第32回世界コンピュータ将棋選手権　「S.Lightweight-EF」アピール文書

2022.5.8

神田 剛志

■開発動機

DL系単体での強さは各大会の結果にも表れてきてはいるものの、ハード側に
それ相応のスペックが必要なように見えます。
そのため、家庭用ローカルPCの範囲で、上位ソフト陣に比肩する棋力を
獲得できることを示したい。
名前の通り軽く速く。末尾のEFはDNNのモデル(EfficientNet)が由来です。

■アピールポイント/開発過程

①モデルアーキテクチャ

本家のResNetをEfficientNetで再構築し、１から学習しなおしました。
第2回電竜戦時のモデルから6層追加した改良型に当たります。
EfficientNet単体ではなく、入力部に7層のResidual blockを入れ、
そこから更にEfficientNetに接続しています。

②USIエンジンのパラメータ設定変更によるNPS向上

GPUに局面を渡す際のバッチサイズを1024に上げています。
これと軽量なモデルと組み合わせにより、ローカルPC環境での平均NPSを向上
させています。

③GCT学習データによる教師あり学習とS.Lightweight-EF自身による強化学習

dlshogiチームが公開してくださっている学習データ（以下ⅰ,ii,iii）と
S.Lightweight-EFの自己対局データを用いた強化学習を実施しています。

ⅰ. floodgateから抽出・作成された学習データ
ii. GCT電竜の自己対局データ
iii. 水匠による入玉局面データ
iv. 書籍「強い将棋ソフトの創りかた」に付属する学習データ
v. S.Lightweight-EF自身による自己対局データ

④KL情報量による時間制御

dlshogi本家に倣い、Policyの確率分布と探索後の確率分布のKL情報量を
用いた時間制御を導入しています。

⑤定跡の使用

初期局面の事前探索結果を利用することで、持ち時間の消費を抑えます。
以下2つの定跡を大会で使用します。

・s-book_blackをベースに、S.Lightweight-EFのモデルを使って
さらに深掘したもの
・S.Lightweight-EFのモデルのみを利用し作成したもの

⑥探索部の変更

PUCTアルゴリズムに従って探索木を降りていく際、各子ノードの着手確率を
利用した簡易的な枝刈りを実施することで、最大UCB値の子ノード選択処理に
かかる時間を短縮し、探索処理を効率化・高速化しています。

⑦MultiStream対応
　　dlshogi本家に倣いMultiStreamに対応することでNPSを向上させます。
⑧入力特徴量作成の改善
　　dlshogi本家に倣い入力特徴量の作成処理を改善し、NPSを向上させます。

■実験結果（アピールポイント/開発過程の各番号に対応しています）
①/③：
　　公開されている「第2回世界将棋AI電竜戦エキシビジョンバージョン」の
　　dlshogiのモデルと、持ち時間1分1秒加算で先手後手入れ替えありで
　　100局連続で対局を行った結果、以下のような結果となりました。
　　（ハードスペック：Ryzen7 3700X, RTX3070, RTX3080を使用）
　　【S.Lightweight-EF】
　　先手勝ち：41(勝率82%)
　　後手勝ち：26(勝率52%)
　　【model-dr2_exhi】
　　先手勝ち：24(勝率48%)
　　後手勝ち：9 (勝率18%)
　　※model-dr2_exhi側の探索パラメータについては、
　　第2回世界将棋AI電竜戦エキシビジョンバージョンにて公開された
　　ビルド済ファイルのデフォルト値を設定しています。
②/⑦/⑧：
　　benchmarkテストを実施した結果を示します。
　　平均NPS：126,016.53
　　最大NPS：185,777
　　最小NPS： 43,738
④/⑥：
　　時間が不足しており手元で計測できず。
　　2次予選以降は利用していません。
⑤：
　　前者に関しては1次予選、2次予選のみ使用。
　　後者に関しては1次予選から決勝リーグまで使用しました。

■追試可否
　　可能。

■使用ライブラリ等
　dlshogi：自己対局データ生成・探索部・モデル学習・定跡作成に利用
　Gikou2 ：検証時のテスト対局に使用
　Suisho3：検証時のテスト対局に使用
　Suisho4：検証時のテスト対局に使用
　elmo for learn：学習データ作成に利用
　BookConv：定跡の変換に使用