

平成 30 年度 東京理科大学 卒業論文

囲碁における着手パターンを用いた AI 検知

東京理科大学 理工学部 情報科学科

武田研究室 4 年

6315035 川口航平

指導教員

武田正之

松澤智史

目次

第1章	はじめに	1
1.1	研究背景	1
1.2	研究の目的	1
1.3	本論文の構成	1
第2章	基礎知識	2
2.1	囲碁	2
2.2	家庭用囲碁 AI とソフト打ち	4
2.2.1	家庭用囲碁 AI	4
2.2.2	ソフト打ち	5
2.3	テンプレートマッチング	6
2.4	深層学習	7
2.4.1	パーセプトロン	7
2.4.2	深層ニューラルネットワーク	8
2.4.3	活性化関数	9
2.4.4	損失関数	10
2.4.5	Zeropadding	10
2.4.6	Dropout	10
2.4.7	データセットの正規化	11
2.4.8	評価指標	11
第3章	関連手法	12
3.1	一手あたりの思考時間を測る	12
3.2	ソフトの第1候補手と実際の着手との一致率を見る	12
第4章	提案手法	13
4.1	概要	13
4.2	データセットの作成	13
4.3	ニューラルネットワークの構築	15
第5章	実験と評価	16
5.1	データセット	16
5.2	ニューラルネットワークの構成	17

5.3	実験結果	19
第 6 章	評価と考察	21
6.1	評価	21
6.2	考察	21
6.2.1	自作 AI と Zen	21
6.2.2	私自身と Zen	22
第 7 章	おわりに	24

目 次

2.1	石の置き方	2
2.2	対局例	3
2.3	天頂の囲碁 7 Zen	4
2.4	ソフト打ちを行う場合の画面	6
2.5	テンプレートマッチング	7
2.6	パーセプトロン	7
2.7	深層ニューラルネットワーク	8
2.8	ReLU 関数	9
2.9	sigmoid 関数	10
4.1	Zen メイン画面	13
4.2	候補手一覧	14
4.3	正解データセット作成手順	14
4.4	ニューラルネットワーク	15
5.1	正解データセット	16
5.2	不正解データセット	16
5.3	ニューラルネットワークの構成	18
5.4	正規化なしニューラルネットワーク	19
5.5	正規化ありニューラルネットワーク	20

表 目 次

2.1	各ゲームの探索空間	3
2.2	Zen オプション	5
5.1	パラメータ	17
6.1	その他の評価指標（正規化なしニューラルネットワーク）	22
6.2	その他の評価指標（正規化ありニューラルネットワーク）	22
6.3	その他の評価指標（正規化なしニューラルネットワーク）	23
6.4	その他の評価指標（正規化ありニューラルネットワーク）	23

第1章 はじめに

1.1 研究背景

近年、AIを用いたウィルスやマルウェアといった悪性プロセスによるサイバー犯罪が増加している。それに伴って、セキュリティ対策をする側もAIを用いることで検知技術が向上している。しかし、通常悪性ではないプロセスを、使用が認められない状況において悪用していることの検知は困難である。

2016年3月にGoogle DeepMind社が開発した囲碁AI「AlphaGo」[1]がトップ棋士李世ドル¹に勝利した。それに続く形で、各国が家庭用PCで動く囲碁AIを発表した。強い相手といつでも打てることや、研究・検討に非常に有用であるので急速に広まった。それと同時に、インターネット上で対局ができる囲碁サイト（以後囲碁サイトと表記する。囲碁だけに）において、AIに勝率の高い着手を探索させて打つ手法（以下ソフト打ちと表記）が横行した。しかし、AIを使用しているという確証を得ることは非常に困難である。囲碁サイトは日本囲碁界における最大の収入源なので、この衰退は囲碁界全体の衰退に直結する。したがって、ソフト打ちの明確な対策が早急に求められている。

