

囲碁ニューラルネットワークの判断根拠の可視化

万代悠作^{1,3,a)} 金子知適^{2,4}

概要: 情報科学の発展と計算機能力の向上により、囲碁や将棋においてもコンピュータプレイヤーが人間のトッププレイヤーの実力を超えた。特に囲碁においては、深層ニューラルネットワークを用いた大規模な強化学習により正確な評価関数を作成することが可能となったことがコンピュータプレイヤーの棋力向上につながった。またボードゲームだけでなく、深層ニューラルネットワークを用いた意思決定は現実社会の多くの場面に広がりつつある。しかしながら現実問題の重要な意思決定を行う場合には、深層ニューラルネットワークが下した判断を人間が評価し、その妥当性を検証できる必要がある。また人間の熟練者を超えるような性能をもつ判断器について、その意思決定過程を可視化できれば人間が学習する手助けになると期待できる。本研究では画像分類器において広く適用されている Saliency Map と SmoothGrad について、囲碁の方策予測と局面価値予測に対し適用し、その出力結果の妥当性を検証した。またモンテカルロ木探索を用いた局面価値予測に対し、木探索を考慮しつつ Saliency Map を計算する手法を提案し、その有用性を評価した。実験ではそれぞれの手法が予測結果に大きく影響を与える座標を検出できることを示した。

Visualization of Neural Network Inference in the Game of Go

YUSAKU MANDAI^{1,3,a)} TOMOYUKI KANEKO^{2,4}

Abstract: Thanks to the recent increase of computational resources and improvements in the computer science, several computer players defeated human top level players in the game of Go and Shogi. Particularly, it has been proven that good evaluation functions can be created with deep neural networks for these games. Deep neural networks are widely used not only in board games but also in many areas including where decisions should be made with high reliability. In such areas it is required that decisions made by a deep neural network can be explained and interpretable. In addition to the reliability purpose, it would be useful to visualize decision process of superhuman deep neural networks for teaching beginners. In this work we investigate performance of Saliency Map and SmoothGrad in the game of Go. These two methods are for visualizing “explanations” of deep neural networks for image classification. We also show that the Saliency Map method can be extended to decisions enhanced by Monte Carlo tree search by considering the process of the search method. Experimental results showed that the two explaining methods are also effective in the game of Go and our proposed explaining method can successfully detect coordinates of the game which are particularly influential to final decisions.

1. はじめに

機械学習の様々な技術的発展と計算機の性能向上に伴い、特定の領域においては人工知能を用いたシステムが人間の熟練者を超える段階となった。そのような領域の例の筆頭として、チェス・囲碁・将棋に代表されるボードゲームが挙げられる。特に囲碁では、以前は困難とされていた局面評価関数の作成が深層ニューラルネットワーク (Deep Neural Network, DNN) を用いることにより成功

¹ 東京大学大学院総合文化研究科
Graduate School of Arts and Sciences, The University of Tokyo

² 東京大学大学院情報学環
Interfaculty Initiative in Information Studies, The University of Tokyo

