

“Double-Fritz with boss” のボスをニューラルネットワークに置き換える研究

菅原 真^{1,a)} 保木 邦仁^{1,b)}

概要: Double-Fritz with boss はベースとなる人工知能 (AI) と人間のボスから構成されるチェスプレイヤーである。このプレイヤーの意思決定の概要は次のようである。まず、ベース AI が二つの候補手 (最善手と次善手) をボスに提示する。次に、ボスが一方の候補手を選択し着手する。本研究では、double-Fritz with boss のボスをニューラルネットワーク (NN) に置き換えることを目指す。この NN は、手番情報とベース AI の出力情報を基に対戦結果 (勝ち・引き分け・負け) を予測する。そして、ボスの候補手の選択はこの予測結果が良い方の候補手を選択する事によりなされる。ベース AI には探索節点数 200,000 の Stockfish7 を用いた。実験により得られた NN は、ベース AI が出力する評価値よりも高い精度で対戦結果を予測することがわかった。しかし、ベース AI よりも強いチェスプレイヤーを生成するには至らなかった。

A study for replacing the boss in “double-Fritz with boss” by neural networks

SHIN SUGAWARA^{1,a)} KUNIHITO HOKI^{1,b)}

1. はじめに

認知科学の分野などにおいて、集合知という概念がしばしば用いられる。集合知とは、ある目的を達成するために共同で知的作業を行う個々の集団のことである [1]。このような集合知の研究は、ゲーム AI を題材としても行われ、その成果の一例として double-Fritz with boss が挙げられる [2]。このシステムの概要は次のようである。チェスのベース AI である Fritz が候補手を 2 つボスに提示する。ここで提示された手の内ベース AI がもっとも有効と考える手を最善手、その次に有効と考える手を次善手とする。そして、その一方をボスが選択し着手する (図 1 参照)。このシステムを用いて対戦実験を行ったところ、チェスの強さの指標とされる elo レーティングがベース AI と比較して 150 ほど上回ったとの報告がなされた。ここで、候補手を選択するボスの elo レーティングはベース AI と比較して 450 ほど低い。しかし、こ

の実験における elo レーティングは十分な対戦回数で測定されていないため統計的に有意な結果とは言えない。

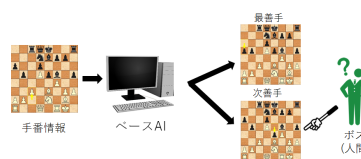


図 1 Double-Fritz with boss の概念図

その一方で、多数決のように機械的に候補手を選択する研究も複数報告されていて、千試合ほどの規模の対戦実験により有意に強い AI が作成されることが確認されている。このような多数決の研究には、複数種類のボードゲームにおいて一定の進展が見られる [3], [4], [5], [6], [7], [8], [9]。しかし、double-Fritz with boss のボスほどの高度な知的判断を伴うような候補手の選択アルゴリズムは未だ提案されていない。

本研究では、チェスのベース AI である Stockfish7 と NN によって構成されたボスを用いて double-Fritz with boss を構築し、ベース AI よりも強い AI を作ることに

¹ 電気通信大学
^{a)} s1211086@gmail.com
^{b)} kunihito.hoki@gmail.com

挑戦する。Stockfish7 はオープンソースのソフトウェアの中で最も強いチェス AI の 1 つである。2014 年 8 月に行われた世界上位のプロプレイヤーとのハンデキャップ戦の成績も話題となった。本研究で取り組む具体的な目標は以下のようである。

- NN を用いて、Stockfish7 の自己対戦の結果を予測する。また NN の大規模化に伴う予測精度の変化の様子を検証する。
- 予測結果がより良い候補手を選択することにより、ベース AI の性能よりも高い性能をもつ AI が作られるのか調査するため、対戦実験を行う。また対戦から得られたデータから NN を用いたボスの性能を定量的に分析する。

