The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

「待った」の概念を取り入れた効率的なオセロの学習

Efficient Learning of Othello Utilizing the Concept of Undoing Decision

成田 穂 *1　木村 大毅 *2
Minori Narita　Daiki Kimura

*1東京大学　*2IBM Research AI
The University of Tokyo

Combination of Monte Carlo Tree Search (MCTS) and deep reinforcement learning represented as methods such as AlphaZero has achieved incredible performance, while it requires high computation resources and much training time. In this study, we propose a novel MCTS-based algorithm, where we introduce “failure rate” to facilitate efficient exploration and hence it shortens training time. This algorithm makes the agent prioritize the exploration of the states that are important to winning. Our method has outperformed AlphaZero in the first few iterations.

1. 序論

近年の強化学習の発展は盛んであり, Deep Q Network [1] をはじめ, 多くの研究 [2–6] でゲーム分野への応用が研究されている. これらの中でも, AlphaZero [6] は, 自己対戦の機構を取り入れることで従来必要であった膨大な学習データの削減に成功し, また一切のドメイン知識を必要とすることなく, わずか 24 時間の学習で将棋のプロ棋士を超える性能を獲得したことから, 大きく注目されている.

AlphaZero の学習は, モンテカルロ木探索（MCTS）を元にした自己対戦からの学習データ生成と, 深層強化学習を用いたネットワークパラメータの更新から構成されている. MCTS は, 複数回のゲームのシミュレーションを行う中で平均報酬の高い手を探索するアルゴリズムである. [6] では, 報酬として予測勝率と, これまで訪れた回数が少ないノードの探索に重きを置く “楽観度” の和を用いて MCTS を実行している. このように探索と利用のバランスがとられたアルゴリズムによって, 高い性能が実現されている.

しかしながら, AlphaZero は学習に非常に大きな計算コストが必要となるという問題点がある. 学習時の自己対戦には約 5,000 個の TPU, ニューラルネットワークのパラメータ更新には 64 個の TPU が用いられており, 一般的な汎用マシンで学習を行えるコストではない. 従って, より少ない計算コストで効率的に学習を行うアルゴリズムが求められる.

ところで, 実際のボードゲームで人間同士が練習対戦するとき, オセロや将棋等では, 相手からの次の一手を受けて, 一手前の自分の手が失敗であると気付いた時, 戻って別の手を試してみたい時がある. そのような局面はしばしば勝敗を分ける重要な局面である. このような練習の場面において, 一手戻ることを「待った」と言う. 我々は, この概念の類推から, 失敗したと判断した場合に, その手を重点的に学習することで, より効率的に学習ができるのではないかと考えた.

「待った」の概念をヒューリスティックに実装する場合, 一手前よりも予測勝率が下がった, すなわちうまくいなかった手の周辺を何度もやり直して, その度にネットワークのパラメータを更新する逐次的な手法が考えられる. しかし, このような手法を取り入れると, 学習データの偏りが大きくなるほか, 様々な手を試すまでにより長い時間がかかってしまい, 効率的とは言えない.

そこで, 我々は完成度の高い AlphaZero のアルゴリズムの形式を活かし, その中に, 勝敗を分ける重要な盤面に重み付けをする仕組みを導入することを考えた. 強化学習の手法の中には Priotized Experience Replay [7] というものがある. これは学習の際に, 例えば TD 誤差のような尺度を用いてサンプリングの優先度をつけて学習を行う手法であり, 学習を効率化が可能である. 我々は, この考え方を MCTS に応用し, 探索の際に優先度をつける手法を提案する. 優先度は, 推定された Q 値（場面における行動価値）の現在の手と前の手の差分（これを「失敗度」と呼ぶ）を算出し, その値が大きく下落した時, 即ち失敗だと判断した局面に重きをおくように定義する. これにより, 探索の中で予測勝率を下げた手の周辺, すなわち勝敗を分ける重要な局面を優先的に探索し, 学習コストを低減する. ランダム性の大きな学習初期において, 大きく失敗した手に探索を誘導することは, 学習の収束にも役立つと予想される. 本稿では提案手法と Priotize を行わない従来手法を 6×6 のオセロに適用し, 評価した. 本手法の利点は以下の 3 点である.