1.2 研究の目的

囲碁の棋譜を入力として、対局者が家庭用囲碁AIの候補手を参照しているかどうかを判定することを本研究の目的とする。

1.3 本論文の構成

本論文の構成は、以下のとおりである。

第2章では、基礎知識を述べる。

第3章では、関連手法を述べる。

第4章では 提案手法を述べる。

第5章では、実験方法、実験結果を述べる。

第6章では、評価と考察を述べる。

第7章では、本論文のまとめを述べる。

¹李世ドル <https://ja.wikipedia.org/wiki/李世ドル>

第2章 基礎知識

2.1 囲碁

囲碁は黒と白の石を用いて、陣地を囲み合う対戦ゲームである。黒が先手、白が後手となり、交互に石を置いていく。石は升目の交点に置くことができる（図 2.1）。

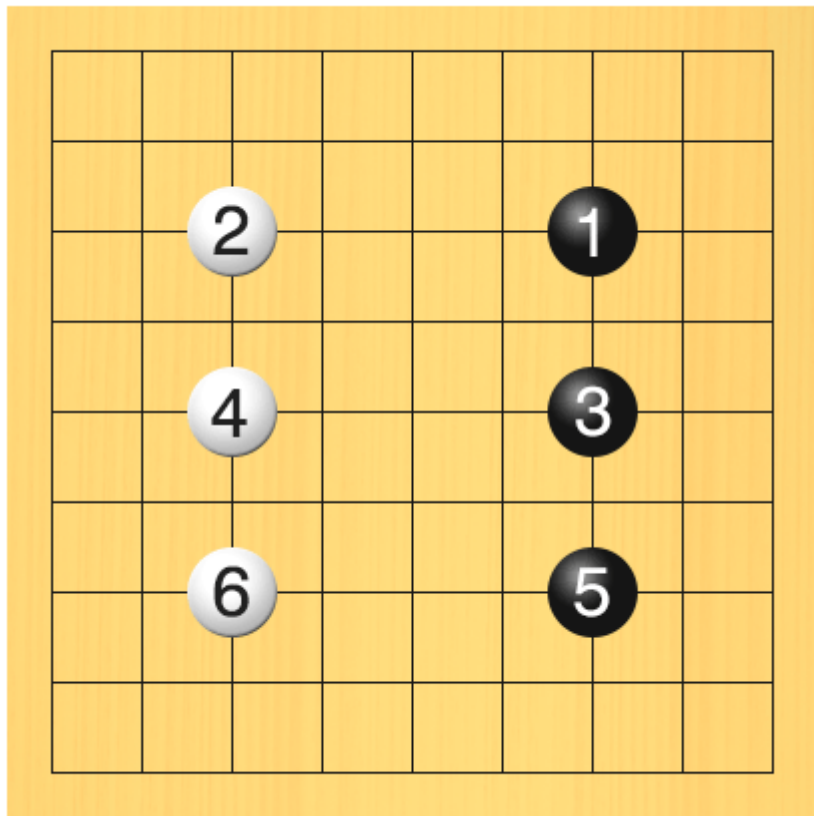


図 2.1: 石の置き方

そして陣地をより多く囲った側の勝利となる。対局例（図 2.2）からどこが陣地になるか示す。

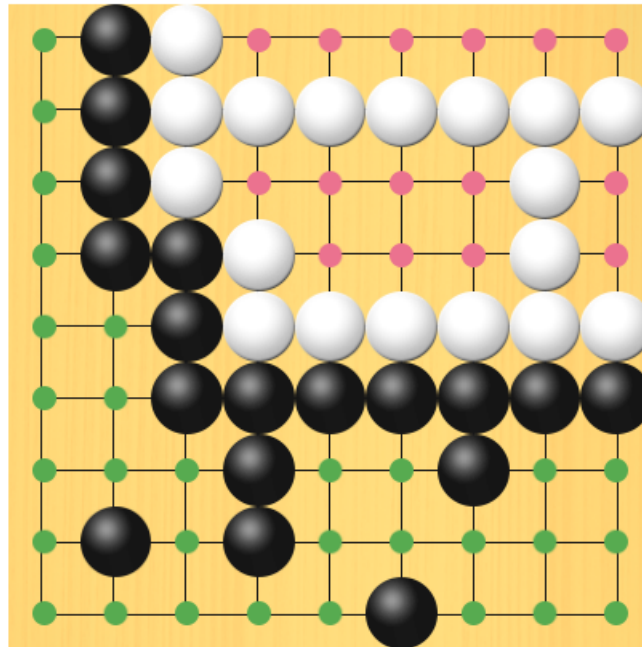


図 2.2: 対局例

左上から右下にかけて、黒石に接している丸印が黒の陣地である。また、右上一帯の白石に接している丸印が白の陣地である。それぞれの丸印の数を数えると、黒の陣地は 30 目、白の陣地は 15 目。よって“黒の 15 目勝ち”となる。また、相手の石の四方を自分の石で囲むと、囲んだ石を取り上げることができる。

上の図では 9×9 の碁盤を用いた対局であるが、実際には 9×9 、 13×13 、 19×19 の碁盤が存在する。最も使用されているのは 19×19 の碁盤である。有名な二人零和有限確定完全情報ゲームの探索空間 [2] を以下の表 2.1 に示す。

表 2.1: 各ゲームの探索空間

ゲーム	探索空間
オセロ	10^{28}
チェス	10^{50}
将棋	10^{71}
囲碁 (9 路盤)	10^{38}
囲碁 (19 路盤)	10^{171}

本研究では 9×9 の碁盤を用いる。

2.2 家庭用囲碁 AI とソフト打ち

2.2.1 家庭用囲碁 AI

日本で最も普及している家庭用囲碁 AI は「天頂の囲碁 7 Zen」¹(以下 Zen と表記) である (図 2.3)。

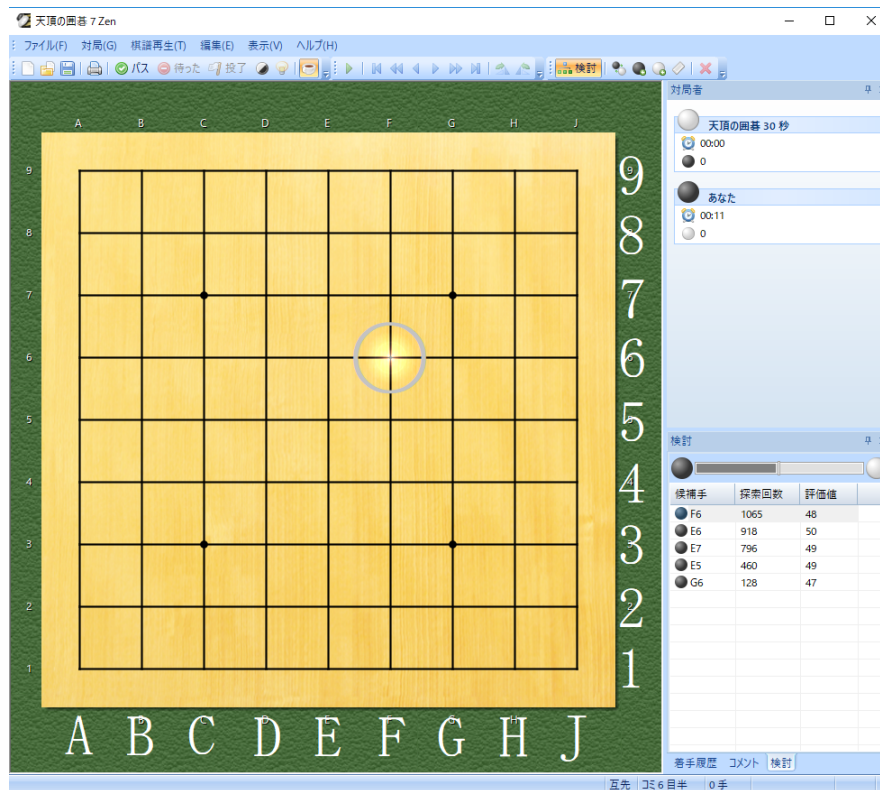


図 2.3: 天頂の囲碁 7 Zen

¹天頂の囲碁 7 Zen <https://book.mynavi.jp/tencho7/>

左には碁盤が大きく表示される。右上には黒番、白番それぞれの対局者、消費時間、取った石の数が表示される。右下には現在の盤面における評価グラフ、評価値が高い候補手、その探索回数と評価値が示されている。評価値の数値は勝率を示すパーセンテージである。また最も評価値が高くなると見込まれる手は、左側の大きな盤面に表示される（図 2.3 では F6）。機能は大きく分けて以下の 3 つである。

