

# 畳み込みニューラルネットワークを用いた 囲碁における1局の棋譜からの棋力推定

荒木 伸夫<sup>1,2,a)</sup> 保木 邦仁<sup>1</sup> 村松 正和<sup>1</sup>

受付日 2016年2月19日, 採録日 2016年9月6日

**概要:** 囲碁のプロ棋士は、1局の棋譜を見ればプレイヤーの棋力が分かるといわれている。本稿では、畳み込みニューラルネットワーク (Convolutional Neural Network; CNN) を使用し、1局の囲碁の棋譜より、プレイヤーの棋力を推定する手法を提案する。プレイヤーのレート値を推定する実験と、プレイヤーを上級/中級/初級にクラス分けする実験を行った。提案手法を実装して囲碁クエストの13路盤棋譜データを用いて学習させて実験したところ、レート値を推定する手法としては従来手法より平均自乗誤差が小さくなった。また、クラス分類する実験においては、1度CNNを用いてレート値を推定してからその値に応じてクラス分けを行う手法と、最初からクラス分類をCNNに学習させる手法の2種類を提案し、それぞれ長所と短所があることを確かめた。

キーワード: 囲碁, Convolutional Neural Network, 棋力推定, 棋力分類, レート値

## Estimating Player's Strength by CNN from One Game Record of Go

NOBUO ARAKI<sup>1,2,a)</sup> KUNIHITO HOKI<sup>1</sup> MASAKAZU MURAMATSU<sup>1</sup>

Received: February 19, 2016, Accepted: September 6, 2016

**Abstract:** It is said that any professional player can estimate a player's strength accurately by looking at just one game record. We propose to use Convolutional Neural Network (CNN) to estimate a Go player's strength from only one game record. We perform two experiments: (i) to estimate a player's rating, and (ii) to classify a player into three classes in strength. We use game records provided by GoQuest to train CNN. For estimating ratings, we compare our method with an existing method to find that our method gives a smaller average mean squared error than that of the existing method. For the classification, we compare two methods: (i) the method that classify a player according to the rating predicted by the CNN, and (ii) the method that trains CNN directly to classify a player based on just one game record. We observed that the two methods have different strong points and weak points.

**Keywords:** Go, Convolutional Neural Network, strength estimation, rate classification, rating

### 1. はじめに

インターネットの普及した現代は、顔をつき合わせなければ囲碁を打てなかった時代と異なり、人はインターネットを介していつでも、どこでも、誰とでも囲碁を打てるようになった。このようなインターネットを介して囲碁を打

てるシステムはインターネット碁会所といわれる。インターネット碁会所のおかげで、特に強い人同士の対局が増加し、アマチュア高段の棋力は数十年前に比べて上がっているといわれている。

一方、囲碁は新たに始めるにはハードルの高いゲームとして知られている。街ではいくつも初心者教室を開催したり、級位者大会を開催したりしているが、なかなか競技者人口は増えていない [1]。初心者に長くゲームを楽しんでもらい、上達を促すのは日本棋院含め多くのプロ組織の設立目的の1つであるし、現在も喫緊の課題である。

それではインターネット碁会所が初心者にとって学習の

<sup>1</sup> 電気通信大学

The University of Electro Communications, Chofu, Tokyo 182-8585, Japan

<sup>2</sup> 日本学術振興会特別研究員 (DC2)

Research Fellow of Japan Society for the Promotion of Science (DC2), Chiyoda, Tokyo 102-0083, Japan

a) a1341001@edu.cc.uec.ac.jp

場となるかという点、現実にはそうはなっていない。たとえば今回、囲碁クエストというインターネット碁会所における棋譜を用いるが、特に棋力の指定はしなかったにもかかわらず、初心者の棋譜が非常に少ない。

現在のインターネット碁会所では、多くの場合、新規参入者は数局～数十局打たないと適正な段位やレートが付かない。本来の実力より過小あるいは過大な段位やレートでの対局が発生することは、インターネット碁会所の利用者にとって精神的な負担となりうる。実際、初期に負け続けてインターネット碁会所を使わなくなってしまう人は特に初心者、子供の中に少なからずいる。初心者にとって囲碁を1局打ち切るのはかなりのエネルギーを必要とすることもこの傾向に拍車をかけている。たとえば5局打てば適正な棋力が推定されると分かっているにもかかわらず打てない人が多いのが実情である。このことは初心者がインターネット碁会所を使ううえでの障壁の1つになっている。

一方、人間は1局の棋譜から棋力を精度良く推定することができるといわれている。特に囲碁のエキスパート（プロ棋士）は棋譜を1局見ればプレイヤーの棋力をかなりの精度で推定できると信じられている。このことに関する文献は特に見当たらないが、日本棋院や関西棋院などの多くのプロ組織において、プロ棋士と1局打っただけで院生の可否を判断したり段級位の免状を出したりすることが行われていることから、これは事実に近いと推測される。

人間が1局の情報から棋力を推定できるならば、計算機でもやらせたいと考えることは自然である。もし計算機が1局だけから棋力を推定できるようになれば、インターネット碁会所でも、あるいは街の碁会所でも、新規参入者に1局打たせて適切な棋力を判定し、以後の対局を適切なハンデキャップで行うことができる。

本研究では、この「1局の棋譜データから棋力を推定する」ということを目標とする。この目標に対し、「畳み込みニューラルネットワーク（Convolutional Neural Network, 以下 CNN）」を用いる手法を提案し、実験を行い、評価する。

CNN は元来、文字認識や顔認識など画像認識に使われてきた技術である [2], [3]。階層の深い CNN (Deep Convolutional Neural Network, 以下 DCNN) が 2012 年に画像認識のコンペティション ILSVRC で 2 位に精度差 10% 以上の大差をつけて優勝する (9 層の DCNN を使用) [4] という出来事もあり、近年注目を浴びている。

