

「ねね将棋」アピール文書

ねね将棋(NEural NETwork Shogi)は、深層学習(Deep Learning)を用いた評価関数により思考する将棋ソフトです。従来の 3 駒関係+ α β 探索に代わるアーキテクチャで強くすることを目指しています。

使用ライブラリ

やねうら王 [1] (ソースコードおよび教師局面) : ユーザ定義エンジンの追加がしやすいため
Apery, elmo : 使用の可能性があったため申請しましたが、不使用となりました

基本構成

USI 通信・合法手の生成までをやねうら王ライブラリで行い、探索部・評価関数は独自に実装しています。

探索部は MCTS を用い、選択的な探索を行います。評価関数において GPU を効率的に動作させるため、100~1000 局面を同時に評価できるようキューを用いたシステムになっています。前年度はシングルスレッドで探索を行っていましたが、これをマルチスレッド化し、末端局面での詰み探索を行うよう改良しました。優勢になってからちゃんと勝ち切れるようになった印象があります。

評価関数は Convolutional Neural Network (CNN)を用います。教師ありで学習しています。前年度はフレームワークとして Chainer を用いており、対局エンジンの動作に python 環境が必要でした。今回は C++から直接呼び出せる CNTK に変更したことで、python 環境が不要となり配布が容易となりました。

定跡はやねうら王の標準定跡です。

CNN

CNN の構造は、 3×3 の convolution 2 層を 1 つのブロックとし、Batch Normalization、ショートカット接続を加えた residual 構造です。Convolution のチャンネル数は 192、ブロック数は 19 です。前年より深さを増やしました。192ch, 16 層→192ch, 42 層

入力は 119 チャンネル、 9×9 サイズです。dlshogi [2]を参考にさせていただき、駒の配置およびある地点に効いている駒の数、王手かどうかを含めています。

出力は 2 系統あります。方策部分は 27 チャンネル、 9×9 サイズです。CNNShogi [3]を参考にさせていただき、移動先マスごとに、駒の種類と移動元を表現する 27 チャンネルです。棋譜の正解の指し手に対して softmax cross entropy loss で学習します。静的評価値部分は 2 チャンネルあり、勝ちと負けに対応します。(1 チャンネルにして、tanh 関数で勝率にし

でも意味的には同じです。フレームワーク上実装しやすいほうを採用しているだけです。) elmo と同様、棋譜の評価値を sigmoid で勝率に直したものと、対局の勝敗それぞれと softmax cross entropy loss をとり、その和を損失として学習します。

チャンネル数・ブロック数を増減させた結果として、現状より計算量が増える方向だと精度の増加よりも nps の低下の影響のほうが大きいようです。

思考時間制御

前回は予想残り時間を均等分割するだけの単純な思考時間制御で、明らかな手がある場面でも時間をかけていました。今回は、指し手の選択確率 (MCTS におけるルートノードから子ノードへの訪問回数の比率) が特定の 1 手で十分大きくなり、残り思考時間内に大小関係が逆転する確率が一定以下になったらその時点で指す機構を実装しました。

「現在トップの指し手確率」 $\times -5.326 + \log_2$ (「現在までのゲーム木のノード数」/1024) $\times -0.306 + \log_2$ (「予定残り思考時間内に評価できるノード数」/1024) $\times 0.377 + 0.798$

という単純なモデルで、今後の思考で指し手が変化する確率を算出することとしました。パラメータは実際に思考を行った結果にフィッティングさせて決定しています。

本番で利用するハードウェアは強力なため、貧弱なハードウェアで作ったデータへのフィッティングの結果を外挿して活用することになります。数局試した感じでは悪くない挙動です。

ハードウェア

Amazon EC2(クラウド)の p3.16xlarge インスタンスを利用予定です。Tesla V100 GPU を 8 台搭載しています。GPU をフル稼働させると 30,000nps 程度出ます。

本来は 16bit 演算で高速化を狙いたかったのですが、実装上の問題で実現できませんでした。

混雑しているとマシンが確保できない場合もあり、運次第です。クラウド利用不能時は方策関数のみで指すモードをローカルマシンで動かします。この場合は lesserkai に勝ったり負けたりする程度の強さです。

目標

1 次予選突破。前回より少し強くなっているはずです。

[1] 磯崎元洋 <https://github.com/yaneurao/YaneuraOu>

[2] TadaoYamaoka <https://github.com/TadaoYamaoka/DeepLearningShogi>

[3] not522 <https://github.com/not522/CNNShogi>

2019 年 1 月 7 日作成 (5 月 1 日改訂) 日高雅俊 <https://github.com/select766>