³ 日本学術振興会特別研究員
JSPS Research Fellow

⁴ 国立研究開発法人科学技術振興機構さきがけ
JST, PRESTO

a) mandai@graco.c.u-tokyo.ac.jp

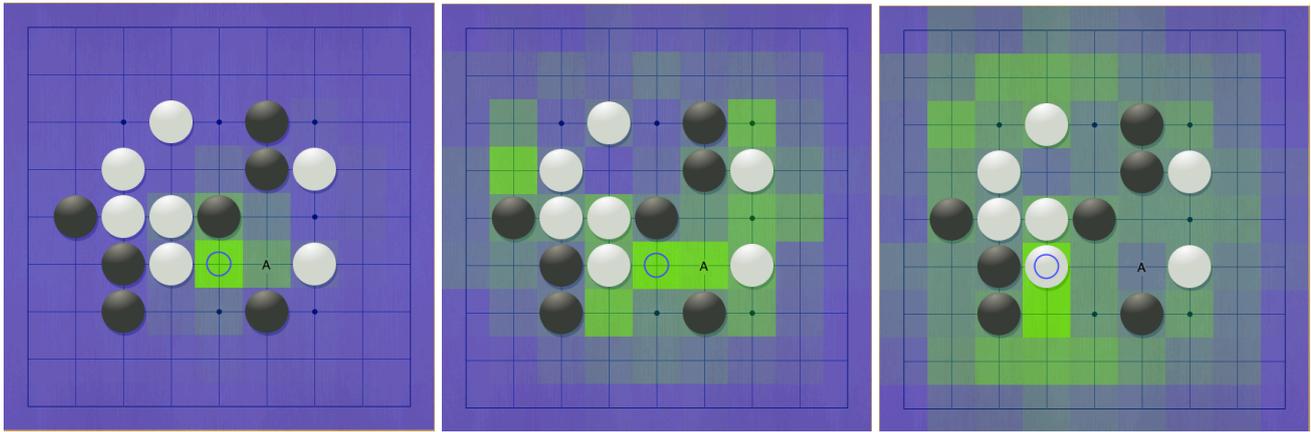


図 1 本稿で述べる手法によって得られた Saliency Map. 左から Poilcy, Value, MCTS による評価値に対する出力

Fig. 1 Examples of Saliency Maps obtained by the methods described in this article. Saliency Map for Policy (left), for Value (middle), and for MCTS evaluation (right).

し、膨大な計算資源を投入して学習を行った DNN を用いる AlphaGo Zero が人間のトッププロを超える実力を持つと報告されている [1].

これらのような人間を超える性能を持つ人工知能システムは今後さらに増えていき、現実社会がそれらに依存する割合は非常に高くなると予想される。しかし、人間を超えた人工知能システムによる判断の信頼性を検証することは困難であり、研究途上の段階であるといえる。人工知能システムの判断の検証の重要な前段階として、その判断の根拠を提示して人間にとって理解しやすい形で可視化することが挙げられる。そのような可視化手法は人間の初心者の練習を手助けするようなシステムなど、幅広い応用先を持つと思われる。

特に画像認識を行う DNN の判断根拠の可視化の研究として入力に対する微分値をもとにした手法が提案されている。これは DNN の出力値を入力について微分し、その微分係数が高い部分を判断の根拠と解釈するものである。入力が画像であれば、出力に関連すると判断された画素の集合が得られる。出力結果に大きな影響を持つ入力の一部分はその予測器が下した決定の判断根拠と解釈することも可能であり、このような手法は広く研究されている。

本研究では囲碁に対して訓練された DNN を対象に可視化手法を適用し、囲碁におけるそれらの有効性について議論を行う。特に Saliency Map [2] と SmoothGrad [3] と呼ばれる手法に着目し、局面の着手判断や形勢判断の根拠が可視化できるかどうかを検証する。具体的には AlphaGo Zero [1] で用いられた方策予測と局面価値予測を行う DNN について可視化手法を適用したときに予測の根拠となった囲碁の盤面上の石を抽出できるかどうかについて実験を行った。また探索アルゴリズムを用いて得られる局面予測について、探索を考慮した判断根拠の可視化手法を提案し、

実験的に評価を行った。実験結果から、評価したアルゴリズムは囲碁の DNN に対しても有効に働くことがわかった。

図 1 に本稿で述べるアルゴリズムをある局面に対して適用したときの Saliency Map を示す。得られた Saliency Map は 4.1 章で説明する方法によって 2 次元画像に変換して 0 から 1 に正規化したのち、0 を青、1 を緑とした場合の画像を表している。

2. 関連研究

2.1 コンピュータ囲碁における深層学習

ボードゲームは人工知能研究の黎明期よりその研究対象となっており、なかでも囲碁はチェスとならび世界中の研究者たちによって広く研究されている。ニューラルネットワークを囲碁の評価関数に用いるという研究は黎明期より存在しており [4]、近年では計算機の能力向上と理論的進歩の成果として、ニューラルネットワークを用いたコンピュータプレイヤーが良い成績を収めている。特筆すべき例として、深層畳込みニューラルネットワーク用いた AlphaGo [5] とその後継の AlphaGo Zero [1] が存在する。AlphaGo の DNN は人間の棋譜から学習を行ったのち、強化学習によってより精度を向上させている。AlphaGo では学習した DNN とモンテカルロ木探索 (Monte Carlo Tree Search, MCTS) を組合せ、人間のトッププレイヤーである Lee Sedol 氏に勝ち越した。AlphaGo Zero では方策を予測する policy network と局面価値を予測する value network の両方をもつ DNN を用いており、この DNN を自己対戦を用いた強化学習により訓練した結果、人間の知識を使うことなく AlphaGo の棋力を超えた。

