

## ディープラーニングを用いた将棋の戦型の判別

### Classification of Play Styles of Shogi Game by Deep Learning

今村 奨<sup>†</sup> 神保 秀司<sup>†</sup>

Sho Imamura<sup>†</sup> Shuji Jimbo<sup>†</sup>

<sup>†</sup> 岡山大学大学院 自然科学研究科

#### 1 概要

コンピュータを用いた将棋を題材とする研究は幅広く行われている。本稿では、ディープラーニングを用いて将棋に存在する戦型（決まった駒の並びや動かし方から派生する盤面）を畳み込みニューラルネットワーク（CNN）を用いて学習させ、盤面が特定の戦型であると判別できるか実験を行った。8つに分けた戦型を学習させたモデルに対して実験を行ったところ、正しく判定できる能力には、居飛車の方が高く振り飛車は他の振り飛車の戦型と間違えやすい、序盤の方が正解率が高く局面が進行するにつれて正解率が下がる、といった差異が見られた。また、棋譜データベースサイトに登録されている棋譜で、コンピュータ同士で行われた対局で戦型が登録されていない棋譜に作成したNNを適用したところ、分類された戦型の特徴がその棋譜の中に見られた。

#### 2 はじめに

将棋はプロ制度の存在する2人で行うボードゲームの一種で、近年では機械学習によって人間よりも強いプログラムを作る研究が盛んである。それらの研究では盤面における指し手をプロが指すような手に近づけたり、コンピュータ同士の自己対局から勝利したときの手を学習し続けることでコンピュータ将棋が人間のプロを超える棋力を得ることに成功している。

しかし将棋の有利不利や指し手予想以外の要素、例えば攻めや受けの度合い、視覚的な盤面の理解などについての研究は正確な指し手を求める研究に比べてマイナーである。現在では、イベント等で対局が行われるたびに将棋の棋譜データベースサイトなどでその対局の対局者、棋譜、戦型といった情報が常に更新され続けている。しかし、コンピュータ将棋同士の対局<sup>1</sup>は戦型の情報が不明のまま棋譜が登録されている。

そこで、本稿ではディープラーニングを用いて戦型を学習させることで、未知の棋譜に対して盤面から戦型を判断できるのではないかと考え、作成したニューラルネットワークが獲得した戦型の分類能力を調べた。

#### 3 提案手法

ここでは、盤面の戦型を学習する手法について提案する。なお、ディープラーニングにはPython言語のKerasフレームワーク [1] を用いた。

#### 3.1 実験用データの準備

##### 3.1.1 棋譜データの取得

今回行ったのは盤面とそれがどの戦型のものかがラベルとして与えられたデータを学習して盤面の戦型を判断する教師あり学習である。戦型毎の棋譜として、将棋の棋譜データベースサイト<sup>2</sup>から62627局の棋譜をダウンロードした。棋譜はCSAと呼ばれる形式で書かれており、何手目にどの駒がどう動いたかの他に日付、対局者、対局が行われた場所、その対局におけるプレイヤーの戦型、といった情報が含まれている。なお、先手と後手のどちらがその戦型で対局したかの区別はついておらず、いずれかのプレイヤーがその戦型で指した棋譜が登録されている。これらから、対局が途中で中断されているものを除き、棋譜数が多い順に上位の戦型8つを取得した。さらにそれらを振り飛車と居飛車に分け、その両方についてそれぞれ4つに分け、表1のように計8つの戦型について合計45206局のデータを用意した。

表 1: 分類する戦型と棋譜数

	戦型	棋譜数
振り飛車	四間飛車	10512
	三間飛車	4817
	中飛車	5804
	向かい飛車	2324
小計		23457
居飛車	矢倉	11193
	角換わり	3108
	横歩取り	3414
	相掛かり	4034
小計		21749
合計		45206

##### 3.1.2 実験用データへの加工

まず、各駒に対して表2のように数値を割り振り、長さ81の1次元配列に初期盤面を設定する。なお、成金のような存在しない駒についても飛ばすことなく値

<sup>1</sup>floodgate(<http://wdoor.c.u-tokyo.ac.jp/shogi/floodgate.html>)

<sup>2</sup>将棋 DB2(<https://shogidb2.com>)

を割り振った。例えば、自分の成桂は 11, 相手の角は 22, 相手の成桂は 27 と表せる。次に, CSA 形式の棋譜データの操作に対応して配列の値を書き換えることで 1 手毎の盤面を表す 1 次元配列を取得し, 繰り返し記録することで投了までの対局の盤面データを得た。最後にこの 1 手毎の盤面を表す 1 次元配列を  $9 \times 9$  の 2 次元配列に整形し, 後の操作でチャンネル分けが出来るようにした。