1. 対局モード メインとなる機能。Zen と対局できる。
2. 自己対戦モード コンピュータ同士に対戦させた棋譜を入手できる。
3. 検討モード 棋譜を入力して、候補手とその評価値を見ることができる。

本研究で課題となるのは検討モードの悪用である。また、Zen を使用する際選択できるオプションを表 2.2 に示す。

表 2.2: Zen オプション

碁盤サイズ	手番	思考時間	段級位
9 × 9	黒番	5 秒	6 級
		10 秒	5 級
15 秒		⋮	
20 秒		1 級	
13 × 13	白番	30 秒	初段
19 × 19		⋮	⋮
		120 秒	九段

思考時間を使用者が決めることが出来るため、どのような条件で対局が行われたとしても、悪用することが可能となる。

2.2.2 ソフト打ち

本稿では、大手囲碁サイト「東洋囲碁」²でソフト打ちを行う際の実例を述べる。Zen の画面と囲碁サイトを並べて表示する（図 2.4）。

²東洋囲碁公式ホームページ <http://www.toyo-igo.com/>

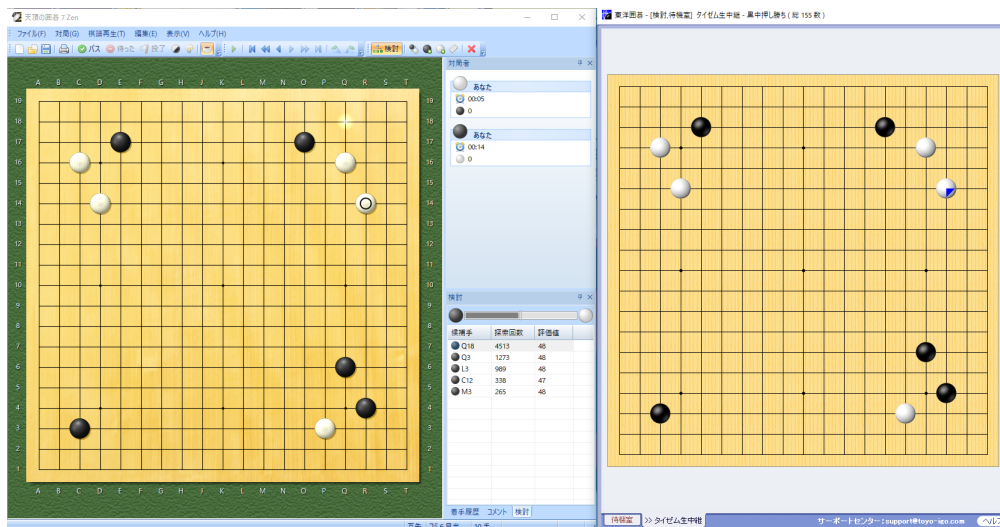


図 2.4: ソフト打ちを行う場合の画面

具体的な手順は以下の通りである。

1. 囲碁サイトで相手が打った手を見る
2. Zen の盤面に入力し、探索させる。
3. 候補手を見て囲碁サイト上の盤面に入力する。
4. 1～3 を終局まで繰り返し行う。

2.3 テンプレートマッチング

テンプレートマッチング (図 2.5) とは、ある画像と同じパターンが画像全体に存在するかを見つけることである。³ テンプレート画像 (探したい画像) と探索対象画像の画素値を比較する。画素値の誤差の二乗和は SSD (*Sum of Squared Difference*) である。画像が完全に一致した場合、 SSD は 0 となる。

$$SSD = \sum_i \sum_j (I(i, j) - T(i, j))^2 \quad (2.1)$$

³テンプレートマッチング <https://www.yukisako.xyz/entry/template-matching>

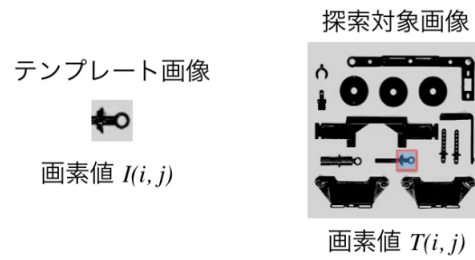


図 2.5: テンプレートマッチング

2.4 深層学習

深層学習とは、その表面的な形を見れば、深い、すなわち、多くの層を持ったニューラルネットワークモデルを用いた機械学習の総称である。[3]

2.4.1 パーセプトロン

ニューラルネットワークの基本形はパーセプトロン（図 2.6）である。パーセプトロンとは、脳の認知機能を模倣したパターン認識のための情報処理モデルである。2 層でのみ構成される最初に提案されたニューラルネットワークであり、図では階層型ニューラルネットワークの一種である。階層型ニューラルネットは、3 層前後のフィードフォワード構造を備えた多層パーセプトロンと呼ばれるものである。

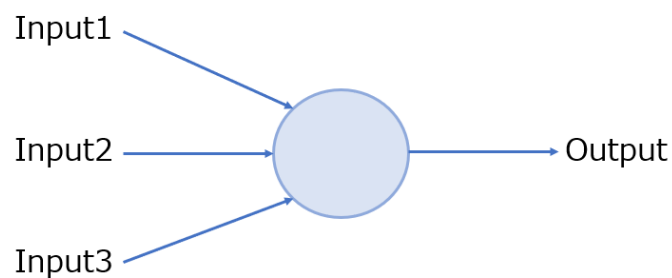


図 2.6: パーセプトロン

2.4.2 深層ニューラルネットワーク

深層ニューラルネットワーク（図 2.7）は、広義には多層のニューラルネットワーク全般のことを指し、狭義にはフィードフォワード構造の階層型ニューラルネットワークを 4 層以上に拡張したものを指す。[3] ニューロンへの入力、入力 $x = (x_1, x_2, \dots, x_n)$ と重み $w = (w_1, w_2, \dots, w_n)$ の積にバイアスである定数 b を加えたもの（式 2.2）である。

$$f(x) = \sum_{k=1}^n x_k w_k + b \quad (2.2)$$

この値に活性化関数を適用したものが出力となる。ニューラルネットワークネットワークモデルに対して明示的な答えが提示されるモデルを教師あり学習と呼び、そのような答えが与えられない場合を教師なし学習と呼ぶ。[4]