また AlphaGo Zero と同様の手法で学習を行った ElfOpenGo [6] と呼ばれる囲碁プログラムも公開されている。ElfOpenGo で用いられている DNN への入力

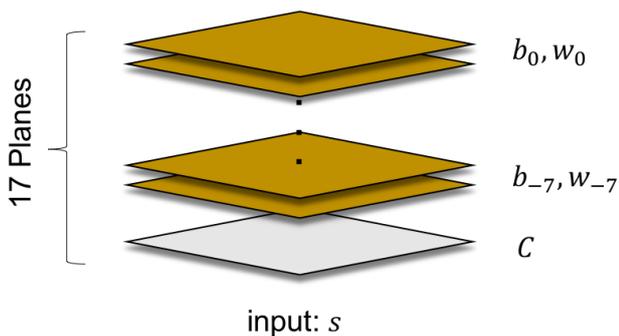


図 2 囲碁において用いられる入力特徴の模式図
Fig. 2 Illustration of the input for DNNs in Go

AlphaGo Zero と同様、人間の知識は用いられておらず、黒石・白石の配置とその履歴、現在の手番を符号化した $17 \times 19 \times 19$ の 3 次元配列となっている。図 2 ではある囲碁の盤面を、7 手前から現在の石の配置を表すプレーン ($b_0, w_0, \dots, b_{-7}, w_{-7}$ の 16 層) と、現在の手番を表すプレーン (1 層) を表している。手番を表すプレーンは、現在の手番が黒であればすべてが 1, 白手番であれば 0 になる。AlphaGo Zero においては自分の石・相手の石の配置とその履歴、現在の手番という符号化を行っている。

AlphaGo Zero が用いている MCTS は DNN の評価によって探索と局面の評価を行うもので、文献 [7], [8] で用いられているような乱数によるシミュレーションによって評価を行うものとは異なる。MCTS における選択の際には PUCT アルゴリズム [9] の変種を用い、局面の policy network の値とこれまでの探索によって得られた局面価値をもとに評価値を計算する。探索木末端までたどり着いたら、value network によって末端局面が評価され、根局面までたどってきた局面の評価値を先程の value network の出力値によって更新する。

2.2 Saliency Map/SmoothGrad

DNN を用いた研究は囲碁などのボードゲームの他にも様々な領域で行われており、すでに実社会の多くの場面で応用されている。しかしながら DNN のような複雑な構造を持つアルゴリズムによって得られた推論結果の理由付けや説明などは現時点での課題であるとされている [10]。

推論結果の説明に関連する研究に、得られた推論結果に寄与した入力の一部を求める研究がある。推論結果に寄与した入力の一部を saliency (顕著性) と呼び、様々なものが提案されている。予測結果に寄与した入力部分を観察することにより、DNN などの予測器の妥当性をある程度検証する事が可能になる。

文献 [2] では予測結果を入力について微分し、その微分係数の大きさを saliency を定義している。例としてクラス分類を行う予測器を考える。この予測器は入力 x_0 が与えられたとき、あるクラス c に対して評価 $S_c(x_0)$ を割り当

てる。この評価を微分し、 x_0 における微分係数 w を、判定結果に対する x_0 の寄与とする。 w の次元は入力の次元と等しく、 x_0 のある要素 $x_0^{(i)}$ の寄与は $w^{(i)}$ に対応する。本稿ではこの手法によって得られた微分係数を Saliency Map と定義する。Saliency Map は手法が簡便なほか、一次近似とみなせば自然な解釈が可能という利点がある。文献 [2] では画像分類器に対して Saliency Map を適用している。画像分類の場合、入力は RGB 画像であれば $3 \times w \times h$ の 3 次元配列となり、Saliency Map の出力結果も同じ次元となる。ここで w, h は画像の縦横ピクセルの数である。得られた Saliency Map を画素ごとの影響度に変換するため、ある画素に対応する 3 つの Saliency Map の出力結果のうち絶対値が最大のものをその画素の影響度としている。

