

モンテカルロ囲碁へのニューラルネットワークの応用について

†佐藤 慎也, ‡山本 修身

†名城大学 理工学研究科 情報工学専攻, ‡名城大学 理工学部 情報工学科

1 はじめに

古くから世界で親しまれてきた戦略性の高いゲームの一つに囲碁がある。囲碁では、9×9または19×19の盤面に黒石と白石を交互に配置していく。この盤面上に、地と呼ばれる陣地を作り相手より多くの地を作り上げることが囲碁の目的である。2014年現在最も強い囲碁プログラムにはモンテカルロ法を用いた木探索アルゴリズム、モンテカルロ木探索 (Monte-Carlo Tree Search) [1] が使用されている。モンテカルロ木探索を用いた囲碁プログラムはモンテカルロ囲碁と総称されて世の中に多く存在している。本稿では機械学習のひとつであるニューラルネットワーク (NN) を用いて囲碁における悪手の判別を試みる。これにより、モンテカルロ囲碁の探索に直接の枝刈りおよびモンテカルロ木探索においてサブルーチンとして用いられるプレイアウトにおける枝刈りを行う。NNを用いた囲碁プログラムに NeuroGo [2] がある。NeuroGoの実力は13級程度のレベルまで到達した。これは現在の市販の囲碁プログラムから見れば強くないが、局面の評価するためのNNはある程度まで実現可能であることがわかる。NeuroGoのように既存研究では良手を発見するための研究が主である。これに対して、本稿では悪手を正確に捉える事を目的とした学習を行う。これによりモンテカルロ囲碁のシミュレーションや木探索での枝刈りへの適用に適した学習を目標とする。NNによる悪手判別を試みる理由に、まずNNを使用する利点として入出力が碁盤に対応しやすことがある。また、悪手を判別することは木探索中の枝狩りに直接適用することができるため、モンテカルロ囲碁への効果が期待できる。

2 悪手を判別するNNの構成

本稿では、悪手を判別するための入力層81ノード、出力層81ノードのNNの作成をする。入出力層はそれぞれの碁盤の9×9座標81個に対応している。図1に示すように碁盤の隅から順に番号を割り振った。NNの入力層 (IN) と出力層 (OUT) は以下のように表現した。

$$IN_i = \begin{cases} 0 & \text{(石なし)} \\ 1 & \text{(黒石)} \\ 2 & \text{(白石)} \end{cases} \quad OUT_i = \begin{cases} 0 & \text{(良手)} \\ 1 & \text{(悪手)} \end{cases} \quad (0 \leq i \leq 80) \quad (1)$$

隠れ層はノード数や層の数によって表現力が違うため、それぞれ変更して実験を行い決定することにした。

また、本稿では教師データとしてインターネット対局場 KGS [3] の棋譜を元に教師データを作成した。ここで、棋譜で着手された良手、それ以外の座標は悪手として座標は (1) 式のように教師データを作成する。本稿では、36,754 対局 (1,778,191 局面) を教師データとして用いた。教師データのうち8割を訓練用データ、2割を評価用のデータとして使用した。学習した後、NNは出力ノードで0以上1以下の実数値を出力する。このとき、出力が1に近い