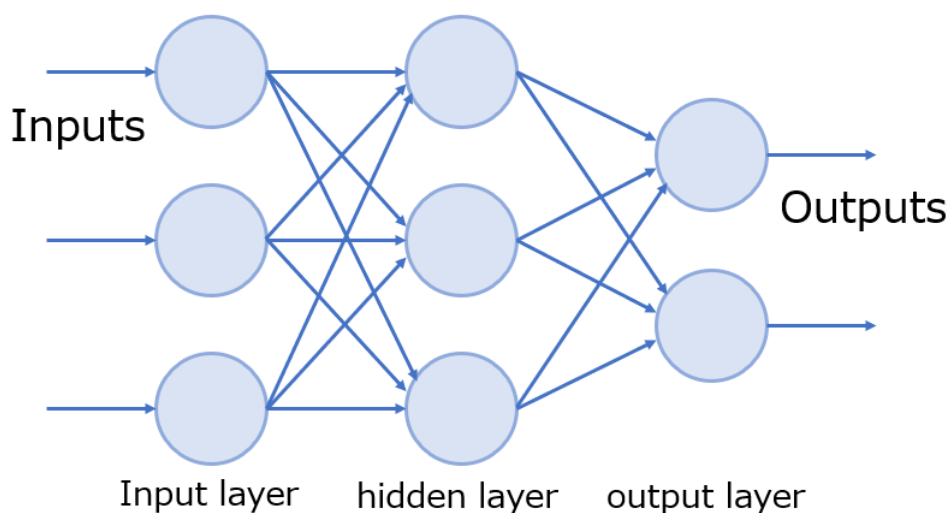


図 2.7: 深層ニューラルネットワーク

2.4.3 活性化関数

入力から算出された値を出力に変換する関数⁴である。

ReLU 関数

入力した値が0以下のとき0になり、1より大きいとき入力をそのまま出力する関数（式 2.3）である。グラフを図 2.8 に示す。

$$h(x) = \begin{cases} x & (x \geq 0) \\ 0 & (x < 0) \end{cases} \quad (2.3)$$

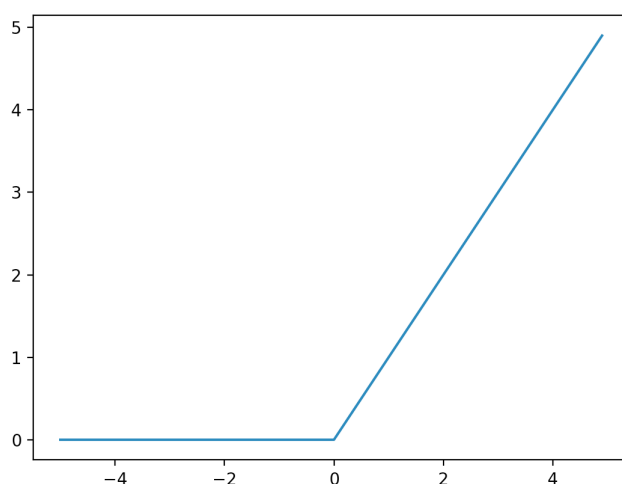


図 2.8: ReLU 関数

sigmoid 関数

入力した値が大きければ大きいほど1に近づき、入力した値が小さければ小さいほど0に近づく関数（式 2.4）である。しかし近づくのみで0,1にはならないため元の入力の値を無視しすぎないメリットがある。グラフを図 2.9 に示す。

$$h(x) = \frac{1}{1 + e^{-x}} \quad (2.4)$$

⁴活性化関数 <https://keras.io/ja/activations/>

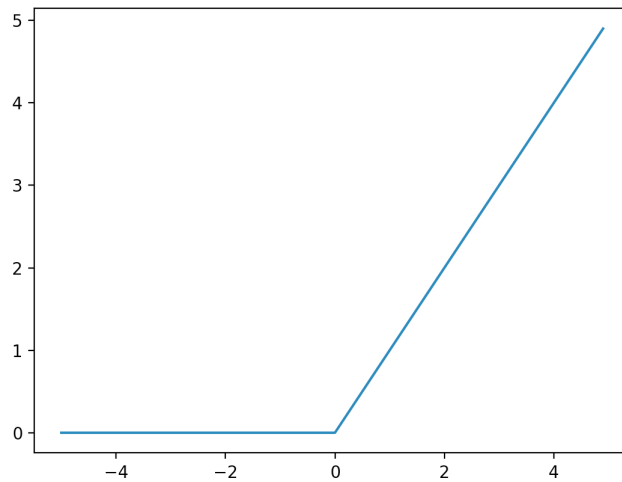


図 2.9: sigmoid 関数

2.4.4 損失関数

ニューラルネットワークが教師データに対してどれだけ適合していないかを算出する関数⁵である。[3] ニューラルネットワークの性能の悪さを示す指標となり、機械学習の目標はこの値を 0 に近づけることである。

binary_crossentropy 関数

1 出力とし、その出力 $y \in [0, 1]$ の値が 0.5 より大きいか否かでクラス判定をする (式 2.5)。

$$E(w) = - \sum_n^N (d_n \log y_n + (1 - d_n) \log(1 - y_n)) \quad (2.5)$$

2.4.5 Zeropadding

Zeropadding とは可変配列を扱う際、固定長に統一するために 0 を代入することである。[3] 入力長を揃えるために用いられる。

2.4.6 Dropout

Dropout とは、ニューラルネットワークの学習時に、一定割合のノードを不活性化させながら学習を行うことである。[3] これによって過学習を防ぎ（緩和し）、精度をあげる。

⁵損失関数 <https://keras.io/ja/losses/>

2.4.7 データセットの正規化

正規化とは、データセットを一定のルールに基づいて変形し、利用しやすくすることである。[3] データセットが単位の異なるもので計算を行ったり、平均と標準偏差を 0、1 にしたりする。

2.4.8 評価指標

機械学習の評価を行う指標のことである。主な指標は精度、適合率、再現率、F 値である。[3] これらの指標の計算には、まず結果を 4 つのパターンに分類する。

1. 真陽性（以下 TP と表記）正解が 1 のときに正しく 1 と予測する。
2. 偽陰性（以下 FN と表記）正解が 1 のときに誤って 0 と予測する。
3. 偽陽性（以下 FP と表記）正解が 0 のときに誤って 1 と予測する。
4. 真陰性（以下 TN と表記）正解が 0 のときに正しく 0 と予測する。