- 予測勝率の下げ幅の大きな手の周辺を優先的に探索することによる学習効率の向上
- 勝敗を分ける重要な局面に重きをおいた探索による学習時間の低減
- 学習初期の探索のランダム性の低減による, 学習の収束促進

2. 提案手法

本章では, 提案手法の基となる AlphaZero の仕組みと, 我々の提案部分を説明する. AlphaZero の学習は, MCTS を用いた自己対戦による学習データ生成と, それを用いたポリシー・バリューネットワークのパラメータ更新部分から構成されている. MCTS は, それぞれが異なるボードの配置からなるノードで構成される木を探索する. それぞれのノードは, 行動価値 $Q(s,a)$ と, これまでのシミュレーションでの訪問回数 $N(s;a)$ を保持している. $Q(s,a)$ は, その行動 a を状態 s にて選択した時に期待される予測勝率である. また, ニューラルネットワークが出力する現在の方策に従った場合に, そのノードから取り得る行動 a の選択確率 $P(s,a)$ も各ノードで保持している. [6]

連絡先: 成田穂，東京大学，千葉県柏市柏の葉 5-1-5 東京大学基盤棟 4E5，mnarita1207@gmail.com

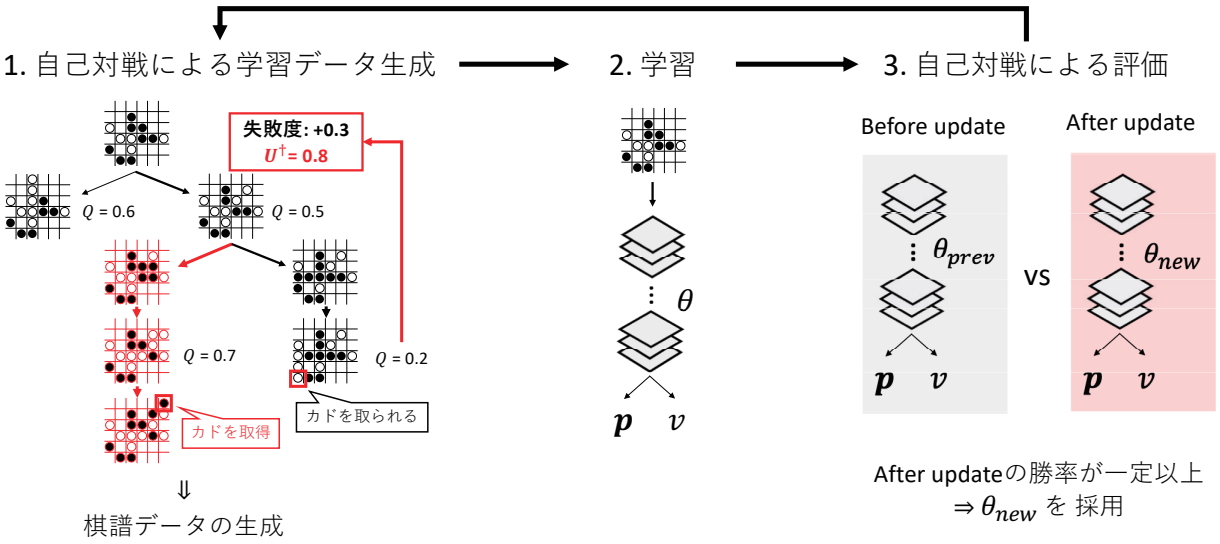

図 1: 失敗度付きモンテカルロ木探索を導入したオセロ学習の概要図

では, 予測勝率 $Q(s,a)$ と, まだ訪れていないノードに重きを置く楽観度 $b(s,a)$ を足し合わせた, 楽観度付き予測勝率 $U(s,a)$ を算出し, これが最大となる行動 a を選択しながら木を下っていく. そのため $U(s,a)$ は, 以下の式 (1),(2) で算出される. なお, c_{puct} は探索と利用のバランスを取るハイパーパラメータである.

$$b(s,a) = c_{puct} P(s,a) \frac{\sum_b N(s,b)}{1+N(s,a)} \tag{1}$$

$$U(s,a) = Q(s,a) + b(s,a) \tag{2}$$

本稿では, この $U(s,a)$ に対して「待って」の概念を応用した “失敗度” を導入する. 失敗度は, 自分の前回の手番における予測勝率から今回の予測勝率を引いた値であり, これが大きいほど探索が促進される. 失敗度 $f(s,a,t)$ の算出式を式 (3) に記す. ただし, c_{pri} は, 重み付けの大きさを決めるハイパーパラメータである. 探索の際に用いる値は, 失敗度と楽観度付き予測勝率 $U(s,a)$ を足し合わせた, 失敗度を考慮した楽観度付き予測勝率 $U^{\dagger}(s,a)$ と定義する. これを式 (4) に記す. なおこの時, 前の手より現在の手の方が予測勝率が高い場合には, 失敗度 $f(s,a,t)$ はゼロに戻る. また, $Q(s_{t-2},a_{t-2})$ は現在のノードから 2 つ前のノード, すなわち前回の自分の手番における行動価値を意味する. このようにして定義した評価値を用いて MCTS を実行し, 学習データの生成を行う.