2. 関連研究

Althöfer らはゲーム AI が提示した複数の意見の中から一つを人間が選択する multiple choice system の研究をチェスや囲碁を題材として行った [2]。この研究では、double-Fritz with boss (図 1 参照)、3-hirn や list-3-hirns などの様々な形態の multiple choice system が提案されている。ここで各形態の multiple choice system について説明する。3-hirn は異なる 2 つのベース AI が提示した 2 つの候補手のうち一方をボスである人間が選択するシステムである。List-3-hirns は 3-hirn と double-Fritz with boss を組み合わせたシステムである。2 つのベース AI を用意し、それらの AI に double-Fritz with boss と同様に最善手と次善手を提示させる。2 つの AI から提示された計 4 つの候補手の中からボスである人間が候補手を選択する。

Althöfer によって行われたそのほとんどの実験において、統計的に有意な結果ではないが、ベース AI よりも概ね強さが向上したことが確認された。本研究はこの研究の double-Fritz with boss から着想を得て始めたものであるが、先行研究では人間が一つの候補手を選択し、本研究では NN が一つの候補手を選択する点に相違が見られる。

多数決のような簡易なアルゴリズムに基づき、機械的に候補手の選択を行う研究もチェスにおいて複数存在する。大森らは、乱数を用いて複数の異なるベース AI を生成し、これらが提示した候補手を一つ選択することで、ベース AI よりも強い AI が生成されると報告した [3], [4]。Spoerer らは多数決が有効に働くいくつかの条件を異なる 3 種のベース AI を用いて調査した [5]。さらに、チェス以外のボードゲームである 5 五将棋、将棋、チェッカーなどでも多数決は有効であることが報告されている [6], [7], [8], [9]。

近年、ゲームにおいて NN が強い AI の開発に貢献している。AlphaZero や DeepStack, TD-gammon などは、NN によって状態価値関数、行動価値関数を精密に近似し

ており、プロよりも強い実力を持っている [10], [11], [12]。AlphaZero はチェスにおける成功例の一つであり、現在の手番を含む 8 手前までの駒の配置や、手番、手数、キャスリングの有無、50 手ルールのカウントなどを畳込み NN (CNN) に入力して着手確率や期待勝ち点を推定する。この研究で用いられた CNN はフィルタ数が 256、層の数が 20 以上となっている。本研究で用いた CNN の規模は、AlphaZero のものより小規模なものである。

3. Stockfish7 同士の対戦の結果予測

この章では、ベース AI の対戦の結果である勝ち、引き分け、負けの予測を行う。まず、CNN に入力するデータ集合の採取、手番情報の数値化、ベース AI の出力情報の数値化、CNN の構成などの実験方法について述べる。最後に実験結果を示す。

3.1 学習データ集合とテストデータ集合

本研究では実際の対戦で到達しそうな手番を採取するため、チェスの序盤データベースとベース AI を用いて対戦記録を多数生成した。このデータベースには、強いチェスプレイヤーの棋譜およそ 27 万局で 12 回以上採用されている着手が登録されている。学習データ集合は手番 m と対戦結果 d の組の集合 $T = \{(m_1, d_1), \dots, (m_n, d_n)\}$ である。チェスの状態数はおよそ 10^{47} ほど存在する*1。しかし、これらの状態全てを列挙することは現実的でない。そのため、実際の対戦で到達しそうな手番を重点的に用いて CNN の学習を行いたい。

対戦記録は次のように作られる。序盤データベースに手番が登録されている場合、これから指し手を選択する。登録されていなければ、98%の確率でベース AI の最善手、2%の確率で次善手を選択する。ここで、対戦開始から終了までの手番とその対戦結果を $m_1 \dots m_{T-1} r$ と表し、最後に序盤データベースから着手した手番を m_b 、最後に次善手を選択した手番を m_s とする。但し、1 つの対戦で一度も次善手が選択されていなければ、 $s = 1$ とする。ここで、最善手のみを選択して対戦結果が得られた手番を採取したい。従って、 $b \geq s$ ならば手番 $m_{b+1} \dots m_{T-1}$ と結果 r を採取する。 $b < s$ ならば最後に次善手が選択された手番から終局までの偶数手番 $m_s m_{s+2} \dots$ と結果 r を採取する。次善手が一度も選択されない対戦は 2 割以下であった。また、最善手と次善手の評価値両方が区間 $[-1, 000, 1, 000]$ に属さない手番は採取しないものとした。さらに、最善手を選択した手番と次善手を選択した手番の比率がおおよそ 1 になるように、最善手を選択した手番は一定確率で学習時に棄却して、CNN の学習を行った。