Saliency Map を改良した手法として SmoothGrad が提案されている [11]。Saliency Map の出力に生じる雑音を除去するために、SmoothGrad では多数の類似画像の平均を取ることで除去できるとしている。類似画像は元の画像 x_0 に正規乱数 $\mathcal{N}(0, \sigma^2)$ をそれぞれの要素に加算することで生成し、生成した n 枚の画像について Saliency Map の出力をそれぞれ得たあと平均をとることで SmoothGrad の出力としている。

3. MCTS の価値予測結果に対する Saliency Map

MCTS によって根局面 s_0 の評価値の予測 $\hat{v}(s_0)$ を計算することができる。ここで $\hat{v}(s_0)$ はもとの局面価値予測 $v(s_0)$ よりも正確であると期待される。この改善された価値予測に対して Saliency Map を適用する手法を提案する。

簡単のため、AlphaGo Zero で採用されたような局面の展開の閾値が 1 の MCTS をプレイアウト回数 n 行ったとする。このとき改善された局面評価値は

$$\hat{v}(s_0) = \frac{1}{|T|} \sum_{s \in T} v(s) \quad (1)$$

となる。ここで T は探索後の探索木に存在する局面の集合で、 $|T| = n + 1$ である。この \hat{v} を s_0 について偏微分すると Saliency Map が得られる。式 (1) 右辺を s_0 で表すことを考える。探索木中の s_0 から到達可能なある局面 s は、囲碁などの決定的なゲームの場合状態遷移関数 \mathcal{T} を用いて $s = \mathcal{T}(s_0, A_s)$ と表すことができる。ここで A_s は s_0 から s まで遷移するために必要な着手の列である。 $\mathcal{T}(s_0, A_s) = \mathcal{T}_s(s_0)$ と略記することにとると、

$$\hat{v}(s_0) = \frac{1}{|T|} \sum_{s \in T} v(\mathcal{T}_s(s_0))$$

$$\frac{\partial}{\partial s_0} \hat{v}(s_0) = \frac{1}{|T|} \sum_{s \in T} \frac{\partial}{\partial \mathcal{T}_s} v(\mathcal{T}_s(s_0)) \frac{\partial}{\partial s_0} \mathcal{T}_s(s_0) \quad (2)$$

と変形される。 $\frac{\partial}{\partial \mathcal{T}_s} v(\mathcal{T}_s(s_0)) = \frac{\partial}{\partial s} v(s)$ であり、これは s における Saliency Map と同様である。

式 (2) を一般の場合について求めることは容易ではないため、囲碁において計算できる状況を考える。局面 s が図 2 のように石の配置の履歴と手番で構成されているとき、つまり $s = \{\mathbf{b}_0, \mathbf{w}_0, \dots, \mathbf{b}_{-7}, \mathbf{w}_{-7}, \mathbf{c}\}$ となっているとき、石の打ち上げがなければ一手進めた場合の状態遷移関数は単純に

$$\mathcal{T}_{s'}(s) = \{ \mathbf{b}_0 + \mathbf{a}_0, \mathbf{w}_0 + \mathbf{a}_0, \dots, \mathbf{b}_{-7} + \mathbf{a}_{-7}, \mathbf{w}_{-7} + \mathbf{a}_{-7}, \mathbf{1} - \mathbf{c} \}$$

のように表すことができる。ここで a_{-i} は i 手前の着手であり、着手する座標に対応する one-hot 形式の配列である。石の打ち上げがない限り、状態遷移関数はこのような定数の足し算と符号の反転で表現することができ、この場合は容易に微分が可能である。よってこの状況では式 (2) は

$$\frac{\partial}{\partial s_0} \hat{v}(s_0) = \frac{1}{|T|} \sum_{s \in T} \text{FlipColor} \left(\frac{\partial}{\partial s} v(s), s_0, s \right) \quad (3)$$

と簡略化でき、Saliency Map を導出することができる。ここで $\text{FlipColor}(\cdot, s, s')$ は s と s' の手番が異なれば手番に対応する微分値を反転するような関数とする。

AlphaGo Zero では局面の符号化を「自分の石」と「相手の石」という方式で行っている [1]。そのような場合では適切に特徴を並び替えることで同様の出力結果が得られる。また MCTS における展開の閾値が 1 より大きい数であった場合は加重平均をとる。