2014 年頃から、DCNN は囲碁 AI の世界でも適用され始めた。たとえば、Clark ら [5] は、対称性を考慮した DCNN を使うことで、囲碁に関する知識に基づいた特徴抽出をほとんど行わなくても 41–44% の精度の着手予測 (Move Prediction) ができることを示した。また、Maddison ら [6] は、囲碁に関する知識に基づいた特徴抽出も組み合わせた DCNN を用いることで、約 55% の精度の Move Prediction

ができることを示した。この先行研究では、着手予測を行う DCNN にプレイヤーの棋力情報も入力する。この試みは、着手から棋力予想を試みる本研究とは逆の試みと見なすことができる。さらに、Tian ら [7] は、DCNN の中で先読みを行うことで、約 57% の精度の Move Prediction ができることを示した。そして、Silver ら [8] は、DCNN と自己対局による強化学習を組み合わせ、さらに DCNN に学習させた囲碁の状態価値関数とモンテカルロ木探索とを組み合わせることで、人間のプロ棋士をハンデキャップなしで倒すことに初めて成功した。これらの研究は、DCNN が囲碁の局面を認識するツールとして有効であることを示唆している。

囲碁に限らず、棋譜から自動的にプレイヤーの段位やレート値を推測する試みは今まで様々に行われてきた。チェスや将棋では、着手の優劣をグランドマスターレベルの思考プログラムを用いて解析し、プレイヤーの棋力を推定する方法が知られている [9], [10]。チェスにおいて最近では棋力を予測する精度を競うコンペティションも開催され、トップグループは1局の棋譜からプレイヤーの Elo レーティングを 160 ほどの平均自乗誤差で見積もる [11]。これらの研究は、コンピュータが人間のトップレベルと同等以上の棋力があることを利用している。Silver ら [8] に基づくアルファ碁がトッププロに勝利するという成功を収めた現在、同様な方法の囲碁への適用は興味深い。しかし現時点では、アルファ碁自身たいへん大きな計算資源を必要とするため、すぐにその方向の研究を行うことは困難と思われる。

Ghoneim ら [12] は囲碁において、GNU Go を用いてプレイヤーのレベルを “casual”, “intermediate”, “advanced” に分類したが、精度が明記されておらず、どの程度成功したのか論文からは判断できない。

Moudrik ら [13] は、あらかじめ手作業で設定した特徴に基づき棋譜から特徴抽出を行い、得られた特徴ベクトルを用いてニューラルネットワークにプレイヤーの棋力を学習させる方法を提案した。論文 [13] では、インターネット碁会所 KGS の 20k-6d のプレイヤーの棋力を、各プレイヤーあたり数十局の棋譜を用いて標準偏差 2.6–2.7k(d) 程度で見積もることに成功している。彼らの手法を用いた Web 上で棋譜から棋力を推測するシステム [14] も公開されている。本稿では文献 [13] の手法を従来手法として実装し、提案手法と比較検討する。

本研究では、インターネット碁会所「囲碁クエスト」[15] の 13 路の棋譜 [16] を用いる。

以下では、まず CNN を説明し、囲碁クエストとその棋譜を説明したのち、レート値を推定する提案手法と比較対象として実装した従来手法について説明する。次にレート値の推定に関する実験とその結果を報告し、従来手法との比較を述べる。それから、クラス分類問題に関して、レート値を推定してからクラス分類する手法とクラス分類自体

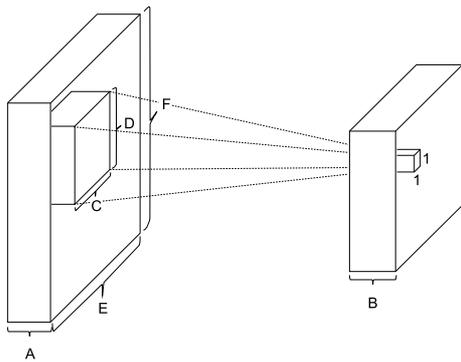


図 1 CNN のイメージ  
Fig. 1 Illustration of CNN.

を CNN によって学習する手法を比較する．最後に結論と今後の展望について述べる．

## 2. 畳み込みニューラルネットワーク

CNN は、ニューラルネットワークの一種である．このネットワークは、全結合層のように層と層の間でノードすべてが結合されるのではなく、図 1 のように前の層の  $C \times D$  のノードと次の層の 1 つのノードを結合される．CNN の 2 層間で行われる計算は、前の層の  $A \times E \times F$  の直方体を  $A$  個のチャンネルからなる  $E \times F$  画素と見なせば、ぼかしやエッジ処理などに代表される画像の畳み込み処理と基本的に同じである．中間層のノードは、直前の層の出力の中に存在する特徴的なパターンを検出する．入力に近い層ではエッジなどの局所的な特徴が抽出され、後段にゆくほど、前段の特徴が複数組み合わせられた、より大域的な特徴が抽出されるようになるといわれている [17], [18]．

このような CNN の階層を深くした DCNN を扱えるライブラリとして近年 Caffe [19] が広く使用されている．Caffe は基本的に Python から使うことを想定しているが、学習後のモデルを C++ から使うこともできる．Caffe を使用した囲碁 AI として oakfoam [20] などがある．

今回の我々の実験でも Caffe (git ハッシュ値 7953918) を用いる．Caffe に必要な入力は、ネットワークの構造を記述したファイル、メタパラメータを記述したファイル、およびデータファイルである．また GPU 使用もフラグを 1 つ切り替えるだけでできる．CNN は階層が深くなってくると計算量が莫大になるため、GPU を使用することは計算時間の節約に大いに役立つ．今回は GPU の Tesla K40 [21] を使用した．

## 3. 囲碁クエストの棋譜

囲碁クエスト [15] とは柵瀬寧氏が開発しているインターネット碁会所である．パソコンから参加することもできるが、スマートフォンやタブレットのアプリからの参加者が多い．9 路と 13 路で対局することができ、自動対局マッチング、チャットなし、中国ルール、AI による自動結果判

表 1 囲碁クエストの棋譜の統計情報

Table 1 Statistics of game records.