テストデータ集合も学習データ集合と同様に生成する。

*1 John's Chess Playground,
<https://tromp.github.io/chess/chess.html>

違いは1つの対戦から採取可能なデータの組のうち、ランダムに1組のみを採取した点にある。今回の実験で用意した学習データ集合はおよそ280万組、テストデータ集合はおよそ10万組からなる。学習データ集合の要素の重複は10%ほど、テストデータ集合のそれは0.5%ほどだった。

3.2 手番情報の数値化

本研究では、ベースAIから得られる出力情報と手番情報を基に各候補手の特徴を抽出しCNNの入力とする。以下に抽出した特徴とその表現方法を記述する。また各特徴はチェス盤のマス目に合わせて、それぞれ64個の数値を8行8列に並べて表現する。

特徴1.候補手の評価値

ベースAIが出力する評価値をシグモイド関数によって区間 $[1, -1]$ の単精度の浮動小数点数に変換する。変換した数値を64個に複製し並べることで、1チャンネルとする。

特徴2.候補手着手後の駒の配置

手番の駒の配置を0, 1で表現する。駒の種類は相手プレイヤーも合わせると12種類あるため、ポーン、ルーク、ビショップ、ナイト、クイーン、キングの位置を1, それ以外を0で表す12チャンネルと自プレイヤー、相手プレイヤーの全ての駒の位置を1, それ以外を0とする2チャンネルで表現する。

特徴3.候補手着手後のチェスの関する詳細なルール

- 候補手着手後のキャスリングの可否
キングとルークを動かしていないかどうかを1チャンネルで表す。もし、クイーン側でキャスリング可能であるならばクイーン側のルークの位置に1, そうでなければ0, キング側でキャスリング可能であるならばキング側のルークの位置に1, そうでなければ0とし、それ以外の位置はすべて0とする。
- 候補手着手後のアンパッサンの可否
ここではポーンが2マス前に動いた場合、ポーンが通過したマスに1, それ以外を0とし、1チャンネルで次の手番におけるアンパッサンの可否を表現する。また、それ以外の場合は全ビットを0とした。
- 候補手着手後の50手ルールの適応
盤上の駒の数とポーンの位置に変化が無く進んだ手数を1チャンネルで表現する。着手一回あたり0.01として、手数を区間 $[0,1]$ の単精度の浮動小数点数に変換し、64個に複製し並べた。

特徴4.候補手に対応した最善応手系列の末端の手番情報
候補手着手後から最善応手系列を指していったときの末端節点の特徴を表現する。駒の配置やキャスリング、アンパッサンの可否、50手ルールも前述と同様に表現し、また最善応手系列の最後の手を指したときの手番が自プレイヤーのとき全ビットが0, 相手

プレイヤーのとき全ビットを1とした1チャンネルで手番を表現する。

これらの入力の他に、全ビット1とした1チャンネルを加えた。なお、CNNが予測する結果は先手の勝ち、引き分け、負けであり、候補手着手後が後手の手番ならば、盤面、評価値および最善応手系列を先手の手番と見做せるように変換した。

CNNに入力として与える特徴を限定することで、double-Fritz with bossに有用な特徴を分析したい。そこで、表1のように異なる複数の入力を用意した。どの入力も37チャンネルとして、未使用のチャンネルは0で埋めることとした。