表 2: 盤上の駒に割り振った値

駒	自	相手	成
歩	1	+16	+8
香	2		
桂	3		
銀	4		
金	5		
角	6		
飛	7		
王	8		

## 4 実験

将棋データベースサイトから取得した棋譜から, 戦型を判別する学習用のニューラルネットワークを keras を用いて作成し, 評価を行った。

### 4.1 使用したニューラルネットワーク

実験用のディープラーニングの CNN は表 3 のような構造を取った。入力には盤面を表す配列で与えられる, 高さ 9 ピクセル  $\times$  幅 9 ピクセル  $\times$  チャンネル数 28 の 3 次元である。各チャンネルは盤上の駒の種類を表し, 各チャンネルに対応する駒がある場所が 1, それ以外の場所が 0 と表される。

入力はまず  $3 \times 3$  のサイズを持つ 96 枚のフィルターを持つ畳み込み層と ReLU 層で処理され, その後 3 つの Residual Block[2] によって処理される。フィルター数は各 Residual Block 毎に 2 倍となるように設定し, 最終的に 768 となる。各 Residual Block の構造は表 4 のようになっている。また, 元画像が  $9 \times 9$  と非常に小さいため, 畳み込み層ではゼロパディングを行い, 画像サイズがこれ以上小さくならないようにした。最後に Global Average Pooling によって処理され, 全結合層で 8 つの戦型のいずれか 1 つのクラスへの分類を表すワンホット表現を出力する。損失関数には categorical crossentropy, 最適化関数には Adam, 全結合層の活性化関数には softmax をそれぞれ用いた。

また, 過学習防止のために, Keras の callback 関数から EarlyStopping を採用し, エポックを進めるうえで検証データの正解率 (accuracy) が前の学習から上がらない状態になったときに自動で学習を止めるようにした。この時の学習率 (lr) は 0.001 とした。

[3] ではさらに持ち駒を全ピクセルに対して枚数に応じた係数を掛けた値を持つチャンネルとしていたが, 持ち駒は戦型の判別には影響しないと考えられるため, 本実験では持ち駒のチャンネルは省略している。

学習データとして, 用意した全ての対局の局面からランダムに 1000000 局面を選択し, 同様に検証データとしてその 5 分の 1 の局面を学習データと重複しないように選択し, 用いた。

表 3: 用いた CNN の全体構造

入力 ( $3 \times 3 \times 28$ )
畳み込み層 (Conv2D, フィルター数 96)
活性化関数 (ReLU)
Residual Block1(フィルター数 192)
Residual Block2(フィルター数 384)
Residual Block3(フィルター数 768)
プーリング層 (Global Average Pooling)
全結合層 (出力空間次元数 8)

表 4: 用いた Residual Block の構造

畳み込み層 1(Conv2D)
活性化関数 1(ReLU)
畳み込み層 2(Conv2D)
活性化関数 2(ReLU)
入力との足し合わせ

### 4.2 ニューラルネットの学習結果

このモデルを用いた学習結果は図 1 のようになった。モデルの損失値と評価値は  $[0.548153, 0.853125]$  (小数点以下第 7 位以下切り捨て) であり, 検証用データに対して約 85% の正解率を得るモデルを得られた。EarlyStopping によってエポックは 14 で学習を停止した。

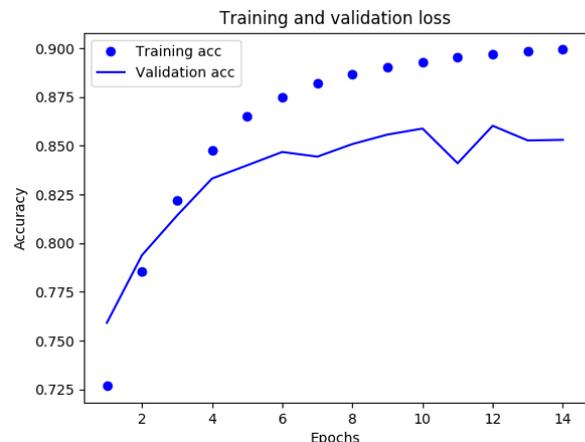


図 1: 作成したモデルの学習グラフ

### 4.3 作成したニューラルネットワークの分類性能

作成したニューラルネットワーク (NN) に学習に用いていない棋譜の戦型を判定させ、得られる結果を確認した。ネットワーク作成の際に学習にも検証にも使用していない棋譜を各戦型毎に 10 ずつ選び、それぞれの棋譜の 1 手目から投了までの各盤面でその盤面がどの割合で 8 つの戦型それぞれに判定されたかを調べた。そして、各棋譜の各盤面での 8 つの戦型それぞれの確率の総和を取り棋譜数で割ることで正解率の平均を取った。