レート値	黒番のレートがこの範囲の棋譜 訓練/テスト	白番のレートがこの範囲の棋譜 訓練/テスト	平均手数 黒/白
0 – 1500	9330/1246	9555/1051	100/99.8
1501 – 2000	16939/2188	16926/2271	96.8/96.1
2001 – 2800	10713/671	10506/778	92.6/94.1

定、短い持ち時間、コンピュータ AI の参加という特徴がある．分かりやすいシステムで、スピーディーな対局が可能なることから、初心者にも魅力あるインターネット碁会所となりうる．また、初心者だけでなくトップアマやプロプレイヤーも参加しており、棋力の層は非常に幅広い．

最初に参加したときはレート 1000 であり、その後 1 局ごとにレートが上下していく．レートのほかに「レート  $X$  以上で  $Y$  連勝したら  $Z$  段」のように勲章的な役割を持つ段位もつく．

今回は柵瀬寧氏の協力を得て、囲碁クエストの 13 路の棋譜 [16] を使用して実験をした．対局時点でのプレイヤーの棋力に応じたレート値が各棋譜に残っているため、今回の実験で使用するには適している．棋譜に記録されている着手以外の情報は、手順、黒番のプレイヤー名とレート値、白番のプレイヤー名とレート値、結果である．

今回の学習では、黒番プレイヤーの棋力推定を行うシステムと白番プレイヤーの棋力推定を行うシステムを別々に用意した．囲碁の棋譜から盤面を再現して入力データにする際には、小林祐樹氏が開発した囲碁ソフト Ray [22] と自作の補助スクリプトを用いた．

黒番プレイヤー推定用と白番プレイヤー推定用に別々に訓練データとテストデータを用意し、これをすべての実験で用いた．たとえば黒番用のデータは以下のように作成した．まず、棋譜 [16] の中から、置き石のある対局と、どちらかのプレイヤーが AI の対局を除く．残った棋譜全体を棋譜数が 9 : 1 に近くなるように訓練データとテストデータに分けるのであるが、訓練データとテストデータに同じプレイヤーが黒番として現れないように注意した．白番でも、同じ棋譜全体から、訓練データとテストデータに同じプレイヤーが白番として現れないように注意して分割を行った．その結果、総棋譜データ量は 40,087 局、黒番用訓練データ量は 36,982 局、テストデータ量は 4,105 局、白番用訓練データ量は 36,987 局、テストデータ量は 4,100 局となった．それらの棋譜のレート値の分布は表 1 のとおりである．分布、平均手数ともに黒番と白番でデータに大きな偏りは見られない．

## 4. 提案手法

CNN の訓練で用いるデータは次のとおりである．

- 盤面情報：黒（白）番時の盤面を表す bit 列
- レート値：黒（白）番のプレイヤーのレート値

このうち盤面情報については、 $N$  手目までの盤面を用いることとする。 $N$  は実験により異なる。 $N$  手に満たない棋譜は最終手以降  $N$  手目までパスが続いたものとして扱う。黒番のプレイヤーのレート値を予測するシステムには黒番の盤面のみ、白番のプレイヤーのレート値を予測するシステムには白番の盤面のみを入力する。したがって黒番の盤面も白番の盤面も 1 棋譜あたり  $N/2$  の盤面を入力することになる。

各盤面については、

- (1) すでに置かれている黒石の位置が 1 でそれ以外が 0 の長さ  $13 \times 13$  の bit 列、
- (2) すでに置かれている白石の位置が 1 でそれ以外が 0 の長さ  $13 \times 13$  の bit 列、
- (3) 次の 1 手が 1 でそれ以外が 0 の長さ  $13 \times 13$  の bit 列を準備する。これらが  $N/2$  盤面分あるので、入力層のビット数は  $13 \times 13 \times (3N/2)$  となる。

今回用いた CNN の構造を図 2 に示す。全体は 5 層のニューラルネットワークになっており、 $3 \times 3$  の畳み込みを 3 回行ったのち、全結合させた内積値を出力する。中間層 1, 2, 3 のチャンネル数はそれぞれ 12, 8, 8 に固定した。これらは、訓練データのラベル数が約 37,000 であることから、パラメータ数がそれを大きく超えないように決めた。後述のとおり、 $N = 50$  のときパラメータ数が 9,961 個になる。畳み込み層の活性化関数は正規化線形関数 [18] を用いる。出力はレート値そのものを意味するものとし、確率的勾配法 [17], [18] により学習させた。

$N$  手までの情報を用いる場合、調整するパラメータの総数はバイアス 29 個を含めて  $162N + 1,861$  個であり、たとえば  $N = 50$  のときには 9,961 個となる。なお、囲碁を CNN で扱う場合に padding をしたり盤端を特別に扱ったりすることが有効であることもあるが、今回予備実験を行った範囲では特に何もしない場合に比べて結果が良くはならなかったため、採用しなかった。