$$f(s,a,t) = \max(\mathrm{c_{pri}}(\mathrm{Q(s_{t-2}, a_{t-2})} - \mathrm{Q(s_t, a_t)}), 0) \tag{3}$$

$$U^{\dagger}(s,a) = Q(s,a) + b(s,a) + f(s,a,t) \tag{4}$$

一方, ネットワークのパラメータ更新は AlphaZero と同様の手法を用いた. すなわち, 自己対戦により得られた棋譜データを用いて, 盤面 s を入力, 手の選択確率 $p = P(a|s)$ と状態 s からの報酬の期待値 $v \approx E[z|s]$ の 2 つを出力として, ネットワーク $(p,v) = f(\theta)$ のパラメータ θ を学習させる.

更に本稿では, AlphaGoZero [5] にて取り入れられている, パラメータの学習前と学習後で自己対戦を行い, 学習後のモデルの勝率が一定以上であるときのみパラメータの更新を行うようにした. これは, 失敗度による優先度付き探索は学習データに偏りを生じさせる要因になりうるため, 勝率による判定を行うことで局所解への収束を抑止することを狙った.

上記で説明した提案手法の概要図を図 1 に示す. 学習データ生成段階において $U^{\dagger}(s,a)$ を評価値として失敗度を考慮した探索を行い, 従来手法となる AlphaZero [6] と比較して学習効率の検証を行った.

3. 実験

本手法を 6 × 6 のオセロに応用し評価を実施する . 学習の各イテレーションで更新されたネットワークは, 前回までの最善のパラメータのものと 40 回対戦させ, 勝率が 60%以上であれば新しいものと置き換えることとした. また, 重み付けの大きさ c_{pri} は 0.5 で定数とした.

まずは, 対戦するエージェントを固定し比較を行う. 既存手法である AlphaZero [6] を一定量（学習イテレーション数 3）学習させた固定エージェントを用意する. なお, このエージェントは, 常に最大の駒をひっくり返す手を選択する単純な Greedy エージェントとの対戦勝率は 100%である [*1]. この固定エージェントと, 提案手法及び AlphaZero をそれぞれイテレーションごとに対戦させた. 固定エージェントに対する勝率を縦軸とした学習曲線を図 2 に示す.

図 2 を見ると, 特に学習の初期において, 提案手法の勝率が高いことがわかる. このことは, 学習の初期段階において, 従来手法に比べて有望な手を選択する傾向があることを示している. 一方で, 4 イテレーション目以降は従来手法の勝率が上がり, 提案手法の優位性は保たれない. この理由はいくつか考えられるが, 1 つには失敗した手の周辺の探索が, 学習の初期に有効に働き, 学習が進むと効果が減少するという可能性がある. これは, 我々がオセロを練習する時に, 「待った」を行って局所的な最適解を探すことが最も効果的になるのが初心者の時であるという直感にも則している. また別の原因としては, ハイパーパラメータとして定めた c_{pri} の値が最適化されてい

*1 固定エージェントとして Greedy エージェントの使用を考えたが, Greedy はあまりに弱いため不採用とした

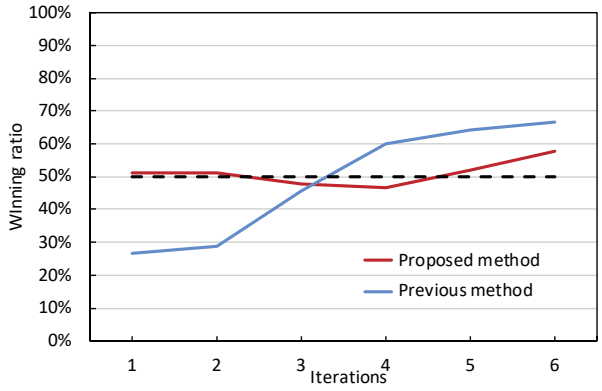

図 2: 学習 iteration が 3 の AlphaZero に対する学習曲線。提案手法と従来手法でそれぞれ 3 つエージェントを生成して総当たりを行い, その平均勝率を縦軸に示した.