4. 重要な石の検出実験

既存手法である Saliency Map と SmoothGrad および 3 章で提案した MCTS による評価値に対する Saliency Map を囲碁 DNN に適用した際に得られる出力に関して評価を行った。実験は 9 路盤用に学習を行った DNN を対象に行った。DNN は AlphaGo Zero と同様の構造を持つが、Residual Block の数を 16 に削減して実装した。入力に用いた特徴も図 2 と同様に、7 手前から現在の石の配置と手番の色のみを用いた $17 \times 9 \times 9$ の特徴を用いた。学習は公開されている aya の自己対戦棋譜 *1 のうちの 2,000,000 局を用い、一局につき一局面を抽出 *2 して教師例とした。学習後の DNN は Pachi (1 手 10,000 プレイアウト) に対して表 1 の対戦結果となった。

評価実験には学習に用いなかった対局を使用した。4.2, 4.3 章では 100 対局 9,947 局面を使用した。4.4 章では 1000 対局を用い、それぞれの対局から一局面をランダムに抽出した 1,000 局面を使用した。

4.1 Saliency Map の結果の集約方法

Saliency Map や SmoothGrad によって得られる出力の

*1 <http://www.ysss-aya.com/ayaself/ayaself.html>
閲覧日 2018 年 10 月 16 日

*2 一局に一回ランダムな着手があり、その直後の局面を採用した

表 1 Pachi (10,000 playouts/move) に対する勝率
Table 1 Winning rates against Pachi (10,000 playouts/move)

	勝率
Policy network (softmax)	13.4 %
Value network (一手読み)	62.9 %
MCTS (400 playouts/move)	91.0 %

次元は入力の次元と等しいが、これらの値の性質を個々に調査することは難しいため、出力結果を集約して次元を圧縮し、元の入力画像のピクセル数と等しくすることが行われている。例として文献 [2] では RGB 画像 $3 \times w \times h$ に対して得られた結果 $3 \times w \times h$ をピクセルごとに絶対値の最大のものを採用することで出力結果を集約している。

様々な集約方法が考えられるが、今回の実験ではそれぞれの座標ごとに、対応する出力の絶対値の総和を取ることによって 9×9 の値に集約し、集約した値をその座標の重要度として囲碁 DNN の判断結果に影響を与えていると仮定した。

4.2 方策評価に対する Saliency Map

方策予測 (policy network) の出力に対して Saliency Map/SmoothGrad の出力が意味のあるものとなっているかどうかを検証するために実験を行った。実験では 4.1 章で説明した集約方法により最も『重要』であるとされた座標の石が出力結果に与える影響を計測した。この実験手法は文献 [12] で行われた評価手法を参考に考案した。局面 s_0 に対して Saliency Map/SmoothGrad を適用し、最も重要である座標を得たのち、その座標に石がある場合はその石を取り除く (なければ何もしない) ことで局面 s' を得る。 s_0, s' における方策予測の出力の変動を交差エントロピー $H(p(s_0), p(s'))$ の大きさによって計測した。この実験で用いている DNN の入力は図 2 と同様、7 手前からの石の配置の履歴が存在するため、石を取り除く場合には 7 手前から現在の座標すべてから石を除く。図 3 に、元の局面の手数ごとに交差エントロピーの平均をとったグラフを示す。SmoothGrad については $n = 128$ とし、 σ の値を 0.05, 0.10, 0.25 として計測を行った。 $\sigma = 0$ は Saliency Map に対応する。また比較のため石を取り除く座標を、ランダムに選択したものおよび出力結果が最小である (つまり重要度が最小の) ものについて計測した結果も示す。重要度最小のものは $\sigma = 0.25$ のときの SmoothGrad によって計測したものを掲載している。他の手法 (Saliency Map, $\sigma = 0.05, 0.10$ の SmoothGrad) によって得られた重要度最小のものでもほとんど同じものが得られたため結果からは除いている。

図から、交差エントロピーの大きさはランダムに石を取り除いた場合よりも Saliency Map/SmoothGrad を用いて選択したときのほうが有意に大きくなっていると読み取れる。また傾向として中盤から終盤において交差エントロ