その結果、作成した NN が各戦型の棋譜を与えた時にそれぞれどの戦型と判定したかの割合は表 5 のようになった。縦軸は検証に用いた棋譜の戦型を表し、横軸は各戦型の 10 の棋譜に対して NN が判別した戦型の割合 (小数点以下第 2 位を四捨五入) の平均である。

表 5: 各戦型の正解率 (百分率)

	四	三	中	向	矢	角	横	相
四	43.8	28.6	34.3	12.5	2.0	7.6	8.6	3.3
三	33.3	42.6	22.7	14.1	1.5	5.6	8.0	6.8
中	20.7	9.6	80.8	8.5	3.0	6.0	7.6	4.0
向	37.9	14.0	23.2	45.7	4.7	11.1	9.4	4.3
矢	12.9	3.9	7.6	6.7	68.4	16.2	8.5	9.0
角	4.8	0.5	6.1	6.4	6.1	89.4	7.8	12.4
横	3.8	4.5	8.2	7.0	2.1	19.5	75.0	13.3
相	4.3	6.2	7.1	6.6	6.2	22.1	20.7	63.1

この結果から、与えられた盤面に対して、対応する戦型が最も高く判断されていることが分かる。最も正解率が高い戦型は角換わりで 89.4%、最も正解率が低い戦型は三間飛車で 42.6% となった。また、振り飛車の戦型は他の振り飛車の戦型と誤って判定する確率が高いこと、居飛車の戦法は他の戦法と誤って判定する確率が低いことが分かる。

### 4.4 進行度毎の正解率

次に、盤面の進行度によって正解率がどのように変化するかを確認した。ただし、対局ごとに投了までの手数異なるため、進行度を統一する指標として、総手数を 3 等分して進行度を序盤中盤終盤と分け、それぞれ戦型での進行度毎の正解率の平均を求めた。対局を総手数で 3 等分し進行度を分ける手法は [4] を参考にした。各進行度におけるそれぞれの戦型に対する正解率は表 6 のようになった。縦軸は検証に用いた戦型を表し、横軸は各進行度に対して NN がその戦型と判別した割合 (小数点以下第 2 位を四捨五入) の平均である。

表 6: 各戦型の正解率

	序盤	中盤	終盤
四	43.8	31.5	24.2
三	59.3	43.7	20.2
中	87.9	71.2	50.3
向	54.5	27.0	18.7
矢	84.3	70.6	66.6
角	92.5	93.3	57.3
横	87.0	81.1	59.9
相	84.3	56.7	31.4

この結果から、殆どの戦型で序盤の正解率が最も高く、局面が進行するにつれて正解率が下がっていくことが分かる。4.3 で正解率が高かった居飛車の戦型は中盤になっても高い正解率を保っていたが、相掛かりのみ正解率の大きな減少が見られた。また、4.3 で正解率が低かった振り飛車の戦型は終盤になると 2 割程度の正解率しか残さなかったが、中飛車のみ居飛車と同等に高い正解率を維持していた。

### 4.5 未知の棋譜の分類への試み

最後に、コンピュータ将棋同士の対局から得られた、戦型が未分類の棋譜に対しても同様にモデルに判別させ、その対局がどの戦型で行われたかの結果に対する評価を試みた。棋譜データベースサイトから、floodgate 上でコンピュータ将棋同士で行われた、戦型が登録されておらず不明となっている棋譜から無作為に 4 つ選択し、作成した NN によってそれぞれどの戦型と判別されるかを確認した。実験は 4.3 と同様に、各棋譜の 1 手目から投了までの各盤面でそれがどの割合で 8 つの戦型それぞれに判定されたかの平均を取った。実験に用いた各棋譜で対局した将棋ソフトと、その棋譜の戦型として最も高く判定された戦型を表 7 に示す。

表 7: 実験に用いたコンピュータ同士の対局かつ戦型未分類の棋譜の判別結果

対局番号	対戦した将棋ソフト 上先手, 下後手	最も高く判別した戦型 とその割合
1	tenuki AobaZero_w768_n_p800	横歩取り (94.2)
2	Krist_483.473stb_100k elmo_WCSC27_479_100k	中飛車 (67.3)
3	gikou2_1 elmo_WCSC27_479_1000k	角換わり (89.7)
4	Krist_483.473stb_100k elmo_WCSC27_479_1000k	四間飛車 (66.2)