そしてそれぞれの指標は以下のように計算する。

- 精度

精度 (Acc) は、全てのパターンを考慮した上で、予測が正しい割合である。
(式 2.6)

$$Acc = \frac{TP + TN}{TP + FN + FP + TN} \quad (2.6)$$

- 適合率

適合率 (Pre) は、1 (0) と予測したパターンの中で、本当に 1 (0) であった割合である。(式 2.7)

$$Pre = \frac{TP}{TP + FP} \quad (2.7)$$

- 再現率

再現率 (Rec) は、実際に 1 (0) であるものが、正しく 1 (0) と予測された割合である。(式 2.8)

$$Rec = \frac{TP}{TP + FN} \quad (2.8)$$

- F 値

適合率、再現率はトレードオフの関係なので、両方の値がバランス良くなるよう適合率と再現率の調和平均によって求めた値である。(式 2.9)

$$F = \frac{2 \cdot Pre \cdot Rec}{Pre + Rec} \quad (2.9)$$

第3章 関連手法

現在のソフト打ち検知がどのような手法で行われているか示す。

3.1 一手あたりの思考時間を測る

囲碁の一局は大きく分けて序盤・中盤・終盤の三つに分かれる。そしてどの分野で考慮時間を使用するかは十人十色である。一般的には得意な分野で多く考慮時間を使用する。一方 AI を用いてソフト打ちを行う場合、全ての分野において考慮時間が一定になる。しかし、この手法は AI 悪用者が一定の早さで操作を行うという仮定の下に成り立っている。したがって AI 悪用者の操作時間が増減したり、AI にとって難しい場面で探索時間を多めに使用した場合に検知することができなくなってしまう。

3.2 ソフトの第1候補手と実際の着手との一致率を見る

一局を通して AI の第1候補との一致率を調べる手法である。[5] しかし、この手法は第1候補以外の着手を織り交ぜて使用された場合に検知をすることができなくなる。また、「AIっぽい」という定性的な評価しか行うことができないという欠点がある。

第4章 提案手法

4.1 概要

一局中の AI との一致数を入力として、Zen か否かを出力する手法について述べる。着手の一致を調べる際、第 1 候補だけ調べるのではなく、第 2、第 3 候補も検知の対象とする。

4.2 データセットの作成

黒番、白番それぞれが一局中何手目に AI の第何候補手と一致したかを表す配列をデータセットとする。本研究では、AI の第 3 候補までの一致を見ることとする。出力は AI であるか否かの二値となるため、正解データセットは Zen であるもの、不正解データセットは Zen でないものとなる。まずデータセット作成に用いる棋譜ファイルを用意する。囲碁は黒番と白番が交互に打つため、対局者は二人いることになる。したがって黒番、白番それぞれデータセットを作成することになるので、1 枚の棋譜から二つのデータセットが作成される。Zen に棋譜を入力すると、以下のような画面（図 4.1、4.2）が表示される。

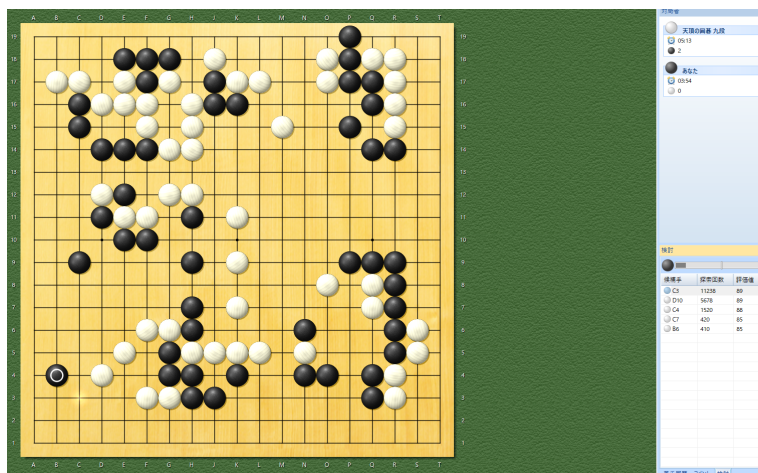


図 4.1: Zen メイン画面

検討		
		
候補手	探索回数	評価値
 C3	11238	89
 D10	5678	89
 C4	1520	88
 C7	420	85
 B6	410	85

図 4.2: 候補手一覧

右下に現在盤面の評価グラフ、AI が示す候補手とその探索回数、候補手に対する評価値が表示される。図 4.2 の候補手一覧をテンプレートマッチングで読み取り、実際の着手と比較することによって第何候補と一致したか、配列に代入していく。

総手数 = n の場合

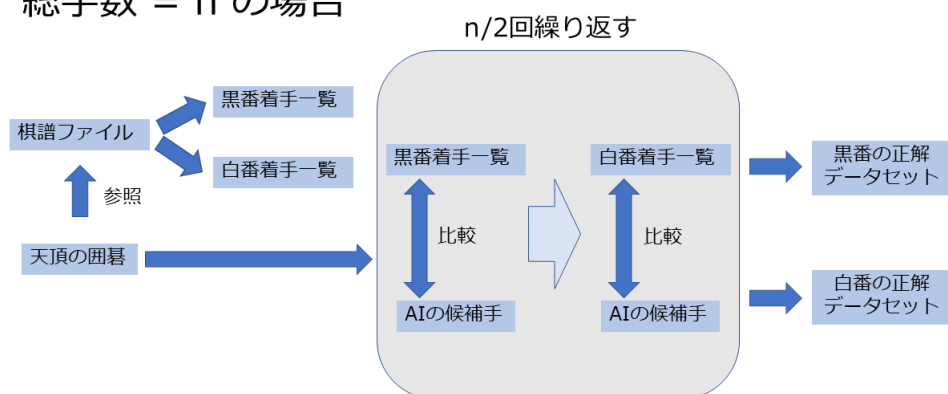


図 4.3: 正解データセット作成手順

配列の先頭には AI か否かを示す値を代入する。正解データセットには index0 に 1 を代入する。例えば、 m 手目が AI の第 1~3 候補と一致したら、その数値を配列の m 番目に代入する。それ以外の着手であった場合は 0 を代入する。また、Zen の棋力を九段、思考時間を 10 秒とした。

