

『海底』アピール文書

迫田真太郎

平成 30 年 5 月 2 日

1 概要

『海底』は 2017 年 1 月から開発が始められた将棋ソフトです。将棋ソフト開発経験を通じて作者の技術力を向上させ、採取的により強い将棋ソフトを作ることを目指しています。

去年は第 27 回世界コンピュータ将棋選手権、第 5 回電王トーナメントに参加し、結果はそれぞれ 3 勝 4 敗の 24 位で一次予選敗退、4 勝 4 敗の 26 位で予選敗退でした。今年は勝ち越しを目指して日々開発を行っています。

将棋ソフトのレート測定しているサイト¹では WCSC27 版が R:707, SDT5 版が R:970 となっており、現時点で最新のバージョンは SDT5 版に対して 9 割程度勝つことのできる棋力となっています。

今回はさらに dlshogi 開発者・山岡さんの著書²を参考に、DeepLearning を用いたバージョンも開発したため、これら線形評価関数・ $\alpha\beta$ 探索のものと、DeepLearning・モンテカルロ木探索のものを楽観合議させるプログラムで参加します。

2 線形評価関数・ $\alpha\beta$ 探索側

2.1 探索部

$\alpha\beta$ 法を基本とした探索を行います。StockFish や強豪ソフトを、開発者自身の技術力が許す範囲で部分的に模倣した形となっており、採用している枝刈りとしては

- Razoring
- Futility pruning
- Null Move Pruning
- 多重反復深化
- PVS(Principal Variation Search)
- LMR(Late Move Reduction)

が挙げられます。

また残り探索深さが 1 未満になった段階で静止探索を行っています。静止探索では最初の一度だけその局面で駒を取る手を全て生成し、以降は直前に動いた駒を取り返す手だけを生成しています。

2.2 評価関数

2 駒関係 (KP,PP) を特徴量とする線形評価関数を用いています。SDT5 版は floodgate の棋譜から Bonanza メソッドを用いて学習したものでしたが、そこから elmo 式の RootStrap を行うことで棋力が向上しました。

棋譜を書き出すのではなく数百局の自己対局を行い、棋力変化の検証と学習用データ生成をまとめて行ったのちその場で学習することを繰り返しているのですが、かなり早い段階で頭打ちとなってしまっています。

2.3 指し手生成など

基本的には Bitboard を用いて指し手生成を行っています。さらにピンや自殺手などの細かい合法性の

¹<http://www.uuunuun.com/>

²『将棋 AI で学ぶディープラーニング』。マイナビ出版

取り扱いのため、各マスについて利きの数と、長い利きがある方向を管理しています。

指し手のオーダリングに関しては駒を取る手を先に生成することを基本とし、駒を取らない手については history を用いて浅い探索での結果が考慮されるようにしています。

3 DeepLearning・モンテカルロ木探索側

3.1 探索部

プレイアウトを行わず、リーフノードにおいて勝率を予測値で近似するモンテカルロ木探索を行います。ニューラルネットワークが出力する指し手の確率分布を基にノードを選択、展開していきます。

勝率予測はマルチタスク学習を行わせたニューラルネットワークの出力と、リーフノードにおいて王手回避、Recapture の手だけを生成する静止探索を行う線形評価関数の出力との平均値を用いて行います。GPU でニューラルネットワークの計算を利用して CPU で静止探索を行うため、それほど探索効率は悪くありません。

3.2 評価関数

2つの畳み込み層を持つ ResidualBlock を6個つなげた構造になっています。floodgate2016年の棋譜から学習させ、最終的な性能はテストデータに対して

指し手一致率：40.1%

勝率予測の一致率：68.8%

となっています。

3.3 実装

学習はライブラリ:python-shogi をもとに tensorflow を python で動かすことで行いました。学習アルゴリズムは dlshogi を参考にしたためこちらもライブラリ登録しています。

実戦では tensorflow を C++ から動かします。探索速度は 2000 シミュレーション/秒ほどとなっています。単体では上の線形評価関数・ $\alpha\beta$ 探索側に対して 35% ほどの勝率となります。

4 構成

2台のリモートマシンへ現地のノートパソコンから ssh で接続し、ノートパソコンにおいて楽観合議を行います。

合議部分には SimpleGougiShogi³ を使用させていただきました。

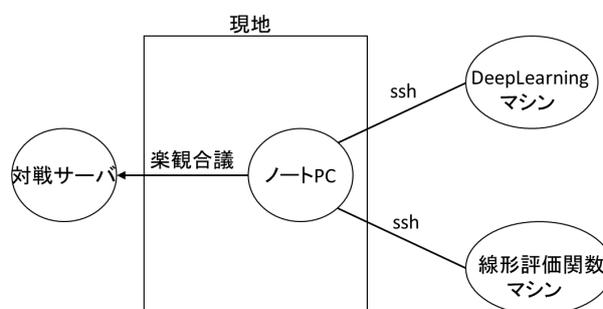


図 1: 構成

³<https://github.com/tttak/SimpleGougiShogi/>