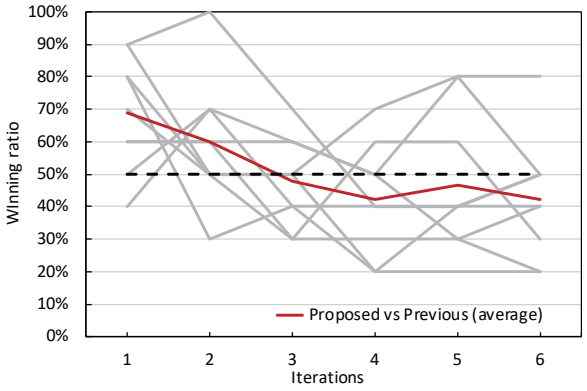

図 3: 提案手法と AlphaZero のエポックごとの勝率. 提案手法と従来手法でそれぞれ 3 つエージェントを生成し, 総当たりを行った. 灰色は各エージェントの対戦結果, 赤色は 9 通りの戦績の平均.

ない可能性がある. 今回は 0.5 で固定値としたが, より低い値や, イテレーションごとに値を連続的に変えていくなどの工夫が考えられる.

次に, 各エポックで提案手法と AlphaZero を対戦させた. その勝率の推移をグラフにしたものを図 3 に示す. これを見ると, 学習の初期においては提案手法が大きく勝ち越しているものの, 学習が進むうちに AlphaZero に段々と負け越すようになっていることが分かる. したがって, 提案手法では学習初期においては効率的に学習が進む一方, 学習が進むとその優位性は失われていくと考えられる. これは, 提案手法では楽に勝てる盤面ではなく, 勝敗を分けるような局面のデータを集中的に集めることになるため, 学習データに偏りが生じ, 勝率を下げていることが原因として予想される. 学習の収束後も高い勝率を維持するには, 失敗度の重みを段階的に減らしていく必要があると考えられる. さらに, 図中に灰色の線で示した各エージェントごとの対戦結果に注目すると, 対戦相手による勝率のばらつきが大きいことが分かる. このことは, 提案手法のアルゴリズムの動作があまり安定的でないことを示している. 失敗度付き探索を安定化させるために, 例えば失敗度の大きさに応じて値を正則化したり, イテレーションごとに失敗度の優先度を段階的に下げていくなどの対策が有効である可能性がある. これについては今後の課題である.

4. 結論

モンテカルロ木探索を効率化するために失敗度という概念を導入し, 探索の際に自分の一つ前の手番との予測勝率の最大値の下げ幅を利用して重要な局面を優先的に探索させることで, 学習の初期において効率的な探索を可能にした. 6×6 のオセロにおいて従来の AlphaZero と比較した結果, 学習初期において効率的に学習が進んでいることが分かった. このような優先度付き探索は学習コストや時間を低減する上で有望なアプローチであると考えられるが, 一方で学習が進むごとにデータのサンプリングの偏りが勝率に悪影響を与えることも分かった. この点については今後の課題である.

謝辞

本研究を進めるにあたり, 東京大学情報理工学系研究科の國吉康夫教授, 原田達也教授をはじめ, 情報理工学系研究科 2018 年度冬学期講義『先端人工知能論 II』ご担当の先生方にご支援を頂きました. ここに感謝の意を述べさせて頂きます.

参考文献

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, et al. “Human-level control through deep reinforcement learning”, *Nature*, 2015.

[2] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell, “Curiosity-driven exploration by self-supervised prediction”, *International Conference on Machine Learning (ICML)*, 2017.

[3] Daiki Kimura, Subhajit Chaudhury, Ryuki Tachibana, and Sakyasingha Dasgupta, “Internal Model from Observations for Reward Shaping”, *International Conference on Machine Learning (ICML) workshop*, 2018.

[4] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, et al. “Mastering the game of Go with deep neural networks and tree search”, *Nature*, 2016.

[5] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, et al. “Mastering the game of Go without human knowledge”, *Nature*, 2017.

[6] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, et al. “Mastering chess and shogi by self-play with a general reinforcement learning algorithm”, *arXiv:1712.01815*, 2017.

[7] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver, “Prioritized experience replay”, *International Conference on Learning Representations (ICLR)*, 2016.