1 度の反復で処理する訓練データ量 (Caffe [19] における

メタパラメータ batch\_size; 以下同様にカッコ内は Caffe のメタパラメータ名を表す) は訓練時 250 とした。学習係数 (base\_lr) は  $10^{-6}$  の固定値 (ただしバイアスに関しては 2 倍)、平滑化係数 (momentum) はデフォルトどおり 0 とした。重みの初期化 (weight\_filler) は Glorot ら [23] の方法 (xavier)、バイアスの初期化 (bias\_filler) は「0 埋めとした」。それ以外のメタパラメータは、基本的に Caffe のデフォルト値を用いた。

以下、混同の恐れのない場合には、提案手法を CNN と略記することにする。

### 5. MBN – 従来手法 [13] の実装

Mourdrik ら [13] は、あらかじめ手作業で設定した特徴に基づき、棋譜から特徴抽出を行い、得られた特徴ベクトルとレート値も用いて 3 層ニューラルネットワークに学習させた。比較のため、我々はこの論文 [13] に記述されている手法を実装した。ただし、13 路盤を対象とすること、文献 [13] のみでは不明な部分があること、使用するソフトが異なることなどから、いくつか異なる部分がある。この章ではそれを説明する。

以下簡単のため、文献 [13] をもとに我々が実装した手法を MBN と呼ぶことにする。

MBN と原論文 [13] との主な相違点を表 2、表 3 に掲げ

表 2 文献 [13] と MBN の比較

Table 2 Comparison of implemented features between [13] and MBN.

	文献 [13]	MBN
盤の大きさ	19 路	13 路
使用ソフト	pachi (2012)	Ray (2016)
NN の層数	3 層	4 層
隠れ層のノード数	20	20 (2 つとも)
学習アルゴリズム	RPROP	SGD
重み初期化	不明	Xavier
バイアス初期化	不明	定数 0
学習レート	不明	$10^{-6}$
Bagging	20	△

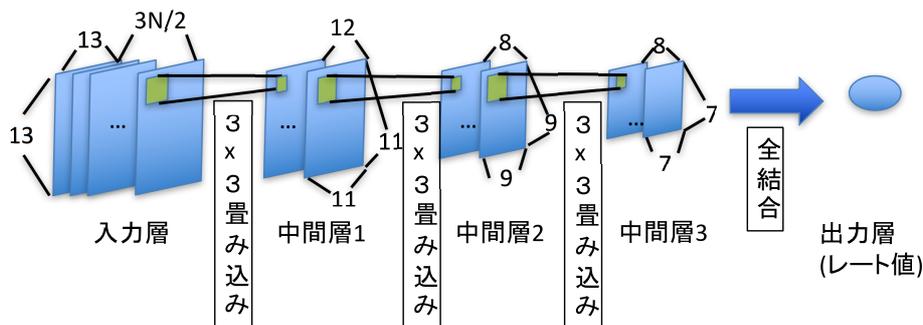


図 2 CNN の構造

Fig. 2 Structure of CNN.

表 3 文献 [13] と MBN で用いられる特徴

Table 3 Comparison of implemented features between [13] and MBN.

	文献 [13]	MBN
アタリ	○	○
アタリからの逃げ (手数, 取った石数)	○	○
直前の手との隣接関係 (手数, 盤端からの距離)	○	○
パターン	○ (形は不明)	○ (MD4)
$\omega$ -local 先手	○	△
勝敗	((勝/負), (投了/目数差))	勝/負

る。○は文献 [13] と同じように再現できているが、△は完全には再現できていないことを表す。開発環境が異なるため、ベースとなる囲碁ソフト、学習アルゴリズムなどで違いがある。また文献 [13] では 3 層ニューラルネットワークを用いているのに対し、MBN では 4 層としている。詳しくは実験の節で述べるが、4 層で MBN の学習が問題なく行われていることは確認している。

文献 [13] においては、20 個の Bagging を行っているが、我々には行っていない。実は我々も 20 個の Bagging を実装した版も作成していくつか実験を行ったが、ほとんど効果がなかったため、MBN では取り除いた。

文献 [13] で用いられている特徴と、MBN で用いられる特徴を表 3 に記す。MBN は 13 路盤を目的としているため、パラメータをいくつか変更している。たとえば手数と取った石数の直積を特徴としているところは、文献 [13] では手数を 1–60, 61–240, 241 以降の 3 段階にしているが、我々は 13 路盤という条件を考慮して 1–50, 51–100, 101 以降の 3 段階としている。また、手数と盤端からの距離の直積では、文献 [13] では手数を 1–10, 11–64, 65–240, 241 以降の 4 段階にしているが、我々はやはり 13 路盤であることを考慮して、手数は 10 手刻みとした。 $\omega$ -local 先手というのは文献 [13] で提案されている特徴であるが、この値はやはり盤の大きさに依存して決めるべきものである。原論文では  $\omega = 10$  のところ半分の  $\omega = 5$  としている。また、パターンについては、どのようなパターンを使用したのかは明示的には書かれていない。我々は Ray の MD4 と呼ばれるマンハッタン距離 4 以内のパターンを用いたもので、ここでも少し相違している可能性がある。

各層で全結合しているため、MBN のパラメータ数は 188701 と、CNN に比べかなり多くなっている。

## 6. レート値推定実験

まず、提案手法 CNN の学習結果を述べ、そのあと従来手法である MBN との比較を行う。

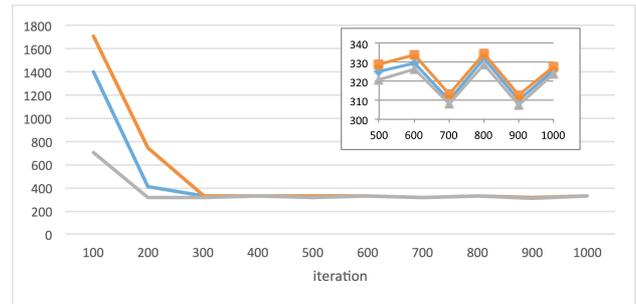


図 3 CNN の学習における適合誤差の減少の様子 (10 試行の最大・平均・最小、縦軸：適合誤差、横軸：反復回数)

Fig. 3 Decrease of error in CNN (Maximum, average, and minimum of 10 trials, left axis: fitting error, and bottom axis: iterations).

