1F2-5

囲碁プログラムへのニューラルネットワークの応用について

On Application of Nueral Network to Go

佐藤慎也 *1　　山本修身 *2
Shinya Sato　　Osami Yamamoto

*1名城大学大学院理工学研究科情報工学専攻
Division of Information Engineering, School of Science and Technology, Meijo University

*2名城大学理工学部情報工学科
Department of Information Engineering, Faculty of Science and Technology, Meijo University

In this paper, we describe the construction of a classifier of good and bad moves in the Go game by deep learning. The classifier inputs the positions of the stones of the both players and the positions on the board to be classified, and it outputs the position of bad moves as the next move. The total classification rate of our classifier was 70.8 %. The classifier output “good move” for 80.6 % of “real good move.” The classifier will be applied to refinement of the MCTS algorithms and evaluation of moves in Go games.

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．2015 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．現在最も実力のあるこのプログラムは 19×19 においてアマチュアの 6 段程度，9×9 においてはプロに近い実力があると考えられている．

強い囲碁プログラムの構築においての課題は，広い探索空間でいかに効率良く探索を行うかという点と，いかに途中の局面をより正確に評価するかの 2 点である．本稿では，機械学習のひとつである深層学習 (Deep Learning: DL) [2] を用いて，途中の局面を評価するための手法を提案する．本稿では，9×9 の碁盤において，ある座標が良手か悪手かの 2 クラスを識別する．すなわち，識別後にある座標が良い手か，悪い手かを出力するような識別機を考える．

良手と悪手の 2 クラスの分類に関して，Neural Network(NN) を用いて試みた研究 [3] がある．[3] では，9×9 碁盤の情報を入力とし，実数値で悪手に近いか良手に近いかを出力する．学習時の教師データは棋譜の手を良手，それ以外とすべて悪手として定義して学習を行った．81 個の実数値の出力のうち，値が小さい方から 30 個を悪手のクラス，その他を良手のクラスとしたとき，真の良手の約 90%が良手のクラスに含まれた．本稿では，[3] との性能比較を行うことで DL を用いた手法の優位性を示す．ゲームの木探索においては，悪い局面の探索をいかに枝狩りして探索を効率化するかということが重要である．本稿の狙いは，悪手の識別によって木探索を効率的に行うことである．

DL を用いた囲碁プログラムの研究として，Clark らによる [4] がある．[4] では入力を 19×19 の碁盤の情報 (配石など) とし，棋譜の手に最も近い手を出力とする．[4] では accuracy が約 44%の学習を実現しており，棋譜の手を高い割合で再現することに成功している．本稿と [4] の違いは，扱う碁盤のサイズが 9×9 のものを扱うこと，および出力において良手 (棋譜に近い手) を 1 つに限定しない点である．棋譜に近い手すべてを良手とする．

連絡先: 佐藤慎也，名城大学大学院理工学研究科情報工学専攻，
E-mail:143430012@ccmailg.meijo-u.ac.jp

2. DL について

DL は教師なしニューラルネットワーク (NN) による特徴抽出 (教師なし学習) と識別機 (教師あり学習) を用いた分類器である．DL の特徴は，事前学習でデータの特徴を自動的に蓄えることで特徴抽出が容易になる点である．従来の NN の学習では，教師データを前処理することで故意にデータの特徴を与える必要があったが，DL では特徴抽出を自動化することができる．

事前学習 (Pre-training) では Autoencoder(AE) [5] や Restricted Boltzmann Machine(RBM) などを用いて特徴を抽出する．今回は AE を用いてデータの特徴抽出を行った．AE は入力層，隠れ層，出力層の 3 層ニューラルネットワークで構成される．AE では，入力層と出力層が同じになるように隠れ層までの重みを学習する教師なしの学習を行う．これを実現することで入力層をより少ない層数の隠れ層で表現できるため次元圧縮を可能にしており，データの特徴を抽出することができると考えられている．事後学習 (Fine-training) では，Pre-training で次元圧縮されたデータ特徴を用いて各クラスに分類を行う．これには Multi Layer Perceptron(MLP) や SVM が使われる．本稿では事前学習に AE，事後学習には MLP を用いて識別を行った．また，DL のライブラリとして pylearn2*1 を用いた．pylearn2 は，モントリオール大学で開発されている DL を実装したライブラリである．また，pylearn2 は数値計算ライブラリ theano がベースになっている．theano は微分計算や GPU の計算コードを出力ができることが優れた特徴である．そのため，GPU に対応したソースコードを出力することが可能であり，学習時に GPU を使用することで学習の効率を向上することができる．CPU で学習することも可能

