

「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。

昨年との主な違いは評価関数の実行部分で TensorRT ライブラリを使用するよう変更し高速化した点です。

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。

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。マルチスレッドで探索し、末端局面での詰め探索を行います。

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。前年度使用していたフレームワーク CNTK が開発終了となったため、PyTorch に変更しました。棋力そのものには影響しません。

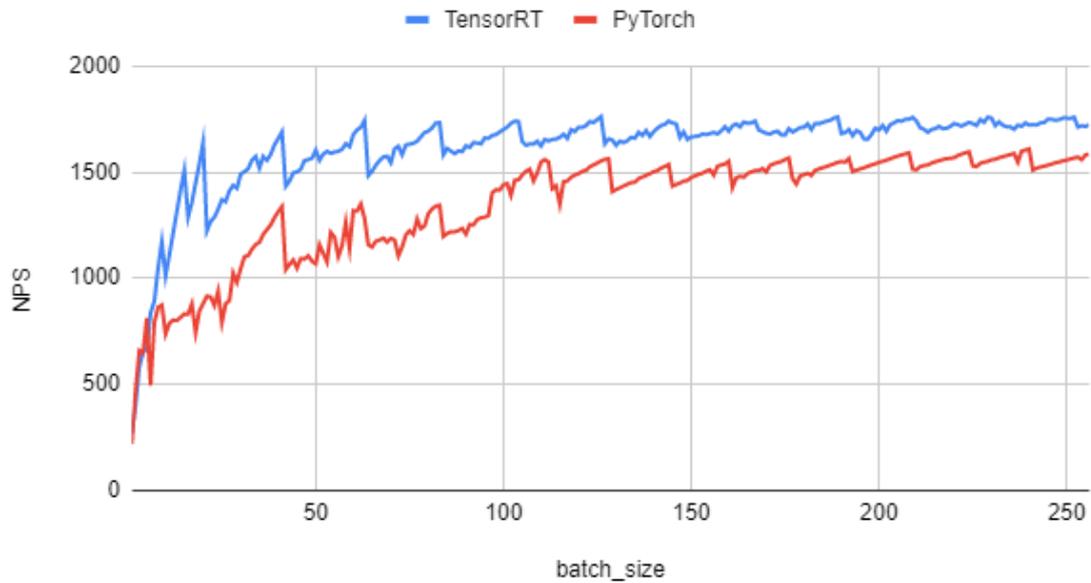
対局時の評価関数の実行には、NVIDIA 社から提供されている TensorRT を用います。

TensorRT は学習済みのモデルを高速に評価することに特化したライブラリです。

PyTorch で学習したモデルを ONNX フォーマット (深層学習フレームワーク間でモデルをやり取りするための共通フォーマット) で出力することにより、TensorRT で読み込みことができます。

今回使用する CNN モデルを TensorRT と PyTorch で実行したときの速度の比較を掲載します。NPS は node per second で、ここでは探索とは関係なくダミーデータを流した場合のバッチサイズ/評価時間で算出します。TensorRT のほうがほとんどのバッチサイズで高い性能を示します。

ライブラリ間のCNN評価速度(GTX1060)



さらに、計算は通常 32bit 浮動小数点で行うところ、16bit 浮動小数点を使用するモードでの実行速度を比較します。GPU8 台を別々のスレッドで並列実行した場合の 1GPU あたりの速度も表示します。16bit の場合は Tensor Core と呼ばれる演算器が使用されていると推察され、大幅な速度向上が実現できます。8GPU を同時に使用する場合は 1GPU あたりの性能が下がっていますが、これは入出力の行列データの転送に同じメモリバスを共用しているためと予想されます。

ビット数・並列実行数によるCNN評価速度(V100)



本番対局では TensorRT を 16bit 浮動小数点モードで動作させる予定で、昨年より nps が向上していることが期待できます。
定跡はやねうら玉の標準定跡です。

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。

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>

2020 年 3 月 23 日作成 日高雅俊 <https://github.com/select766>