### 6.1 CNN によるレート値推定

最初に、実験結果の乱数依存性を調べるために、乱数の種を変えた 1000 反復の学習を 10 セット行った。図 3 は CNN の 1000 反復の学習を 10 セット行った際の、100 反復ごとにテストデータに対する平均自乗誤差の平方根（以下これを適合誤差と呼ぶ）をプロットしたものである。乱数の種を変えた 10 セットの学習について、その最大、最小、平均をプロットしている。この図では黒番で 50 手までの棋譜データを用いている。

10 セットの試行いずれも、初期に急激に適合誤差が減少し、その後 300 回を過ぎたあたりからは安定していることが見て取れる。図 3 の右上には、同じ図の 500 反復以上の部分を拡大したものを示す。試行による適合誤差の差はどの反復においても 10 以下であり、しかも学習が進むに従いだんだんと差が減少していることが分かる。白番でも同様の実験を行ったが、傾向は変わらなかった。そこで、以降では特に断らない限り、1 回の試行の結果を提示することとする。

CNN, DCNN では、長い時間学習させると良い結果が得られる場合がある。その可能性を探るため、黒番で  $N = 50$  手まで用いた場合に、30,000 回の反復を行った様子を図 4 に示す。横軸は対数目盛であることに注意する。また、図 4 の右上には、1000 反復以降の部分の拡大図を示す。

図 4 より、1000 反復を過ぎても適合誤差が下がっていることが分かる。これは  $N = 50$  の白番でもほとんど同様の結果が得られている。計算資源の都合から、本研究では 30,000 回反復し、その中で最も良いモデルを選択することを基本とすることにしたが、30,000 回よりもっと増やすとより良い結果が得られる可能性はある。

次に、使用する盤面の数を変えて、どのように学習結果が変化するかを見る。表 4 に、 $N = 2, 26, 50$  と変えて 30000 反復させた場合のベストな適合誤差とそのときの反復を記す。なお、(B) は黒番の予測を、(W) は白番の予測を表している。これを見ると、 $N = 50$  のときが最も適合誤差が

小さいことが分かる. また,  $N = 76$  および  $N = 100$  の場合も学習させてみたが, 30,000 回反復させても適合誤差はほぼ初期値のままであった. パラメータが多くなると学習が困難になり, より多くの反復を必要としたり, あるいは最適化のパラメータを調整したりしなければならないよう

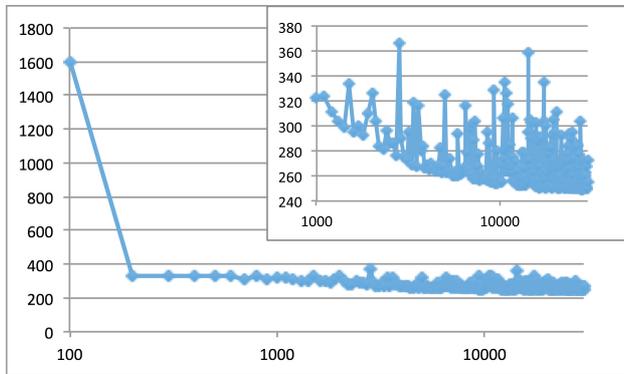


図 4 CNN の学習における適合誤差の現象の様子 (30000 反復, 縦軸: 適合誤差, 横軸: 反復回数)

Fig. 4 Decrease of fitting error in CNN (30000 iterations, left axis: fitting error, and bottom axis: iterations).

表 4 適合誤差と相関係数の比較

Table 4 Comparison in error and correlation.

手法	反復	適合誤差	相関係数
MBN(B)	10200	310.492	0.2984
MBN(W)	8600	285.741	0.2891
CNN, $N = 50$ (B)	27600	249.318	0.6091
CNN, $N = 50$ (W)	29700	243.272	0.5818
CNN, $N = 26$ (B)	29600	262.326	0.5533
CNN, $N = 26$ (W)	25200	259.227	0.4936
CNN, $N = 2$ (B)	24700	283.022	0.4345
CNN, $N = 2$ (W)	18400	270.319	0.3964
AVE(B)	-	329	-
AVE(W)	-	302	-

である. この点に関しては今後の課題とする.

また, ベースラインとして「いつも訓練データの平均値を返す」レート値予測器を構成した. その結果を表 4 の AVE の行に示す. AVE と CNN を比較すると,  $N = 2$  においてさえ, すなわち 1 手のみの情報からでも, CNN が AVE よりかなり良い適合誤差を得ていることが分かる.

1 手のみからこのような良い結果を得られるのは予想外であったので, 今回の棋譜の初手に関して, 棋力と位置の関係を調べた. するとまず, レートが 2000 以上の上級者は, 2000 未満の人たちより黒番の 1 手目を右上に打つ傾向が強いことが分かった. これは, 伝統的に囲碁においては, 黒番の初手を右上に打つのがマナーとされていることに起因すると思われる. また, 左下や 2 線に初手を打つ人はほとんど初心者であった. このようなことから, 1 手のみでも棋力に関してある程度の情報を持っていることが類推され, CNN はそれを抽出していると考えられる.

MBN の行, 相関係数の列に関しては次の節で説明する.