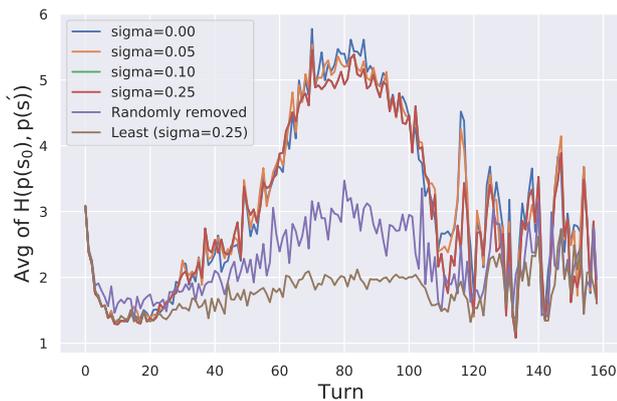


図 3 方策評価の手数ごとの平均変動値 ($H(p(s_0), p(s'))$)

Fig. 3 Average cross-entropy difference as functions of turns

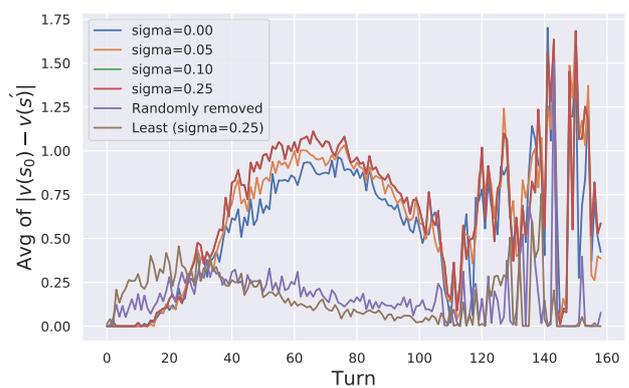


図 5 局面価値評価の手数ごとの平均変動値 ($|v(s_0) - v(s')|$)

Fig. 5 Average evaluation difference as functions of turns

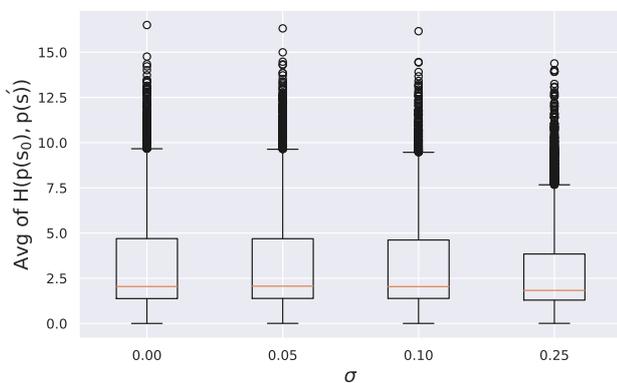


図 4 パラメータごとの方策評価の平均変動値 ($H(p(s_0), p(s'))$)

Fig. 4 Average cross-entropy difference by parameter σ

ピーの変動が大きい。序盤はどこかの石を取り除いても大きく予測が外れ、100 手目以降はデータ数が少ないため傾向が不安定である。したがって、囲碁 DNN の方策予測に対して適用した結果は中盤から終盤においては方策予測に大きく関わるような石を重要であると認識していると考えられる。

Saliency Map と SmoothGrad の違いを見るため、計測した交差エントロピーを箱ひげ図で表したものが図 4 である。Saliency Map と SmoothGrad で結果にあまり差が見られず、どれも同じような出力結果が得られていると考えられる。

4.3 局面価値予測に対する Saliency Map

次に、局面価値予測 (value network) に対して 4.2 章と同様の実験を行った。今回は交差エントロピーではなく、局面間の価値予測の絶対値 $|v(s_0) - v(s')|$ を指標として用いた。図 5 に手数ごとに評価値の変動を平均をとったグラフを、図 6 に手法ごとに箱ひげ図を描画したグラフを示す。