例：黒番の第 10 手目が AI の第 2 候補と一致した場合 $\text{black}[10] = 2$

また、不正解データセットには $\text{index}0$ に 0 を、配列の長さを 15～50 の乱数とした。 $\text{index}1$ 以降には 0～3 を乱数で代入した。

以上の方法で正解データセットを 1000、不正解データセットを 1000 用意した。

4.3 ニューラルネットワークの構築

データセットが可変長であるため、0 パディングを行い長さ 51 の固定長である numpy 配列にした。先頭がラベル、それ以外を入力とする。入力は 50 次元となる。本研究では AI か否かの 2 値分類を行うので、出力に `binary_crossentropy` 関数を用いる。また、データセットをトレーニングデータ:テストデータ = 8:2 の割合で分割し、学習させた。出力層は AI であるとき 1、AI でないとき 0 を出力した。構築したニューラルネットワークは以下（図 4.4）のようになっている。

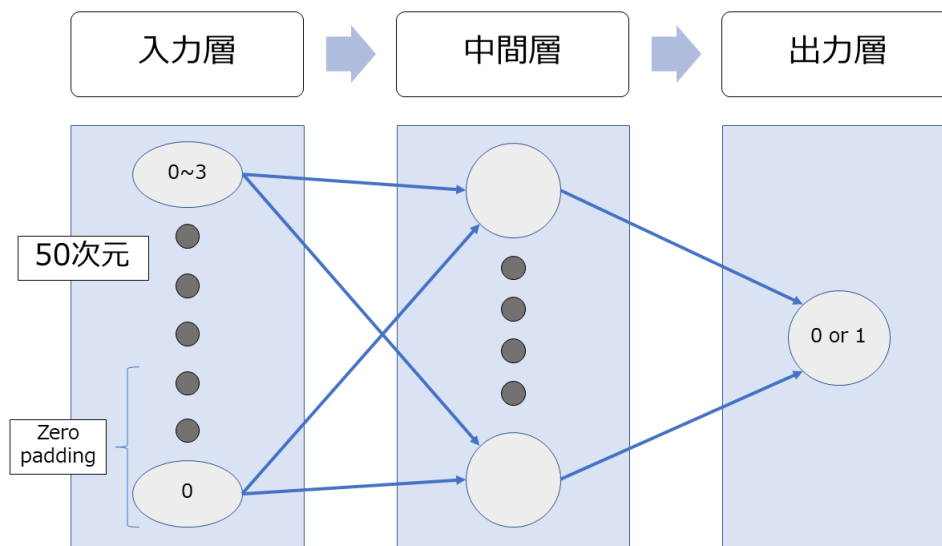


図 4.4: ニューラルネットワーク

第5章 実験と評価

5.1 データセット

実験に使用する正解データセット、不正解データセットは以下のような構成になっている（図 5.1、5.2）。

正解データセット (棋譜の総手数=n手,黒番の場合)

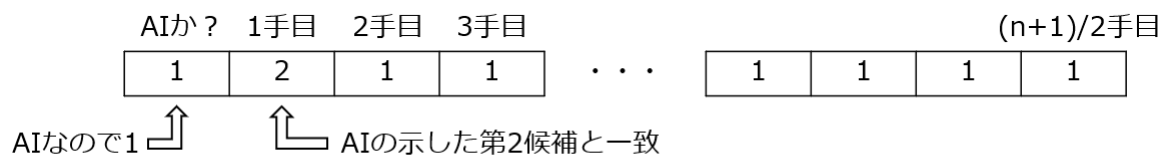


図 5.1: 正解データセット

不正解データセット (nは15~50の乱数)

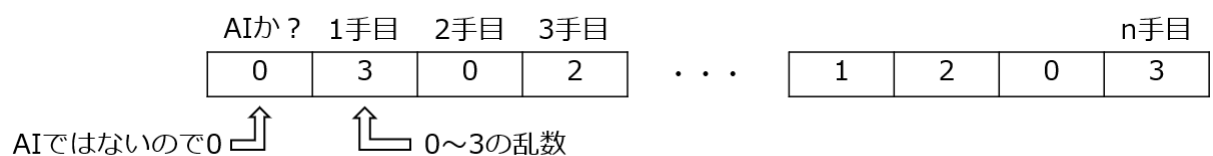


図 5.2: 不正解データセット

5.2 ニューラルネットワークの構成

ニューラルネットワークのパラメータを表5.1に、構成を図5.3に示す。エポック数は100としているが、過学習を防ぐため、コールバック関数¹として EarlyStopping、ModelCheckpoint を使用した。また、適宜 Dropout 層を挿入した。今回、全結合のみで構成したニューラルネットワーク（以下正規化なしニューラルネットワーク）と、データセットに対し正規化を行ってから全結合を行ったニューラルネットワーク（以下正規化ありニューラルネットワーク）を構築した。

表 5.1: パラメータ

入力層	50
中間層	5000
活性化関数	ReLU
バッチサイズ	32
エポック	100
学習率	0.1
データ比 (訓練:テスト)	8:2
validation	0.2