## 6.2 MBN によるレート値推定との比較

図 6 は MBN による黒番の学習を 30000 反復まで行った場合の目的関数値の減少の様子である. まず注意してほしいのは, 初期値がすでにかなり良いことである. たとえば図 4 では初期値が 1600 を超えているのに対し, 図 6 では 340 以下から始まっている. それでも, 1000 反復程度まで安定して適合誤差が減少しており, 学習が成功していることが分かる. しかし, 10000 反復を過ぎるとむしろ適合誤差は上昇しており, 過適合の疑いがある. 白番でも実験したが, 30000 反復では過適合の傾向は同じであった. なお, MBN のパラメータ数は 188701 である.

以下では MBN に関しては 30000 反復のうちで最も良い適合誤差を示した 10200 反復目のネットワークを用いて実験を行う.

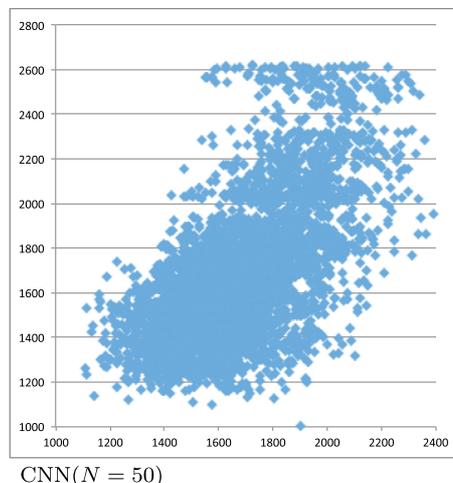
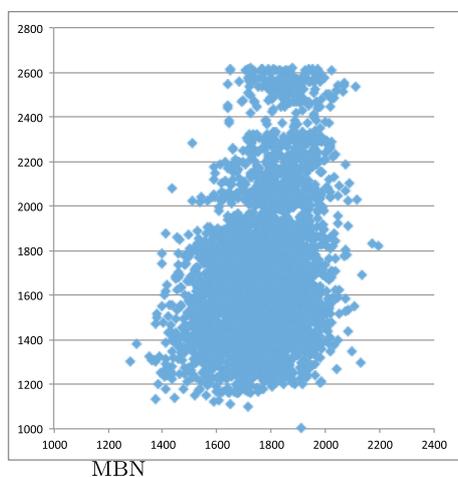


図 5 レート値の出力と正答の散布図 (縦軸: 正答, 横軸: 出力)

Fig. 5 Scatter plots of outputs and correct answers of rating (Left axis: correct answer and bottom axis: output).

表 5 クラス分類実験の結果 (黒番)

Table 5 Result of classification.

CNN-R				CNN-C			
	出力が 0	出力が 1	出力が 2		出力が 0	出力が 1	出力が 2
正解が 0	480(11.7%)	759(18.5%)	7(0.2%)	正解が 0	527(12.8%)	612(14.9%)	107(2.6%)
正解が 1	365(8.9%)	1718(41.9%)	105(10.6%)	正解が 1	396(9.6%)	1402(34.2%)	390(9.5%)
正解が 2	5(0.1%)	436(10.6%)	230(5.6%)	正解が 2	21(0.5%)	237(5.8%)	413(10.1%)
合計	850	2913	342	合計	929	2251	910

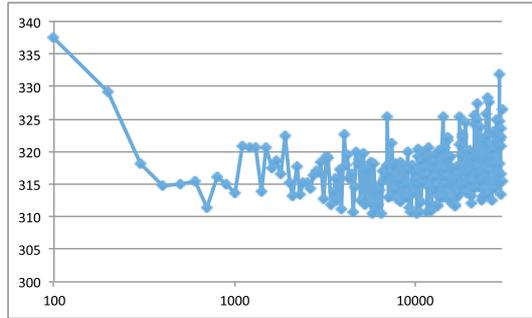


図 6 既存手法 [13] における適合誤差の減少の様子 (縦軸: 適合誤差, 横軸: 反復回数)

Fig. 6 Decrease of objective function in MBN (Left axis: fitting error and bottom axis: iterations).

MBN の黒番 10200 反復目の適合誤差を表 4 に示す。N = 2 の場合の CNN, すなわち 1 手のみから予測する場合でさえ、黒番においても白番においても適合誤差は MBN より優れていることが分かる。

MBN と CNN (N = 50, 27600 反復目) に関して、出力と正答をプロットした散布図を図 5 に示す。この散布図の相関係数は表 4 の一番右のカラムに載せる。数字からも、図 5 からも、MBN では出力と正答の間の相関が強くないことが分かる。

これらの実験から、提案手法は従来手法よりもレート値の推定精度がすぐれていることが確かめられた。

### 7. クラス分類実験

新規参入者の棋力を判定して、適切な組合せを見出すという目的から考えると、正確なレート値を推定できなくてもある程度のクラス分けができれば実用上は問題ない場合も多い。そこでこの章では、棋力のクラス分けに関する実験を行う。クラスはレート 1500 未満を初級者 (ラベル 0), レート 1500 以上 2000 未満を中級者 (ラベル 1), 2000 以上を上級者 (ラベル 2) とする。棋譜におけるこれらのクラスにおけるプレイヤーの分布は表 1 のとおりである。

ここでは 2 つの分類器を作成し、比較検討する。

(1) 前章におけるレート値推定ネットワークにおいて、ベストな適合誤差を示したのものにおいて、出力されたレート値からクラス分けするもの。これを以下 CNN-R と呼ぶ。

(2) 図 2 のネットワークの最終層を 3 つのクラスを表す 3 ノードに変更し学習させたもの。これを以下 CNN-C と呼ぶ。

CNN-C においては、最終層への活性化関数として Softmax 関数を用い、交差エントロピーを最小化した。その他のほとんどのメタパラメータは CNN-R と同じであるが、予備実験を行った結果、学習のステップサイズ (base.lr) のみは 0.01 の固定値 (ただしバイアスに関しては 2 倍) とした。