全体の傾向として方策予測のときと同様にランダムに石を取り除いた場合よりも変動が大きい。序盤の傾向として、

ランダムに取り除いたものや重要度最小の石を取り除いたものは予測の変動が大きく、Saliency Map/SmoothGrad で得られたものはほとんど変動しない。これはランダムに石を取り除くと学習データに含まれないような稀な局面に変化してしまい、そのような局面では局面評価値が大きく変化してしまっているためだと考えられる。中盤から終盤にかけては Saliency Map/SmoothGrad が認識した石の影響が大きく現れている。方策予測のときと同様、100 手目以降はデータ数が少ないため特定の傾向を見出すことは難しい。

方策予測のときとは異なり、Saliency Map と SmoothGrad を比較した場合 σ が増えるにつれて評価値の変動がより大きくなっていることが 2 つの図から読み取れる。考えられる説明として、方策予測の場合は石の有無がより大きく影響を与えるために SmoothGrad による正規乱数の影響が大きく、その手法の利点である多数の類似した入力の平均による雑音の除去を打ち消してしまったということが挙げられる。一方で局面価値予測の場合はノイズを加えた場合でも価値予測に影響はほとんどなく、SmoothGrad の雑音除去が効果的に働いたと考えられるため、この実験で用いた指標に関しては SmoothGrad のほうが性能がよくなったと考えられる。

4.4 MCTS に対する Saliency Map

最後に 3 章で提案した、MCTS によって得られた局面価値予測に対する Saliency Map に対して評価実験を行った。適用した局面には石の取り上げを含む場合があるが、Saliency Map の計算はすべて式 (3) で求めた。この実験では重要だと出力した座標に対して正規乱数によるノイズを加えた入力を用いて評価値の変動を計測した。以前の実験のように石を取り除いてしまうと探索する局面が大きく異なってしまうことを防ぐためである。図 7 に正規乱数の標準偏差を変えつつ評価値の変動の絶対値 $|\hat{v}(s_0) - \hat{v}(s')|$ を計測した結果を示す。先程の実験と同様、ノイズを加え

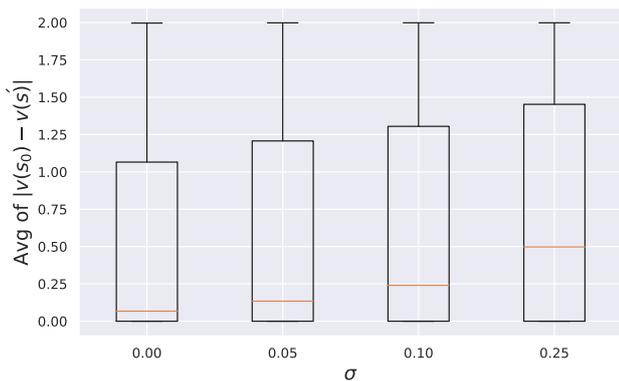


図 6 パラメータごとの局面価値評価の平均変動値 ($|v(s_0) - v(s')|$)

Fig. 6 Average evaluation difference by parameter σ

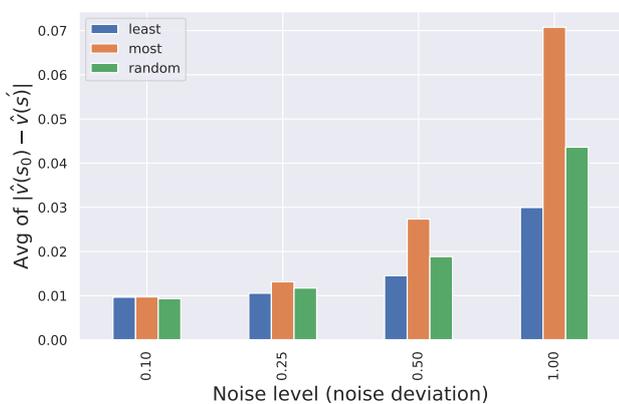


図 7 ノイズレベルごとの MCTS 評価値の平均変動値 ($|\hat{v}(s_0) - \hat{v}(s')|$)

Fig. 7 Average MCTS evaluation difference by noise level

る座標をランダムに選択した場合と重要度が最も低いものにした場合を比較のために掲載する。雑音の影響が大きくなるにつれて、提案手法である MCTS に対する Saliency Map によって得られた座標の影響の度合いが大きくなることが確認できる。先述の実験と比較して評価値の変動が非常に小さいが、これは石を取り除く先述の実験とは異なり、雑音が入力に乗っているのみなので比較対象とはならないことに留意されたい。ランダムに取り除く場合と比較すると最重要とされた座標に雑音を乗せた場合は大きく変化しており、提案手法により重要な座標を検出できていると考えられる。