∗1　http://deeplearning.net/software/pylearn2/

だが，並列力した GPU の学習効率と比べると学習の収束の早さが大きく異なる．

3. データの表現方法

本稿では，9 × 9 サイズの囲碁における良手と悪手の識別を行う．学習のために，過去の対局記録 (棋譜) を用いて入力データと教師データを作成した．

3.1 入力データについて

入力データは碁盤の状態および悪手と良手の判別の対象を入力する．ここで，9 × 9 の碁盤を表現するために 1 座標に対応するようなサイズ 81 の表現を考える．碁盤の配石はこの碁盤表現を 2 つ用いて表現した．1 つは自分の石の配置，もう 1 つは相手の石の配石である．それぞれ石がある箇所については 1，それ以外については 0 として 9 × 9 の 81 入力で表現した．加えて，判別の対象となる座標については，81 座標の対象となる座標のみ 1 それ以外を 0 とするように表現した．上記の合計 243 を入力として学習を行った (図 1 を参照)．

入力データの碁盤の表現については，DL による囲碁の先行研究 [4] を参考にした．[4] と異なる点は，判別の対象とする座標を入力で与える点である．本稿では，ある点が良手か悪手かを判別するため，1 座標単位で識別する方がより厳密なデータ表現であると考えた．[4] では，碁盤の情報から最も棋譜の手に近い手を識別することが目的であるため，判別の対象となる座標は必要ないと考えられる．

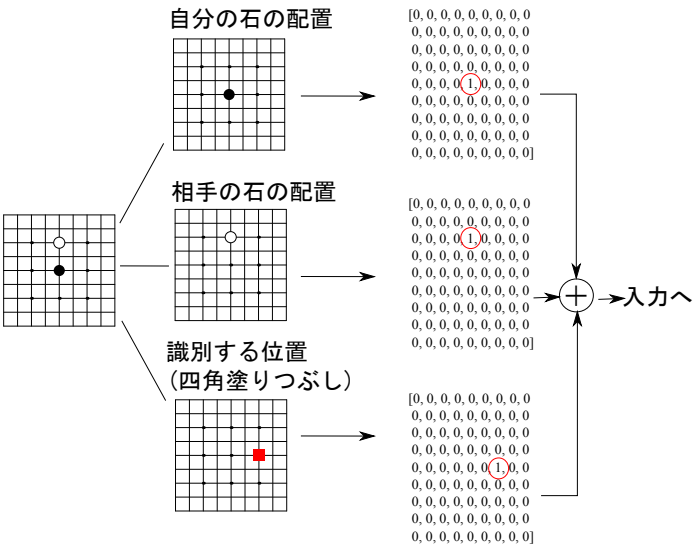

図 1: 入力データを作成するイメージ図．碁盤の 81 ノードを 3 つで計 243 ノードで入力を表現する．一番左側の図は判定したい盤面，右の線で延びる 3 つの碁盤は上から自分の石の配置，相手の石の配置，悪手 (良手) の判定をする位置をそれぞれ示している．これら 3 つの碁盤を 0 と 1 で表現し 81 × 3 で 243 の入力を作成する．

3.2 教師データについて

教師データは過去のゲームの記録として棋譜を用いた．使用した棋譜データは cgos[*2] で行われた試合記録を用いた．その中でも，比較的高いプログラムの対局データを用いた．良手と悪手の判別のために，棋譜から良手と悪手のそれぞれを教師データとして用意する．ここで前提として，棋譜で選ばれた手は悪手ではないという仮定を行う．これは，比較的強いプログラムが悪手を選択する確率は低いと考えられるためである．今回用いた教師データは，656559 局面 (11858 ゲーム) から棋譜の手の座標を良手，それ以外の座標からランダムに 1 つ選択したものを悪手として，1 局面につき 1 つづつ教師データを作成した．すなわち，良手と悪手合わせて計 1313118 個の教師データを訓練用データとして用いた．テストデータとして，上記の教師データとは別の棋譜 (1317 ゲーム) から良手と悪手のデータを計 166612 個用意した．