¹コールバック関数 <https://keras.io/ja/callbacks/>

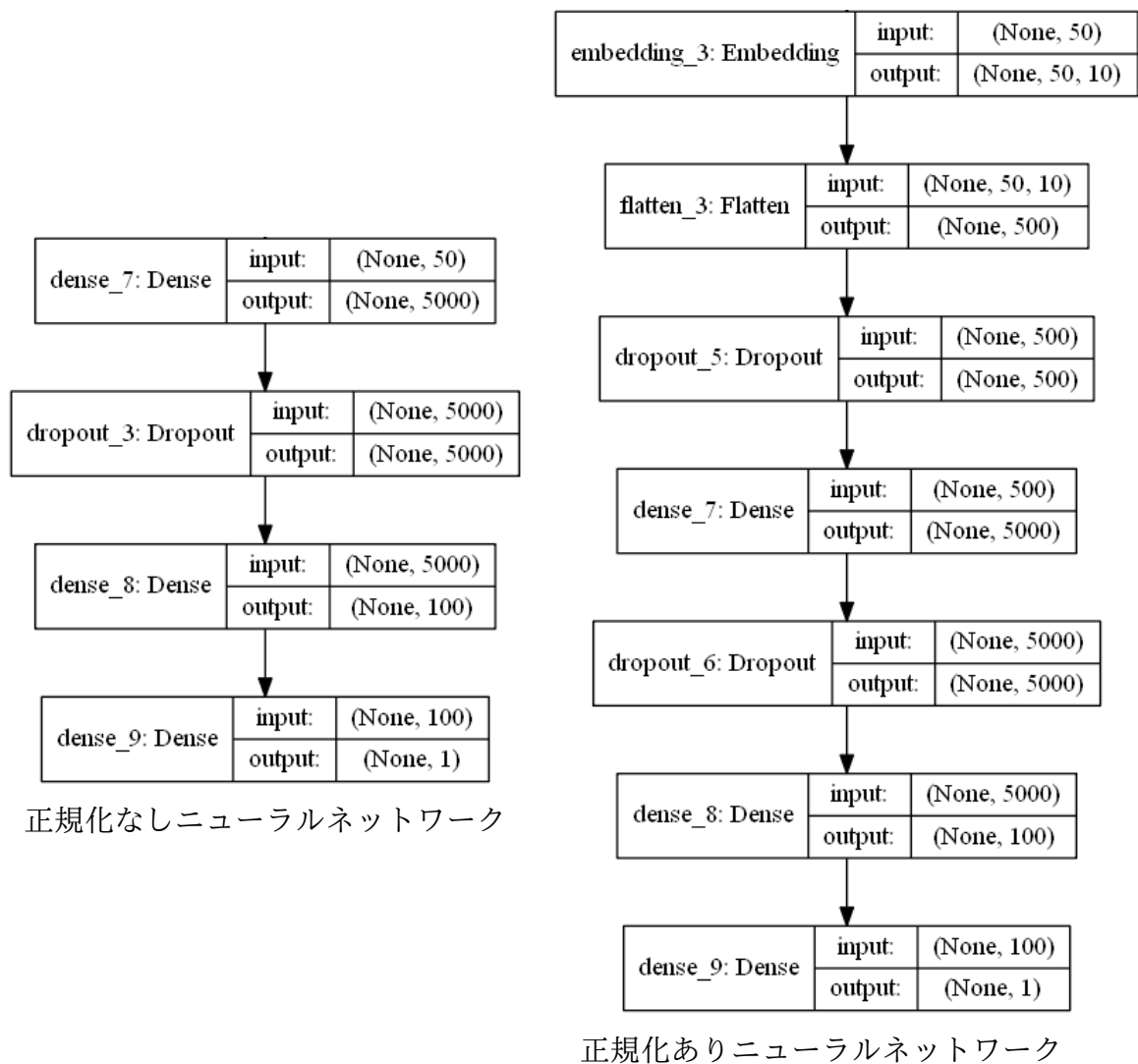


図 5.3: ニューラルネットワークの構成

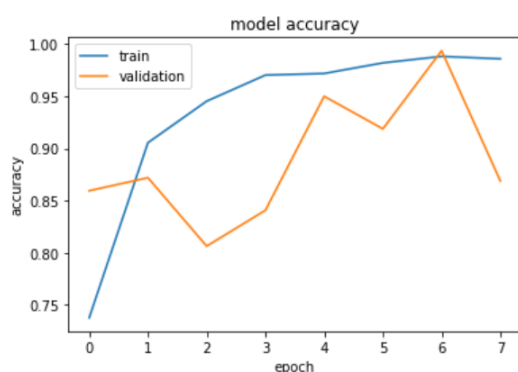
5.3 実験結果

全ての手法を 10 回試行して、評価精度と評価損失の平均、また最後の試行における評価グラフを示す。

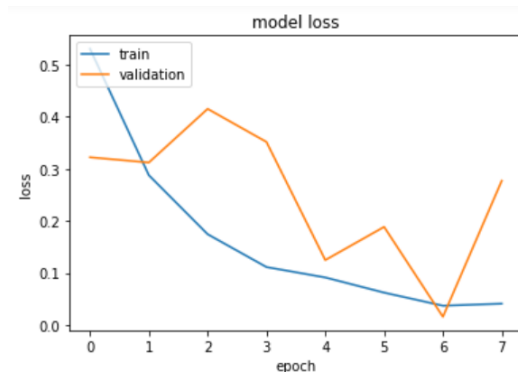
- 正規化なしニューラルネットワーク
エポック数は 7 前後で止まった (図 5.4)。

評価精度 0.92150

評価損失 0.15409



評価精度グラフ



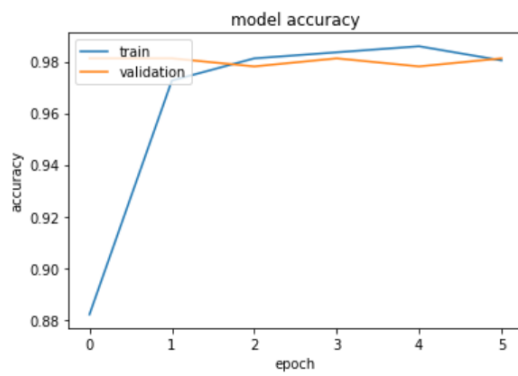
評価損失グラフ

図 5.4: 正規化なしニューラルネットワーク

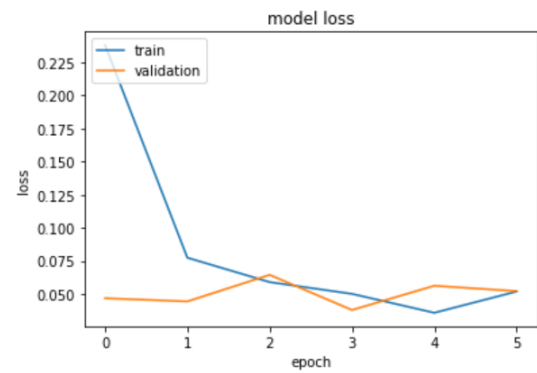
- 正規化ありニューラルネットワーク
エポック数は 5 前後で止まった (図 5.5)。データセットに正規化を加えなかった場合と比較して、評価精度は高い値を、評価損失は小さい値を得ることが出来た。