5. おわりに

本稿では Saliency Map と SmoothGrad を用いて囲碁の DNN の判断の根拠を可視化することを目的とした。また MCTS による局面価値予測に対して Saliency Map を適用する手法を考案した。提案手法では囲碁の状態遷移関数を簡素化し、石の打ち上げがない場合では正確に計算できることを示した。

評価実験では既存の文献で用いられた評価方法を踏襲し

て Saliency Map/SmoothGrad が『重視』する座標の妥当性を検証した。また同様のアイデアにより提案手法を評価し、その効果を検証した。

今後の課題として、人間が用いる囲碁の特徴との対応づけがまず挙げられる。今回評価した指標はすべて単一の座標のみを対象としているが、普通人間が判断の根拠とするものは複数の座標のパターンである。このような対応づけができればより人間が DNN を解釈しやすくなるため有意義であると考えられる。

謝辞 この研究の一部は、JSPS 科研費 16H02927, 17J09685 と JST さきがけの支援を受けています。

参考文献

- [1] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T. and Hassabis, D.: Mastering the game of Go without human knowledge, *Nature*, Vol. 550 (online), available from <http://dx.doi.org/10.1038/nature24270> (2017).
- [2] Simonyan, K., Vedaldi, A. and Zisserman, A.: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, *CoRR*, Vol. abs/1312.6034 (online), available from <http://arxiv.org/abs/1312.6034> (2013).
- [3] Smilkov, D., Thorat, N., Kim, B., Viégas, F. B. and Wattenberg, M.: SmoothGrad: removing noise by adding noise, *CoRR*, Vol. abs/1706.03825 (online), available from <http://arxiv.org/abs/1706.03825> (2017).
- [4] Schraudolph, N. N., Dayan, P. and Sejnowski, T. J.: Temporal Difference Learning of Position Evaluation in the Game of Go, *Advances in Neural Information Processing Systems 6* (Cowan, J. D., Tesauro, G. and Al-spector, J., eds.), Morgan-Kaufmann, pp. 817–824 (online), available from <http://papers.nips.cc/paper/820-temporal-difference-learning-of-position-evaluation-in-the-game-of-go.pdf> (1994).
- [5] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T. and Hassabis, D.: Mastering the game of Go with deep neural networks and tree search, *Nature*, Vol. 529 (online), available from <http://dx.doi.org/10.1038/nature16961> (2016).
- [6] Tian, Y., Jerry Ma*, Qucheng Gong*, Sengupta, S., Chen, Z. and Zitnick, C. L.: ELF OpenGo, <https://github.com/pytorch/ELF> (2018).
- [7] Gelly, S. and Silver, D.: Combining Online and Offline Knowledge in UCT, *Proceedings of the 24th International Conference on Machine Learning, ICML '07*, New York, NY, USA, ACM, pp. 273–280 (online), DOI: 10.1145/1273496.1273531 (2007).
- [8] Couloum, R.: Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search, *Computers and Games* (van den Herik, H. J., Ciancarini, P. and Donkers, H. H. L. M. J., eds.), Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 72–83 (2007).
- [9] Rosin, C. D.: Nested Rollout Policy Adaptation for

Monte Carlo Tree Search, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011* (Walsh, T., ed.), IJCAI/AAAI, pp. 649–654 (online), DOI: 10.5591/978-1-57735-516-8/IJCAI11-115 (2011).

- [10] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F. and Pedreschi, D.: A Survey of Methods for Explaining Black Box Models, *ACM Comput. Surv.*, Vol. 51, No. 5, pp. 93:1–93:42 (online), DOI: 10.1145/3236009 (2018).
- [11] Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K. and Müller, K.-R.: How to Explain Individual Classification Decisions, *J. Mach. Learn. Res.*, Vol. 11, pp. 1803–1831 (online), available from <http://dl.acm.org/citation.cfm?id=1756006.1859912> (2010).
- [12] Ikeno, K. and Hara, S.: Maximizing Invariant Data Perturbation with Stochastic Optimization, *ArXiv e-prints* (2018).