戦型不明の棋譜の判別はいずれも 8 つの戦型の中から特に高い評価で分類される戦型の一つを得た。実験で用いた棋譜について目視で確認したところ、図 2 から

図5のように、対局中にその戦型の特徴である駒組みが見られた。また、その戦型が確認できる手番は棋譜ごとに異なった。



図 2: 16 手目で横歩取りの戦型になっている

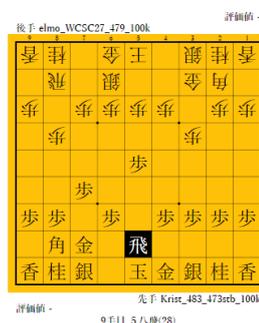


図 3: 9 手目で先手が飛車を5筋に振っている

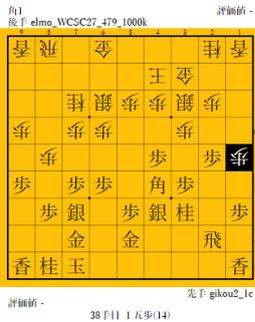


図 4: 開始直後に角交換し、38 手目でお互いに角換わりの戦型になっている

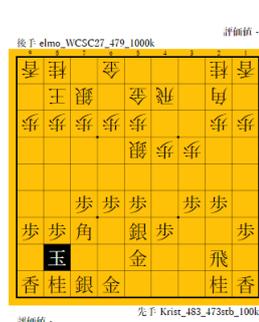


図 5: 23 手目で後手が四間飛車の戦型になっている

なお、今回の実験では8つの戦型以外の戦型に分類される棋譜を今回作成した NN で分類したときの結果については得られていない。そのような棋譜の場合恐らく8つの中から最もよく似ている戦型が他より高くなると考えられるが、学習の際に8つの戦型以外を「その他」として分類し、結果を9種類に分けるべきであった。

## 5 まとめ

本稿では、将棋の盤面に対してディープラーニングを用いて8つの戦型への分類を試みた。2次元の盤面データを盤上の駒の種類と等しい28のチャンネルに分けて3次元としたものを入力とし、畳み込みニューラルネットワークを用いて学習を行った。その結果、学習したモデルは正しい戦型を最も高く判別するように学習できているが、特に振り飛車において、似ている戦法も高く判別した。また、対局の進行度によって正解率がどのように変化するかを調べた実験では、局面が進行するほどその盤面の戦型を判別することが難しくなるということが分かった。戦型が未分類の棋譜に対して行った実験では、棋譜が高い割合で分類された戦型に対して盤面を確認したところ、確かにその戦型

に見られる特徴が確認できたが、8つの戦型以外へ分類される棋譜に対しての結果は確認できていない。

今後の課題として、ネットワークの構造や学習条件を見直し、学習モデルの正解率を上昇させること、判別可能な戦型を8つから増やすことが考えられる。特に現状では今回判別した8つ以外の戦型に対応する盤面を与えると正しい動作をしないと考えられる。コンピュータ同士の対局ではどの戦型にも属さない盤面になることも考えられたためこの点を今後判別可能な戦型を拡張する際に改善したい。また、先手と後手の両方で異なる戦型の特徴が見られた時にどのような分類になるかは未確認である。そのような場合、データベースではどのような基準で戦型を登録しているかを踏まえて戦型が未分類の棋譜への分類を試みたい。

上記に加えて、与えられた盤面に対してどの部分が戦型判別の根拠となっているかを可視化するために、盤面に対して顕著性抽出手法 [2] を適用することにも取り組みたい。

## 6 謝辞

本研究の一部は、株式会社アースライズカンパニーの支援により実施された。

## 参考文献

- [1] Francois Chollet. Python と Keras によるディープラーニング. 株式会社マイナビ出版, 2018.
- [2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, Vol. abs/1512.03385, , 2015.
- [3] 中屋敷太一, 金子知適. 将棋用ニューラルネットワークへの顕著性抽出手法の適用. ゲームプログラミングワークショップ 2018 論文集, 第 2018 巻, pp. 1-8, nov 2018.
- [4] 大森翔太郎, 金子知適. 将棋における棋風を学習するための棋譜分析の取り組み. ゲームプログラミングワークショップ 2015 論文集, 第 2015 巻, pp. 32-39, oct 2015.