評価精度 0.99725

評価損失 0.01582



評価精度グラフ



評価損失グラフ

図 5.5: 正規化ありニューラルネットワーク

第6章 評価と考察

6.1 評価

実験では、正規化なしニューラルネットワークと、正規化ありニューラルネットワークを構築した。その結果、正規化ありニューラルネットワークでより良い評価精度、評価損失を得た。

6.2 考察

それぞれのニューラルネットワークが新しいデータセットに対してどれほど適合しているか考察していく。実験段階では Zen の第何候補と一致したか示した配列と、0～3の乱数が代入された配列を比較し、Zen かどうか判定するに留まった。本稿では、私が独自で製作した AI（以下自作 AI と表記）と Zen、私自身が入力した着手と Zen をどの程度判定できるか考察する。それぞれのニューラルネットワークにおいて評価精度、評価損失を示した後、評価指標を用いて適合率、再現率、F 値の評価を行う。参考までに、自作 AI は四段程度、私自身は七段の免状と普及指導員の資格を所持している。¹

6.2.1 自作 AI と Zen

テストデータは自作 AI のものを 25、Zen のものを 25 の計 50 を用意した。それぞれのニューラルネットワークにおいて 10 回試行を行った際の平均の結果を以下に示す。

- 正規化なしニューラルネットワーク

評価精度 0.84600

評価損失 0.35807

- 正規化ありニューラルネットワーク

評価精度 0.85800

¹日本棋院 免状・資格・制度 <http://archive.nihonkiin.or.jp/profile/sikaku/menjo.html>

評価損失 0.47291

正規化ありニューラルネットワークの方が僅かに高い評価精度を示した。また、自作 AI と Zen に棋力の差があるので、いずれのニューラルネットワークにおいても高い精度で判定を行うことが出来た。実際における AI の悪用は自作 AI と同程度の棋力帯で最も行われているので、非常に望ましい結果が得られた。囲碁サイトでは 9×9 の盤面ではなく 19×19 の盤面での対局がほとんどである。したがって探索空間の狭い 9×9 の盤面でこの精度であれば、より探索空間の広い 19×19 の盤面では両者間の棋力差が顕著になるので、さらに高い精度が期待できる。また、その他の評価指標を表 6.1、6.2 に示す。

- 正規化なしニューラルネットワーク

表 6.1: その他の評価指標 (正規化なしニューラルネットワーク)

判定	適合率	再現率	F 値
0	0.970	0.750	0.834
1	0.796	0.972	0.872

- 正規化ありニューラルネットワーク

表 6.2: その他の評価指標 (正規化ありニューラルネットワーク)

判定	適合率	再現率	F 値
0	0.948	0.720	0.810
1	0.783	0.960	0.859

どちらのニューラルネットワークにおいても F 値は高い値を得た。また、判定 1 の再現率が非常に高い値を示しており、これは「実際に AI である (1) ものを AI である (1) と予測できている」ことを表している。それと比較して判定 0 の再現率は少し低い値となっており、これは「実際に AI でない (0) ものを AI でない (0) と予測するのが苦手である」ことを表している。しかし AI でないものを AI であると判定することは AI を見逃す訳ではないので、問題にならない。

6.2.2 私自身と Zen

テストデータは私自身のものを 25、Zen のものを 25 の計 50 を用意した。それぞれのニューラルネットワークにおいて 10 回試行を行った際の平均の結果を以下に示す。

- 正規化なしニューラルネットワーク

評価精度 0.80000

評価損失 0.47555

- 正規化ありニューラルネットワーク

評価精度 0.64200

評価損失 1.13579

正規化なしニューラルネットワークの判定の精度が8割であったことに対し、正規化ありニューラルネットワークにおける判定の精度は6割近くに留まってしまった。このことより、正規化なしニューラルネットワークの方が新しいデータセットに適合していると言える。また、自作 AI と Zen の判定を行った場合より精度が落ちた原因としては、私と Zen の棋力にほとんど差がなく、私の着手が Zen の示す候補手と一致していることが多く見られたことにある。しかし、今回の実験では 9×9 の盤面、Zen の思考時間は 10 秒としている。実際悪用が行われる場面は 19×19 の盤面であり、Zen の思考時間はさらに長くなる。また、 19×19 の盤面で同様の実験を行った場合、私と Zen により大きな棋力差が生じる。そして棋力差が生じた場合判定可能であることは前項で示した通りである。また、その他の評価指標を表 6.3、6.4 に示す。

- 正規化なしニューラルネットワーク

表 6.3: その他の評価指標（正規化なしニューラルネットワーク）

判定	適合率	再現率	F 値
0	0.971	0.712	0.825
1	0.778	0.976	0.862

- 正規化ありニューラルネットワーク

表 6.4: その他の評価指標（正規化ありニューラルネットワーク）

判定	適合率	再現率	F 値
0	0.983	0.248	0.382
1	0.570	0.992	0.723

正規化なしニューラルネットワークにおける F 値が8割以上を保ったのに対して、正規化ありニューラルネットワークにおける F 値がかなり低い値を取った。評価精度の差と合わせて、正規化なしニューラルネットワークがより機能していることが示された。

第7章 おわりに

本研究では、囲碁の棋譜を元に AI かどうか判定するシステムを、深層学習を用いて作成した。このシステムによって対局者が AI を悪用しているか定量的に評価することが可能になった。また、入手可能な AI であれば候補手との一致数がデータとして得られるため、新しい AI も検知することができる。一方、新しく作られた入手が不可能な AI の検知に関しては、候補手と一致しているかという情報が得られないため検知ができないという課題が残った。

本研究を通じて、囲碁 AI の悪用、ひいては良性プログラムの悪用の検知を望める。

謝辞

本研究を進めるにあたって、常に適切なご指導ご鞭撻をいただいた武田正之教授、松澤智史助教、並びにネットワークセキュリティゼミをはじめとする武田研究室の皆さんに深く感謝いたします。

参考文献

- [1] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel and Demis Hassabis : Mastering the game of Go without human knowledge, Nature volume 550, pp354–359, 2017
- [2] 美添 一樹 ‘コンピュータ囲碁における モンテカルロ法 ～理論編～’
- [3] 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 ダヌシカ ‘深層学習’
- [4] 浅川 伸一 ‘Python で体験する深層学習’
- [5] 大槻 知史 ‘最強囲碁 AI アルファ碁 解体新書’