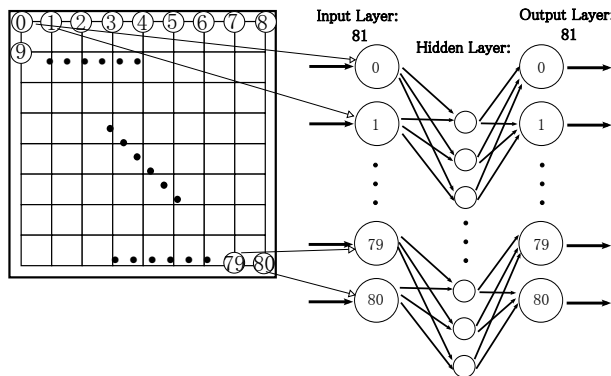


図1: NNと碁盤の対応。入力、出力ともに碁盤の座標に対応する。

ほど悪手、0に近いほど良手であると解釈する。この出力から悪手がどこにあるかを判定する。

3 NNの性能の評価

本稿では、NNによって悪手の判別する。NNで悪手の判別が可能であることを確認するためにNNの性能の評価を行う。ここで構成したNNはモンテカルロ囲碁への応用が目的である。そのため、木探索中でどこまでが悪手として枝刈りできるかという点について評価する。すなわち、悪手が何手まで正確に判別できているかを評価する。そこで、棋譜の着手 (良手) をNNで悪手と判別する場合を誤りとして、本稿のNNがどの程度の誤判別をするか確かめる。ここでは、NNに対して評価用データ (2節参照) を入力し、評価用データ中の何%誤判定するか誤り率を検証した。その結果を図2で示す。この図2を見ると30手程度の枝刈りをした場合に約10%の誤り率があることがわかる。実際には32手までで誤り率10%以下となる。

4 モンテカルロ囲碁へ適用と結果

前節で述べた悪手を判別するNNをモンテカルロ囲碁に応用した場合の効果について検証する。効果の検証は悪手を考慮したモンテカルロ囲碁と考慮しないモンテカルロ囲碁で実際にゲームを繰り返して行い、考慮したモンテカルロ囲碁の勝率からどの程度強いプログラムとなったかを検証した。モンテカルロ囲碁への応用では (1) モンテカルロ木探索でNNが判定する悪手を枝刈りする。 (2) プレイアウトでNNが判定する悪手を枝刈りする。という応用が考えられる。いずれもNNの出力が悪手の箇所を枝刈りする。NNの出力が悪手に近い順にx手を枝刈りしたときの効果と比較して評価する。盤面上のそれぞれの点をNNの出力値順にソートした列を  $p_0, p_1, \dots, p_{80}$  として、それぞれの出力値を  $v(p_i)$  と書けば、

$$v(p_0) \leq v(p_1) \leq \dots \leq v(p_x) \leq \dots \leq v(p_{80})$$

On Applying a Neural Network for Monte Carlo Go  
 † Shinya Sato, Division of Information Engineering, School of Science and Technology, Meijo University  
 ‡ Osami Yamamoto, Department of Information Engineering, Faculty of Science and Technology, Meijo University

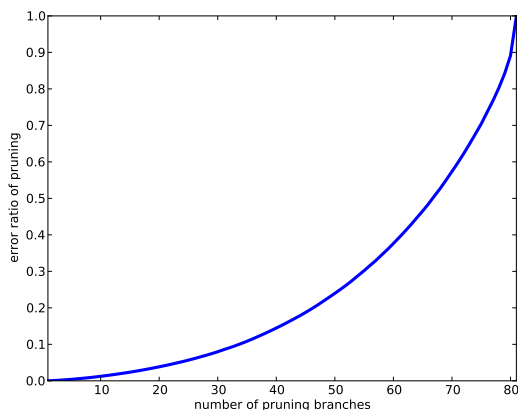


図 2: 枝刈りの誤り率. 横軸は NN の出力によって枝刈りを行う個数, 縦軸は枝刈りの誤り率である. ここで枝刈りの誤りは枝刈りの候補と棋譜の着手が一致した場合を誤りとしている.

となっている. このとき,  $p_0, \dots, p_{x-1}$  が良手,  $p_x, \dots, p_{80}$  が悪手と定義する.

実験ではプログラムがどの程度強くなったかを評価する. そのため, 枝刈り前と枝刈り後のプログラムで繰り返し対局を行った. 本稿では黒番を枝刈り適用後のプログラム, 白番を枝刈り適用前のプログラムとし, 白番には 6.5 目のコミを付与する. コミは白が不利という囲碁の前提で設けられているルールで白は与えられた分だけゲーム終了時に陣地を多く数えることができるルールである. モンテカルロ木探索のパラメータの設定はシミュレーション回数を 10,000 回とし, ノードを展開するタイミングはシミュレーションを 10 回以上行ったときとした. 実験の対戦回数は 400 から 500 ゲームとした. 以下に 2 つの実験結果を示す.

(1) モンテカルロ木探索での枝刈りの実験. モンテカルロ木探索そのものに対して枝刈りを行う. 図 3 にて枝刈りの個数を変化させたときの勝率の変化を示す. このグラフを見ると, 注意として白と黒のもとの勝率が同じでないという点がある. つまり, 勝率の基準が 0.5 ではない. 実際に枝刈りなしのプログラム同士でゲームを行った際, 黒の勝率は 0.37 程度であった. そのため, この値を基準に勝率を確認する. グラフから得られる結果として, 枝刈り前の黒の勝率から見て枝刈り後の勝率は 0.37 から 0.61 まで最大 0.24 程度向上させることができた.