30,000 回の反復ののち得られたネットワークを用いて実験した結果を表 5 に掲げる。なお、分類の実験では 15000 反復以降は正解率 (後述) の変化が、57.3% を中央としておおよそ 1% 未満であり、図 4 の適合誤差のような値のバラつきは見られなかった。ここで 0 は初級クラスのラベル、1 は中級クラスのラベル、2 は上級クラスのラベルを表す。

正解率, すなわち

$$\frac{\text{正解と出力が一致した棋譜数}}{\text{全棋譜数}}$$

を計算すると CNN-R は 59.1%, CNN-C は 57.1% であり、わずかながら CNN-R の方が良い。

また、上級を初級と誤ったり、初級を上級と誤ったりすることは罪が重いと考えられるので、

$$\frac{\text{正解が 2 で出力が 0 の個数} + \text{正解が 0 で出力が 2 の個数}}{\text{出力が 0 の個数} + \text{出力が 2 の個数}}$$

を最悪分類割合と呼ぶことにして計算すると、CNN-R が 0.9%, CNN-C は 7.0% であり、CNN-R の方がかなり良い。図 5 で見たように、レート値を学習する CNN は出力レート値と真の値に強い相関がある。よって、真の値と非常に離れたレート値を出力し、最悪分類になってしまう確率は低いと考えられる。

一方、CNN-R の出力は明らかに 1 が多く、偏っている。実際、正解が 0 の人が正しく 0 と判定される割合は CNN-R が 38.5%, CNN-C が 42.3% であり、正解が 2 の人が正しく 2 と判定される割合は CNN-R が 34.3%, CNN-C が 61.5% とかなりの開きがある。もともとの訓練データにおいて中級者の棋譜は初級者および上級者の 2 倍程度あり (表 1 参照), CNN-R では特にそれに引きずられる傾向が顕著である。数の少ない初級者、上級者をきちんと見分けたいならば、最初からクラス分類を目的とした CNN を構

成して学習させた方が性能が良いことが確認された。

なお、ここでは黒番のみによる実験を報告したが、白番であっても個々の数値は異なるものの、傾向には差異が見られなかった。

## 8. 結論と今後の展望

CNN を用いて 1 局の棋譜からプレイヤーの棋力を推定する方法を提案し、既存研究 [13] と比較実験を行った。その結果、1 局の情報からレート値を推定することに限れば、提案手法は既存研究よりも良い精度を持つことが確かめられた。

また、レート値推定 CNN を用いて、出力の推定値を 3 分類する手法とそもそも 3 分類する CNN を構成する手法を比較した。最悪分類割合に関してはレート値推定 CNN を用いた方が良い結果を得たが、一方、この手法は「訓練データ量の小さいクラス (上級および初級)」の判定に関して問題があり、最初から 3 分類することを学習した CNN の方がこれらのデータに関しては正答率が高かった。これらの分類器は、目的によって使い分ける必要がある。

今後の研究課題として、以下のことが考えられる。

### (1) 大きな $N$ に対する学習方法の確立

現状では  $N = 76$  の場合ですでに学習が困難になっている。  $N$  が 50 手以下であるということは、小ヨセが無視されているということである。正解が分かる小ヨセで明らかなミスがあれば棋力推定の有力な手がかりとなるが、現状では小ヨセを対象とできていない。

これを克服するには、メタパラメータの慎重な調整、汎化性能のより高い特徴を利用するなどのモデルの変更、訓練データ量を増加させる様々なヒューリスティックの適用などを用いて学習が困難となっている理由をつきとめなければならない。大きな  $N$  での学習を可能にし、そのうえで  $N$  を 100 手以上まで上げた場合に有意に適合誤差が下がるのかどうかを調べることは重要な今後の課題である。

### (2) 複数局の棋譜情報を利用して性能を向上させる手法の検討

従来手法 [13] は数十局の情報を扱い、1 局のときよりも性能を向上させることが可能である。しかし我々の提案手法は、「数十局の棋譜から棋力を推定する」ことはおろか、数局の棋譜を用いることさえ現状では困難である。棋譜数が増えると、それにつれて入力データが膨大になり、必要とする訓練データ量も増大し、さらに学習が困難になると考えられる。

解決への方向性は上記の研究課題 1 と似ているかもしれないが、複数局の棋譜を利用した学習が可能になった場合に、1 局による学習よりもどの程度性能が向上するのかどうかは調査すべき課題である。

### (3) 相手プレイヤーの棋力分布の偏りの改善

囲碁クエストでは、近い棋力のプレイヤー同士を組み合わせる傾向があるので、訓練データの 2 プレイヤ間には、レートの相関があると思われる。現状のモデルでも、相手が過去に着手した石の位置が入力データとして与えられており、相手プレイヤーの情報は間接的に入力されていると考えられるので、このレートの相関が学習に影響を与えている可能性は考えられる。この影響は「インターネット碁会場の新規参入者に対する 1 局からのレート値の推定」という当初目的に照らすと適切ではない。

この影響を減らすためには、2 プレイヤのレートがちがらばるような訓練データがあればよい。しかし現状では、それでは十分な数の訓練データが得られない。これに対し、サンプル法の改善などを検討していく必要がある。

### (4) 棋力推定に対する他のアプローチの開発

文献 [6] においては、棋力を入力として与え、着手を予測することに成功している。これを応用し、棋力別に手を予測し、1 手ごとの誤差を計算して累積の誤差などを用いて棋力を推定するというアプローチを考えられる。このようなアプローチでどの程度本研究の結果を上回ることができるのか、あるいはできないのか、調査すべき研究課題である。

