dlshogi アピール文章

山岡忠夫
(開発者ハンドルネーム)

1　特徴

- ディープラーニングを使用
- 指し手を予測する Policy Network
- 局面の勝率を予測する Value Network
- モンテカルロ木探索
- 既存将棋プログラムの自己対局データを使った強化学習
- ブートストラップ法による Value Network の学習
- マルチタスク学習
- 詰み探索
- 序盤局面の事前探索（定跡化）

2　使用ライブラリ

- elmo[1]（Commits on May 29, 2017）

※学習データ生成、局面管理、合法手生成のために部分的に使用

3　各特長の具体的な詳細（独自性のアピール）

3.1　ディープラーニングを使用

DNN(Deep Neural Network)を使用して指し手を生成する。

従来の探索アルゴリズム(α β法)、評価関数(3 駒関係)は使用していない。

3.2　Policy Network

局面の遷移確率を Policy Network を使用して計算する。

Policy Networkの構成には、Wide Residual Network[2]を使用した。

入力の畳み込み 1 層と、ResNet 5 ブロック(畳み込み 2 層で構成)と出力層の合計 12 の畳み込み層で構成した。

---

[1] https://github.com/mk-takizawa/elmo_for_learn
[2] https://arxiv.org/abs/1605.07146

3.3 Value Network

局面の勝率を Value Network を使用して計算する。

Value Network は、Policy Network と出力層以外同じ構成で、出力層に全結合層をつなげ、シグモイド関数で勝率を出力する。

3.4 モンテカルロ木探索

対局時の指し手生成には、Policy Network と Value Network を活用したモンテカルロ木探索を使用する。

ノードを選択する方策に、Policy Networkによる遷移確率をボーナス項に使用したPUCTアルゴリズムを使用する。PUCTアルゴリズムは、AlphaGoの論文[3]に掲載された式を使用した。

また、末端ノードでの価値の評価に、Value Network で計算した勝率を使用する。

通常のモンテカルロ木探索では、末端ノードからプレイアウトを行った結果（勝敗）を報酬とするが、プレイアウトを行わず Value Network の値を使用する。

3.4.1 並列化

モンテカルロ木探索は、並列化が容易だが、Policy Network と Value Network を計算するための GPU の数以上の並列化を行うと GPU の使用で競合が発生する。

GPU は複数の計算要求をバッチで処理することが可能であるため、各スレッドからの要求をキューイングして、1 つの専用スレッドでバッチ処理することで競合を回避する。

3.5 既存将棋プログラムの自己対局データを使った強化学習

Policy Network の教師データには、elmo で生成した自己対局データを使用する。

単純に elmo の指し手を学習するのではなく、学習局面の価値と勝敗データと関連付けて学習を行う。

良い局面から負けになった手は、悪手として負の報酬を与え、悪い局面から勝ちになった手は善手として正の報酬を与える。

学習アルゴリズムには、AlphaGo の論文に掲載されている REINFORCE アルゴリズムを使用した。

3.6 ブートストラップ法によるValue Networkの学習

Value Network の学習の損失関数は、勝敗を教師データとした交差エントロピーと、elmo による深い探索(深さ 8)の評価値を教師データとした交差エントロピーの和

[3] https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf

とした。

このように、本来の報酬（勝敗）とは別の推定量（探索結果の評価値）を用いてパラメータを更新する手法をブートストラップという。

経験的にブートストラップ手法は、非ブートストラップ手法より性能が良いことが知られている。

3.7　マルチタスク学習

Policy Network と Value Network のネットワーク構成が同じ層を共通化し、出力層を分けることで、同時に学習を行う。

関連する複数のタスクを同時に学習することをマルチタスク学習という。

タスク間に関連がある場合、単独で学習するよりも精度が向上する。

また、対局時に Policy Network と Value Network を同時に計算できるため、高速化の効果もある。

3.8　詰み探索

モンテカルロ木探索は最善手よりも安全な手を選ぶ傾向があるため詰みのある局面で手を抜くことがある。

対策として、詰み専用の探索を行い、詰みの場合はその手を指す。

また、モンテカルロ木探索の末端ノードでも、数手の詰み探索を行い、詰みの局面を評価できるようする。Policy Network と Value Network の計算中に、CPU が待ち状態の間に詰み探索を行うため、探索速度が落ちることはない。

3.9　序盤局面の事前探索（定跡化）

出現頻度の高い序盤局面は、対局時に探索しなくても、事前に探索を行い定跡化しておくことができる。また、事前に探索することで、対局時よりも探索に時間をかけることができる。

4　学習について

4.1　学習データ、パラメータ

- 学習データ：elmo で生成した 35.8 億局面
- ミニバッチサイズ：64
- 学習アルゴリズム：SGD
- 学習回数：3 エポック
- 1 エポックごとに学習率を半減

4.2　学習結果

Policy Network の一致率：46.0%※

Value Network の一致率：78.1%

※強化学習を行っているため単純に一致率での評価はできないが参考として記載

以上