp/home/hashimoto/geister/GPW/2017/index.html (2017).
[3] 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 論文集，pp. 13–16 (2018).
[4] 栃川純平，竹内聖悟：ガイスターの初期盤面における相手駒推定の有効性，ゲームプログラミングワークショップ 2020 論文集，pp. 10–15 (2020).
[5] 川上直人：https://github.com/j1211/Naotti-2020 (2020).
[6] 三塩武徳，小谷善行：ゲームの不完全情報推定アルゴリズム UPP とそのガイスターへの応用，情報処理学会第 31 回ゲーム情報学研究会，Vol. 2014-GI-31, No. 4, pp. 1–6 (2014).
[7] 木村勇太，伊藤毅志：深層強化学習を用いたガイスター AI の構築，ゲームプログラミングワークショップ 2019 論文集，pp. 130–135 (2019).
[8] 鴛淵隆斗，佐藤直之：ガイスターゲームにおけるモンテカルロ法を利用した駒推定およびブラフ手の生成可能性の検証，情報処理学会第 43 回ゲーム情報学研究会，Vol. 2020-GI-43, No. 20, pp. 1–7 (2020).

付　　録

A.1 PR プレイヤの詳細

GAT2020 ガイスター AI 大会における優勝プログラム Naotti2020-3 [5] を改変した方策によりプレイするプレイヤを以下のとおり作成した．まず，Naotti2020-3 の negamax 探索における手の列挙順序をランダムにするよう改変した．そのように改変したプレイヤを相互対戦させることで訓練データを作成した．ポリシーネットワークは，本研究の畳み込みニューラルネットワーク count と類似の構造を持つ．違いは，畳み込み層のフィルタ数が 128 であること，一つ目の全結合層の出力数が 512 であること，最終的な出力が 136 要素（位置 36×4 方向）であることである．最終的に得られたポリシーネットワークにおいて，選択された手との一致率は 40.6%であった．