表1 各入力

入力	特徴
A	1
B	1,2
C	1,2,3
D	1,2,3,4

3.3 CNNの構成

本研究では2つの異なるCNN(CNN1とCNN2)を構築した。まずCNN1について述べ、次にCNN2について述べる。

CNN1は畳み込み9層と全結合1層の計10層から構成される。入力層から続く畳み込み8層は、それぞれC枚のフィルタからなる。これらの畳み込み8層は入力と出力のサイズが変わらないよう、ストライドは1, パディングは1, フィルタのカーネルサイズは 3×3 とした。活性化関数にはReLUを用いた。CNNの名称には括弧付きでCの値を記す。畳み込み8層から続く9層目の畳み込み層のフィルタは1枚とした。この畳み込み層も同様に、ストライドは1, パディングは1, フィルタのカーネルサイズは 3×3 , 活性化関数はReLUとした。全結合層は選択された候補手の勝ち、引き分け、負けの確率分布に対応する3つの数値を出力する。これらの数値はソフトマックス関数によって正規化される。最も大きな数値に対応する結果を、このCNNの予測とする。

CNN2はCNN1の全結合層の出力数を1つにしたものである。出力はシグモイド関数によって変換される。CNN1との違いは出力される数値が結果でなく、期待勝ち点(勝ち1, 引き分け0.5, 負け0)となっている点にある。CNN2は出力された数値が0.66より大きいとき勝ち, 0.33より小さいとき負け, それ以外は引き分けと予測する。

CNN1とCNN2どちらも、結果と予測が一致していたとき予測が正しいとして、予測正答率を計測する。

深層学習用フレームワークcaffe(version 1.0.0)を用いて、交差エントロピーを最小化して重みを調整する。こ

これは, CNN1 は Adam[14], CNN2 は確率的勾配降下法によって行う. 1 バッチのサイズを 100 とし, 1,000 バッチ処理するたびにテストデータ集合から 1,000 バッチ分の組 (X, d) を用いて予測正答率を計算した.

CNN を用いた結果の予測に加えて, ベース AI が出力する評価値と結果 d の分布のみから結果を予測する方法 (Th 法) も用いる. この方法では, まず, 2 つの閾値 a, b ($a > b$) を定める. そして, 評価値が a 以上のときは勝ち, 評価値が b 以下のときは負け, それ以外の場合には引き分けと予測する. 閾値 a, b の決定は, 学習データ集合の結果と評価値の分布から, この予測正答率が最も高くなるように行う.

3.4 実験結果

CNN の学習曲線がおおよそ収束したときの予測正答率を表 2 に示す. 不確かさは標準誤差から見積もられた

表 2 各手法による勝ち, 引き分け, 負けの予測正答率

手法	予測正答率
Th 法	0.867 ± 0.002
CNN1-A(32)	0.867 ± 0.002
CNN1-B(32)	0.876 ± 0.002
CNN1-C(32)	0.878 ± 0.002
CNN1-D(1)	0.872 ± 0.002
CNN1-D(2)	0.872 ± 0.002
CNN1-D(4)	0.872 ± 0.002
CNN1-D(8)	0.876 ± 0.002
CNN1-D(16)	0.877 ± 0.002
CNN1-D(32)	0.878 ± 0.002
CNN2-D(32)	0.874 ± 0.002

95%信頼区間である. ベース AI の評価値のみを用いて結果を予測する CNN1-A(32) と Th 法の正答率に有意な差がないことから, CNN1-A(32) の実験はおおよそ正しくなされていたと考えられる. なお, CNN1-A(32) が評価値から結果を予測する様子は図 2 に示す.