*2 囲碁プログラムを通信で対戦するために設けられたサーバ．http://cgos.boardspace.net/

4. 学習手法とその設定について

本稿では，良手と悪手の判別を行う際に Pre-training として AE，Fine-training として MLP を用いた．まず，Pre-training では AE によるデータの特徴抽出を行う．本稿では AE の層を積み重ねて段階的に特徴を抽出する Stacked Denoising Autoencoder(SDA) [6] を用いた．SDA では層ごとに特徴を抽出することができるため，層の数を増やすことでより表現力を持たせることができる．ここでは，3 つの設定値で学習を行い層の数を 3，4，5 に設定した．入力 243 ノードに対し，経験的に各層の隠れ層のノード数は 1 層目は 162，2 層目は 81，3 層目は 41，4 層目は 21，5 層目は 10 とした．3，4 層の SDA では 3，4 層目までを使用する．また，学習率は 0.001，バッチサイズは 128 とした．学習の epoch 数は 3，4 層の識別器は最大 10 に設定し，5 層の識別器は最大 100 とした．次に，Fine-training では MLP によって，SDA によって次元圧縮された特徴量を用いてクラス分離するためのパラメータを学習する，このとき，MLP のパラメータとして学習率を 0.05，バッチサイズは 128 として学習を行った．epoch 数は 3，4 層の識別器で最大 100，5 層の識別器で最大 500 として学習を行った．このとき，使用した計算機として CPU は Intel(R) Xeon(R) CPU E5-1620 v3 @ 3.50GHz 16GB Memory ，GPU は NVIDIA Tesla C2075 (6GB, 448 CUDA cores) を使用し，学習時には GPU を用いた．

5. 良手と悪手の識別実験の結果

良手と悪手を判別する識別機の学習を DL を用いて行った結果を示す．テストデータによる識別結果を表 1 から 3 に示す．まず，表 1 の SDA が 3 層の場合の識別結果は全体の accuracy (正解率) は 68.3%，真の良手を良手と識別する割合は 80.5%，真の悪手を悪手と識別する割合は 56.1%であった．次に，表 2 の SDA が 4 層の場合の識別結果は全体の accuracy (正解率) は 70.8% 真の良手を良手と識別する割合は 80.6%，真の悪手を悪手と識別する割合は 61.0%であった．最後に，表 3 の SDA が 5 層の場合の識別結果は全体の accuracy (正解率) は 82.6% 真の良手を良手と識別する割合は 74.3%，真の悪手を悪手と識別する割合は 90.9%であった．accuracy の計算式としては以下を用いた．それぞれ，真の良手を良手と判別するものを TP，真の良手を悪手と判別するものを FN，真の悪手を良手と判別するものを FP，真の悪手を悪手と判別するものを TN とすると：

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

表 1: SDA3 層と MLP によるテストデータの識別結果．括弧外の表中の数字は対応するデータの数，括弧中の数字は真の良手 (悪手) 中の何%が該当するかを示す．

	真の良手	真の悪手
識別結果が良手	67,140(80.5%)	36,526(43.9%)
識別結果が悪手	16,166(19.5%)	46,780(56.1%)

表 2: SDA4 層と MLP によるテストデータの識別結果．括弧外の表中の数字は対応するデータの数，括弧中の数字は真の良手 (悪手) 中の何%が該当するかを示す．

	真の良手	真の悪手
識別結果が良手	67,182(80.6%)	32,483(39.0%)
識別結果が悪手	16,124(19.4%)	50,823(61.0%)

と表される．全体の accuracy で比較すると 5 層のものが最も高い値が出ているが，真の良手を良手と判別する割合を比較すると 5 層が最も低い値となっている．このように学習の設定によって大きく異なる結果が得られるため，この調整を試行錯誤する必要がある．ここでは，特に 4 層の SDA の識別結果をある局面で確認する．識別結果は図 2 のようになっている．様々な局面で識別の結果を見ると未知の局面，特にゲーム中盤から終盤にかけて比較的大きな範囲を良手と判別するような識別機であることがわかる．

[3] の NN による良手と悪手を判別する識別器と比較すると，[3] では一定の個数のみ悪手を定義することが可能だったが，悪手の多い場面では多く，少ない場面では少なく悪手を識別できていることがわかる．また，正解率を見ても良手を良手と識別する割合が 74%から 80%と高い値を示している．[3] で同様の識別率の場合には 45 から 50 手しか悪手を定義できない．この事実から，序盤のような選択できる手が多いときに DL を用いた識別器は優位性があることがわかる．

また，DL を用いた識別機との比較として [4] と比較してみると，用途や目的が違うため一概にはどちらが良いと言えないが，[4] では悪手の識別までは行えていないことや複数の良手の可能性を考慮できる点では本稿の識別器の方が機能として優れている．しかし，[4] の accuracy を見ると棋譜の手を約 44%再現することができるため，最も棋譜の手に近い手を発見するという点で [4] は大きく優位性があることがわかる．

6. まとめと今後の課題

本稿では，棋譜から DL を用いて良手と悪手を識別する分類機を作成した．その結果，識別機の性能がテストデータにおいて良手と悪手のそれぞれを約 70% から 80%正しく識別することを確認した．良手を良手と正しく判別する割合は約 74%から 80%であり，おおむね正しい分類ができていることが確認できた．実際にある局面における識別結果をみると，教師あり NN を用いた悪手の識別 [3] では行えなかった同じ局面でも黒石と白石の場合それぞれで識別結果を適宜変更することや，特定の座標の識別が可能になっているため機能としては優位性があることを確認した．