(2) プレイアウトでの枝刈りの実験. モンテカルロ木探索で使用するシミュレーション, プレイアウトに対して枝刈りを行う. 枝刈りは NN が悪手である手に対して行うことにする. プレイアウトはかならず終局状態に達する必要があるため, 着手点の少ない局面では枝刈りしすぎないように工夫が必要である. そのため, プレイアウトでの枝刈りの際は盤面上の空点 (石が置かれていない点) の個数に対して一定の割合だけ枝刈りを行う. 実験としてプレイアウトでの枝刈りを空点の個数の 60% に対して行った. その結果, 枝刈り前の黒の勝率 0.37 から 0.40 へ変化した. これはモンテカルロ木探索への枝刈りと比較して勝率の向上が大きくないが, わずかに効果が確認できた.

さらに, 他のプログラムとゲームを行う検証を行った. ここで用意した囲碁プログラムはモンテカルロ囲碁を用いない gnugo を使用した. その対戦成績を表 1 で示す. 結果として, 枝刈りによる勝率の向上はあまり見られなかった. 原因として今回自作したプログラムと gnugo の間に実力差がありすぎることや, そもそも枝刈りの質が良くないことが考えられる.

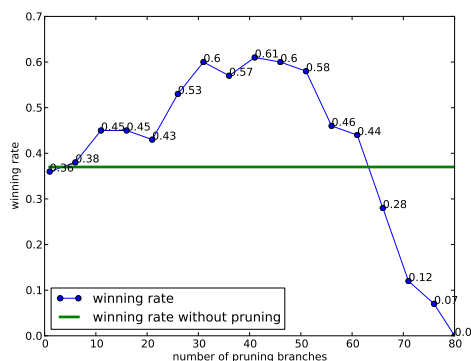


図 3: 枝刈り前と枝刈り後の対局における枝刈り後の勝率の変化. 0.37 で引かれた線は枝刈りなしの黒の勝率 (基準線).

表 1: gnugo との対戦成績. 枝刈りする深さを変化させる実験.

枝刈りの深さ	枝刈り後の勝率
枝刈りなし	0.039
root から	0.004
深さ 1 から	0.028
深さ 2 から	0.034
深さ 3 から	0.032

## 5 まとめと今後の課題

本稿では悪手を識別するための NN を作成し, これをモンテカルロ囲碁へ木の枝刈りという形で適用した. 作成した NN の評価を行ったところ, 理想的な出力には及ばないものの近い形の NN を作成することができた. NN による悪手の枝刈りの実験は (1) モンテカルロ木探索での枝刈り, (2) プレイアウトでの枝刈りの 2 つについて行った. その効果は枝刈りの適用前と適用後のプログラムで対局を行い, その勝率がどれだけ向上したかを確認した. (1) では枝刈りを NN の出力が悪手に近い順に 41 手行ったとき, 最大で勝率を 0.24 向上させることができた. また (2) では空点 (石が置かれていない点) のうち NN の出力が悪手に近い順に 60% を枝刈りし, 勝率が 0.03 向上することを確認した.

今後の課題としては学習方法についての工夫, 局面に応じた NN の適用が考えられる. まず, 学習方法については NeuroGo の NN の構成を取り入れるということが考えられる. NeuroGo では, 石の連結成分や目の数などを考慮して NN を構築している. 本稿で用いた NN でもこの様な工夫によってより精度の良い悪手の判別が行える可能性がある. 次に, 枝刈り時の課題では局面に応じて NN を用意することが考えられる. 囲碁において序盤や終盤のような明らかに性質の異なる局面では専用で別の NN を用意することで状況に合った柔軟な悪手の判別が可能になることが考えられる.

### 参考文献

[1] Bruggmann, B.: Monte Carlo Go. Technical report. Physics Department, Syracuse University, 1993.

[2] Enzenberger, M.: Evaluation in Go by a neural network using soft segmentation. *10th Advances in Computer Games Conference, Graz*, pp. 97-108, 2003.

[3] The KGS Go Server. <https://www.gokgs.com/>