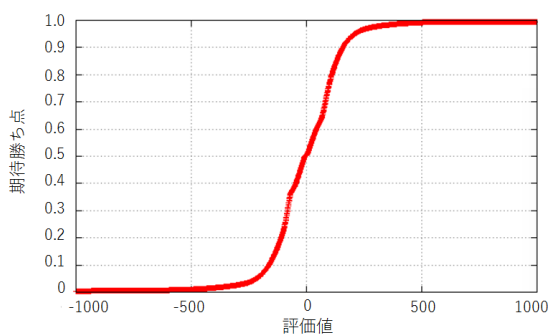


図 2 CNN1-A(32) に入力した評価値に対応する期待勝ち点

手番情報を入力に加えた CNN1-B(32), CNN1-C(32), CNN1-D(32) は, 評価値のみを入力とした CNN1-A(32)

と比較して, 高い予測精度を示した. その一方で, 詳細なチェスのルールを入力する CNN1-C(32) や, それに加えて最善応手系列の末端節点における手番情報を入力する CNN1-D(32) の予測正答率は, CNN1-B(32) と比較して, 有意な差が見られなかった.

CNN1-D(1), CNN1-D(2), CNN1-D(4) の予測正答率には有意な差は見受けられないが, これらと CNN1-D(16), CNN1-D(32) との比較から畳込み層のチャンネル数の増加に従い, 予測正答率が増加する傾向にあることが伺える. Th 法と CNN1-D(1) の予測正答率の比較から, 少なくともフィルタ数 1 の CNN でも評価値以上の特徴を認識できていると推測される.

表 3, 4, 5 に Th 法と CNN1-D(32) の予測と結果の分布を示す. 表 3 と 4 より, Th 法と比較して CNN1-

表 3 CNN1-D(32) の予測と正答の比較

	勝ち (正答)	引分 (正答)	負け (正答)
勝ち (予測)	18,242	1,689	231
引分 (予測)	4,177	48,091	3,869
負け (予測)	384	1,940	21,177

表 4 Th 法の予測と正答の比較

	勝ち (正答)	引分 (正答)	負け (正答)
勝ち (予測)	17,455	1,446	82
引分 (予測)	5,094	48,346	4,489
負け (予測)	254	1,928	20,706

表 5 Th 法の予測と CNN1-D(32) の予測の比較

	勝ち (CNN)	引分 (CNN)	負け (CNN)
勝ち (Th)	18,405	578	0
引分 (Th)	1,757	54,798	1,374
負け (Th)	0	761	22,127

D(32) の勝ち, 負けの予測数が増え, 引き分けの予測数が減っている. また表 5 より, Th 法と CNN1-D(32) の予測を比較すると, Th 法が勝ちと予測しているときに CNN1-D(32) が負け, Th 法が負けと予測しているときに CNN1-D(32) が勝ちと予測することは無いことが伺える. これらのことより, Th 法の誤った引き分けの予測が CNN1-D(32) では正しく予測される傾向が伺える.

4. CNN を用いたボスの対戦実験

4 章では 3 章で構築した CNN を用いて候補手の選択を行うボスを構築し, その性能を調査した. このボスは, 学習データの採取方法と同様に序盤データベースを用いて, さらに評価値が区間 $[-1,000, 1,000]$ に属さない次善手は選択しないとされた. そして, 最善手のみを選択するボスと 3,000 回対戦させたところ, どの構成の CNN を

用いても対戦実験で得られた期待勝ち点には、0.5 と有意な差がみられなかった。結果の予測精度のわずかな向上が必ずしも期待勝ち点の向上に結び付いている訳ではないことがわかる。

予測精度と期待勝ち点の相関をより詳しく調査するため、バッチ処理回数ごとの CNN1-D(32) の期待勝ち点と予測正答率の関係を図 3 に示す。横軸はバッチ処理回数、縦軸は期待勝ち点を表していて、赤い点が期待勝ち点、点に付随する赤い線が標準誤差から見積もられる 95%信頼区間、緑の点が予測正答率である。図 3 より、バッチ処理回数 10^5 から 10^6 で予測正答率と期待勝ち点が急激に増加していることがわかる。これより、予測正答率と期待勝ち点にはある程度の相関があることがわかる。しかし 10^7 以降は、予測正答率はバッチ処理回数に伴わずかに増加しつつづけているが、期待勝ち点は有意に増加していない。