今後の課題として，この識別機を用いて枝狩りや [3] との性能の比較することが考えられる．[3] では枝刈りによって勝率が向上したことを確認したので，さらに勝率が向上することが期待できる．また，性能を向上するために入力データを工夫することが考えられる．囲碁においては，棋譜の手以外がすべて悪手である保証はないので，この教師データの悪手は事実に即していない．これを改善するためには，悪手についても正しい情報を持った教師データを使って学習する他ない．囲碁における学習データは一般的に棋譜のみが使われているが，これ以外の局面を段階的に評価した教師データを用いて学習するようなことが考えられる．さらに，本稿では 9×9 の碁盤において DL を用いた良手と悪手の識別が有効であることを示したが，19×19 の碁盤において同様の手法が有効であるか否かを検証することが考えられる．19×19 の碁盤において考えられる今回の手法の問題として，データが単純に大きくなることよって学習時間が長くなること，9×9 よりも類似局面が少ないことによる学習が収束しない問題などが考えられる．最後に，本稿でも 3 つの学習時の設定の結果を示したが，どの設定を用いるのが最も良いかということも試行錯誤する必要がある．

表 3: SDA5 層と MLP によるテストデータの識別結果．括弧外の表中の数字は対応するデータの数，括弧中の数字は真の良手 (悪手) 中の何%が該当するかを示す．

	真の良手	真の悪手
識別結果が良手	61,949(74.3%)	7,575(9.1%)
識別結果が悪手	21,357(25.7%)	75,731(90.9%)

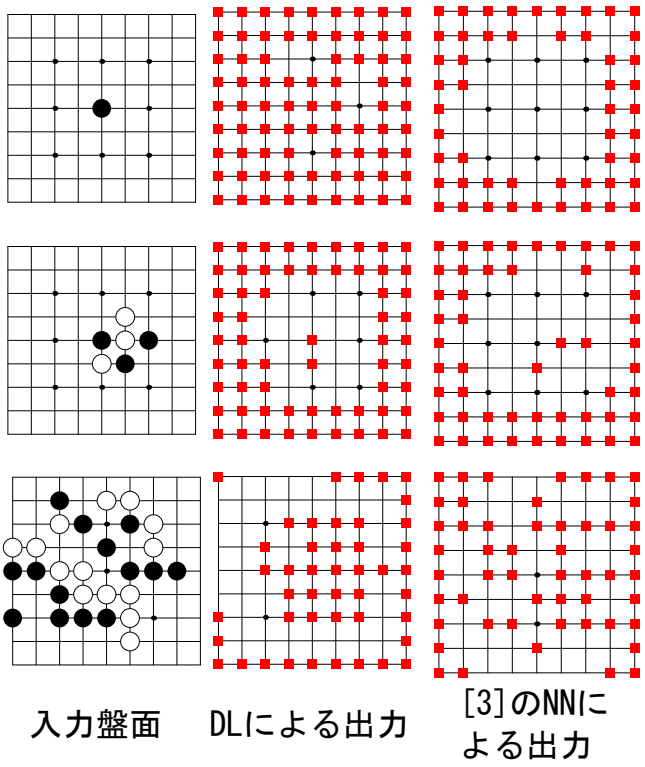

図 2: 3 つの局面における識別結果．左側の図は入力する局面，中央の図は DL による識別器において識別結果が悪手の位置を色で塗りつぶした図，右側の図は [3] による NN を用いた識別器において，50 個の悪手を色で塗りつぶした図である．

参考文献

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

Available from http://www.ideanest.com/vegos/MonteCarloGo.pdf

[2] Y.Bengio: Learning deep architectures for AI. Foundations and trends®in Machine Learning (2009).

Available from http://www.iro.umontreal.ca/~bengioy/papers/ftml_book.pdf

[3] 佐藤慎也: 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．情報処理学会 全国大会，3P-05 (2015).

[4] C.Clark and A.Storkey: Teaching deep convolutional neural networks to play Go. Cornell University Library，arXiv:1412.3409v1 (2014).

Available from http://arxiv.org/pdf/1412.3409v1.pdf

[5] G.E.Hinton and R.R.Salakhutdinov: Reducing the dimensionality of data with neural networks. *Science* 313 (5786) 504-507 (2006).

Available from http://www.cs.toronto.edu/~hinton/science.pdf

[6] P.Vincent: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Reserch* 11 3317-3408 (2010).

Available from http://jmlr.csail.mit.edu/papers/volume11/vincent10a/vincent10a.pdf