**謝辞** 囲碁クエストの棋譜を提供してくださった棚瀬寧氏に深く感謝いたします。メタ査読者と 2 名の査読者からは初稿に対して、論文の改善につながる多くの有用なご意見をいただきました。この場を借りて御礼申し上げます。この研究は、科学研究費補助金基盤研究 (B) 26280005、基盤研究 (C) 26330025、基盤研究 (C) 16K00503、および特別研究員奨励費 15J11695 による補助を受けている。

## 参考文献

- [1] 公益財団法人日本生産性本部 (編): レジャー白書 2015 国内旅行のゆくえと余暇, 生産性出版 (2015).
- [2] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P.: Gradient-Based Learning Applied to Document Recognition, *Proc. IEEE*, Vol.86, pp.2278–2324 (1998).
- [3] Lawrence, S., Giles, C.L., Tsoi, A.C. and Back, A.D.: Face Recognition: A Convolutional Neural-Network Approach, *IEEE Trans. Neural Networks*, Vol.8, No.1, pp.98–113 (1997).
- [4] Krizhevsky, A., Sutskever, I. and Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems 25*, Pereira, F., Burges, C., Bottou, L. and Weinberger, K. (Eds.), pp.1097–1105, Curran Associates, Inc. (2012).
- [5] Clark, C. and Storkey, A.: Teaching Deep Convolutional Neural Networks to Play Go, *Proc. ICML 2015* (2015).
- [6] Maddison, C.J., Huang, A., Sutskever, I. and Silver, D.: Move Evaluation in Go Using Deep Convolutional Neural Networks, *International Conference on Learning Representations* (2016).

- sentations (2015).
- [7] Tian, Y. and Zhu, Y.: Better Computer Go Player with Neural Network and Long-Term Prediction, *International Conference on Learning Representations* (2016).
- [8] 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).
- [9] Guid, M. and Bratko, I.: Using Heuristic-Search Based Engines for Estimating Human Skill at Chess, *ICGA Journal*, Vol.2, No.34, pp.71–81 (2001).
- [10] 山下 宏：将棋名人のレーティングと棋譜分析，ゲームプログラミングワークショップ2014 論文集 (2014).
- [11] Kaggle: FindingElo (2014-2015), available from <https://www.kaggle.com/c/finding-elo>.
- [12] Ghoneim, A.S., Essam, D.L. and Abbass, H.A.: Competency Awareness in Strategic Decision Making, *2011 IEEE 1st International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, pp.106–109 (2011).
- [13] Moudřík, J., Baudiš, P. and Neruda, R.: Evaluating Go Game Records for Prediction of Player Attributes, *2015 IEEE Conference on Computational Intelligence and Games (CIG)*, pp.162–168 (2015).
- [14] Moudřík, J. and Baudiš, P.: GoStyle – Determine playing style in the game of Go (2013), available from <http://www.gostyle.j2m.cz/>.
- [15] 棚瀬 寧：囲碁クエスト，入手先 (<http://wars.fm/go9?lang=ja>).
- [16] 棚瀬 寧：囲碁クエスト棋譜，by private communication.
- [17] 岡谷貴之：深層学習，講談社 (2015).
- [18] 麻生英樹，安田宗樹，前田新一，岡野原大輔，岡谷貴之，久保陽太郎，ボレガラダムシカ：深層学習，近代科学社 (2015).
- [19] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T.: Caffe: Convolutional Architecture for Fast Feature Embedding, arXiv preprint arXiv:1408.5093 (2014).
- [20] van Niekerk, F. and Schmicker, D.: oakfoam, available from <https://bitbucket.org/dsmic/oakfoam>.
- [21] NVIDIA: Tesla GPU でデータセンターを高速化，入手先 (<http://www.nvidia.co.jp/object/tesla-servers-jp.html>).
- [22] 小林祐樹：モンテカルロ木探索を用いた強い囲碁プログラムの設計と開発，修士論文，電気通信大学 (2016).
- [23] Glorot, X. and Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks, *International Conference Onartificial Intelligence and Statistics* (not specified, ed.), Amsterdam, Netherlands, pp.249–256 (2010).



荒木 伸夫 (正会員)

2006年東京大学理学部情報科学科卒業。2008年同大学大学院情報理工学系研究科修士課程修了。同年株式会社ジー・サーチ入社。2013年退職。同年電気通信大学大学院情報理工学研究科博士後期課程入学。2015年日本学術振興会特別研究員 (DC2)，現在に至る。囲碁 AI の研究に従事。情報処理学会ゲーム情報学研究会準会員。



保木 邦仁 (正会員)

1998年東北大学理学部卒業以降，化学領域での研究活動に従事。2000年東北大学大学院理学研究科博士前期課程修了。2003年同研究科博士後期課程修了。2003年から2006年にかけて，トロント大学博士研究員。2006年東北大学大学院理学研究科研究支援者。2007年から2009年にかけて，同研究科助手。2009年東北大学高等教育開発推進センターへ移動。2010年電気通信大学先端領域教育研究センター特任助教，2015年電気通信大学大学院情報理工学研究科准教授，現在に至る。情報処理学会ゲーム情報学研究会会員。



村松 正和 (正会員)

1994年総合研究大学院大学数物科学研究科博士後期課程修了，博士 (学術) 取得。1994年上智大学機械工学科助手。2000年電気通信大学電気通信学部情報工学科講師，2001年助教授，2006年教授，現在に至る。最適化とゲーム情報学に興味を持つ。Computer Go Forum 副会長。情報処理学会ゲーム情報学研究会会員。