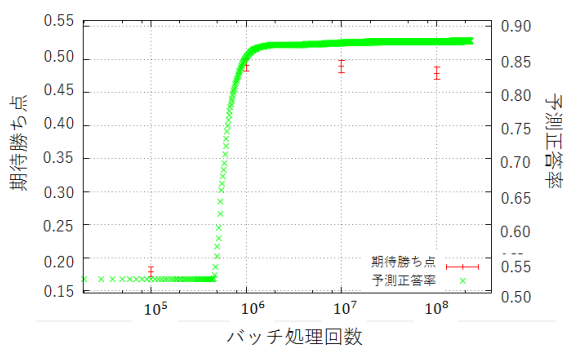


図 3 バッチ処理回数の増加に伴う予測正答率と期待勝ち点の推移

対戦実験において、序盤データベースによる着手数、CNN1-D(32) による着手数を表 6 に示す。全体の 60% 以上の手番においてボスによる候補手の選択がなされていたことがわかる。そのうち、およそ 17% 程度の手番で次善手が選択されていることが確認された。これらのことから、期待勝ち点は 0.5 と有意な差はなかったが、次善手の選択は確かななされていたことがわかる。また、参考のため、図 4 には次善手選択確率の手数依存性を対戦結果で場合分けして示した。

表 6 3,000 回のプレイにおける CNN1-D(32) の着手選択方法の分布 (総着手数は 264,494)

着手の分類	数
序盤データベース	59,542
評価値区間内の着手	178,342
次善手の着手	31,383

ここで、次善手の選択がどの程度期待勝ち点を低下させ得るのかを確かめるために、候補手の評価値の差が一定の数値 v 以下のとき無作為に選択するようなボスを用いて対戦実験を行った。 v を 1, 2, 4, 8, 16, 32 とし常に最善手を選択するボスと 3,000 戦させた結果を表 7

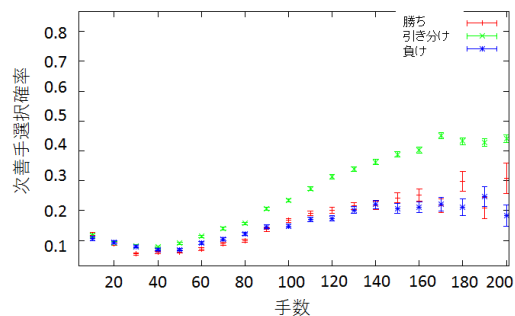


図 4 各手数における次善手選択確率の推移

に示す。 v が 4 の場合は期待勝ち点に 0.5 と有意な差がなく、かつ、次善手選択確率も CNN1-D(32) と同程度であった。従って、本研究で構成した CNN を用いたボスの性能は、評価値のみを見て次善手をときおり選択するような方法と同程度である。なお、参考のため、CNN を用いたボスが次善手を選択した場合の、候補手の評価値の分布を図 5 に示した。

表 7 候補手の評価値の差が v 以下のとき、無作為に指し手を選択するボス AI の期待勝ち点と次善手選択確率

v	次善手選択確率	期待勝ち点
1	0.13	0.49 ± 0.02
2	0.15	0.48 ± 0.02
4	0.17	0.48 ± 0.02
8	0.23	0.44 ± 0.02
16	0.30	0.37 ± 0.02
32	0.37	0.24 ± 0.02

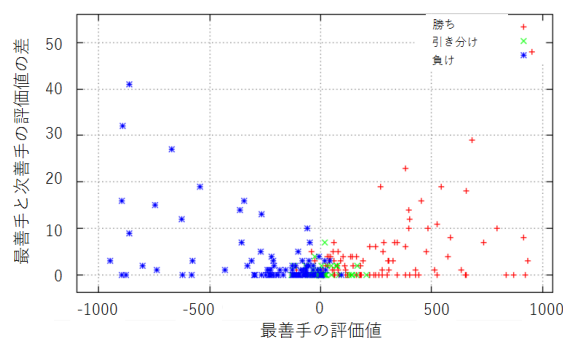


図 5 CNN1-D(32) が勝ち、引き分け、負けでかつ次善手を選択した手番の最善手、次善手の評価値の分布

5. おわりに

本研究では、CNN を用いてチェスの対戦結果の予測を行った。構築した CNN はベース AI の出力する評価値に加えて手番情報を入力する。この CNN は評価値と対戦結果の分布のみから結果を予測する単純な手法以上の予測精度を持つことが確認された。また CNN の大規模化に伴い、予測精度は有意に向上した。

そして, CNN の出力から期待勝ち点を推定し指し手を選択する double-Fritz with boss のボスを構築した. ボスの予測正答率と期待勝ち点にはある程度の相関がみられたが, どの構成の CNN を用いても, 対戦実験で得られた期待勝ち点は 0.5 と有意な差がなく, ベース AI よりも強い AI を生成するには至らなかった. また, CNN を用いて構築したボスの性能は, 評価値をみてときおり次善手を選択するような方法と同程度だった.

CNN のフィルタ数の増加に伴い予測正答率が増加したことから, より大規模な CNN を構築することによって予測正答率を大きく向上させることができれば, 期待勝ち点を向上させ得るのではないかと考えられる.

参考文献

- [1] 赤間世紀. 集合知入門. I/O books. 工学社, 2014.
- [2] Ingo Althöfer and Raymond G. Snatzke. Playing games with multiple choice systems. In *International Conference on Computers and Games*, pp. 142–153. Springer, 2002.
- [3] Kunihiro Hoki, Seiya Omori, and Takeshi Ito. Analysis of performance of consultation methods in computer chess. *Journal of Information Science and Engineering*, Vol. 30, No. 3, pp. 701–712, 2014.
- [4] 大森誠也, 保木邦仁, 伊藤毅志. チェスプログラムを用いた合議アルゴリズムの効果の検証. 研究報告ゲーム情報学 (GI), Vol. 26, No. 5, pp. 1–7, 2011.
- [5] Kristian T. Spoerer, Toshihisa Okaneya, Kokoro Ikeda, and Hiroyuki Iida. Further investigations of 3-member simple majority voting for chess. In *International Conference on Computers and Games*, pp. 199–207. Springer, 2013.
- [6] 小幡拓弥, 埴雅織, 伊藤毅志. 思考ゲームによる合議アルゴリズム 単純多数決の有効性について. 研究報告ゲーム情報学 (GI), Vol. 22, No. 2, pp. 1–5, 2009.
- [7] 伊藤毅志, 小幡拓弥, 杉山卓弥, 保木邦仁. 将棋における合議アルゴリズム—多数決による手の選択. 情報処理学会論文誌, Vol. 52, No. 11, pp. 3030–3037, 2011.
- [8] 杉山卓弥, 小幡拓弥, 齋藤博昭, 保木邦仁, 伊藤毅志. 将棋における合議アルゴリズム—局面評価値に基づいた指し手の選択. 情報処理学会論文誌, Vol. 51, No. 11, pp. 2048–2054, 2010.
- [9] Danilo S. Carvalho, Minh L. Nguyen, and Hiroyuki Iida. An analysis of majority voting in homogeneous groups for checkers: Understanding group performance through unbalance. In *Advances in Computer Games*, pp. 213–223. Springer, 2017.
- [10] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhruv Kumar, and Thore Graepel. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.
- [11] Matej Moravčík, Martin Schmid, Neil Burch, William Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert level artificial intelligence in heads-up no-limit poker. *Science*, Vol. 356, No. 6337, pp. 508–513, 2017.
- [12] Gerald Tesauro. TD-gammon: A self-teaching backgammon program. In *Applications of Neural Networks*, pp. 267–285. Springer, 1995.
- [13] 岡谷貴之. 深層学習. MLP 機械学習プロフェッショナル

- ルシリーズ. 講談社, 2015.
- [14] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014