第３３回世界コンピュータ将棋選手権「アストラ将棋」アピール文書

令和５年４月
恒岡 正年

1. 全体の構成
　dlshogi の探索部、学習部を利用。
2. 独自に実装した部分
　Network 構造、学習方法、model の生成の工夫。（詳細は後述）
3. 開発動機
　ディープラーニングの特に学習部に興味があり、「強い将棋ソフトの創り方」という書籍の内容をトレースする事で開発をはじめた。
　したがって、自前の model を作る事に注力している。
4. 主な開発内容
　独自構造の model の作成・学習方法
　棋譜生成：6000po 程度、及び 15 秒+F1.2 秒程度の物を中心に現在約 30 万の棋譜作成
　学習には、自己生成した棋譜以外に上記書籍の学習用データを利用
　AobaZero の棋譜も一時期利用したが、現在は未使用。
　探索パラメータの調整(手作業・optuna 利用)
5. 探索ロジックの検討(UctSearch.cpp の改造)
　標準の UCB 値の計算式は n が十分に大きい時に最良の計算式となるが、限られた試行回数では必ずしも最適な方法ではないと思う。
　オリジナルの計算式を独自の式と置き換える。
　探索パラメータの再調整後、R+36 程度の向上を確認。
6. 持ち時間の節約のための簡単な定跡を手入力で準備した。
　ベストラインでは深く、それを外れた場合は(有利になっているはずなので)浅い。
7. 今後の開発予定(大会後)
　指し手選択ロジックの追加(後述)
　思考時間制御(後述)

---

8. Network 構造

Network を次の 3 つに分ける。

(1)データ入力部 (2)ResNET 部 (3)データ出力部

1）データ入力部

3 つの情報の合流位置とデータのサイズを振って実験した。

現在はオリジナルに Conv2d()を 1 つ追加した構造にしてる。

データサイズは、9x9 以外にも 11x11、11x9 を試した。Play Out 数固定での評価では有意に強くなるが、探索速度の低下が大きく探索時間一定の条件では弱くなった。ネットワーク層数またはカーネル数を増やす方が良い。

将来的にネットワーク層数またはカーネル数を増やしても改善効果が小さくなった場合に、データサイズ 11x11 を試してみたい。

2）ResNET 部

オリジナルの ResNET に限らず何通りかの形状(Conv2D()3 層の ResNET や入れ子構造の ResNET 他)のネットワークを試したが、最終的にオリジナルと同じ Conv2d() 2 層の ResNET にした。

データサイズは 9x9。(11x11、11x9 も試したが不採用)

カーネルサイズは 3x3 のみ評価。

データ入力側と出力側でカーネル数が異なる構造を採用。

ResNET 部の層数は 25。

3）データ出力部(Policy/Value Network の分岐後)

オリジナルの構造に Conv2D()を 1 つ追加している。

ResNET を何層か追加する実験も行ったが優位性は認められなかった。

4）その他

活性化関数は 主に ReLU を、最も出力に近い Conv2D()にのみ SiLU を使っている。

計算量は入力部と出力部に Conv2D()を追加しているため ResNET27 層相当。

9. 学習方法

1）今回の使用モデルでは、学習率(lr)：0.2 から始めて 0.000000012 まで等比数列的に減少させ学習した。減少率は主に 0.85。最後の数エポックは

学習率を固定値とした。
100 エッポックの学習で 2 週間～3 週間かかる。
　2）バッチサイスは 512 から初めて GPU メモリの許す限り順次大きくしている。
　3）マクロバッチを実装し、最終的に 12 倍まで学習単位を大きくしている。8 倍～12 倍に最適値が有りそう。
(FP16 だとこれより大きくすると精度が足りない。(学習率) x (バッチサイズ) x (マクロバッチ倍率) が一定値を超えると収束付近で学習効果が無くなる様に思う。)
　4）小さめの学習セットで 1 エポックの学習を数回実行し、最も D 値 (=Loss-(PolicyAcc+ValueAcc)*0.5)が小さくなった物を採用し以降の学習を行っている。初期値の分布の素性の良さそうな物を選ぶ。
　5）25 エポックまでに 3 回、入力に近い部分の ResNET を入れ替えて学習する。これにより収束を早める効果を狙っている。
　6）26 エポック以降も D 値をモニターし、D 値の減少が期待値以下の場合に、意図的にイレギュラーなデータを与える。データの与え方は 2 種類の手法を用意した。
　特定の次元が局所解に陥っている可能性を考慮しそこから脱出するのを期待している。
　7） D 値をモニターしていると収束に近づいているかどうかがわかるので D 値の挙動によって終了するエポックを決める。今回使用するモデルの場合約 100 エポックまで学習した。何エポックまで学習するかはモデルに依存する。

10. model の生成の工夫
　学習済のモデルは、最も Loss 値が低い、または最も Accuracy が高い物が勝率が高いモデルとは限らない。これは、Loss 値や Accuracy 値は出現頻度の高い特徴を持つ局面での正解率に強く依存し、出現頻度の低い特徴を持つ局面での正解率の寄与割合が低いためと考えた。将棋の勝敗は出現頻度の低い特徴の局面で正しい手を指せるかにも依存する。
　すべてのモデルでベンチマークを取るのを回避しつつ、この問題の影響を低減する工夫を行った。

11. 定跡

　手入力で定跡を用意した。思考時間の節約を目的としており、最長でも30 手程度と思う。

今回のモデルを用いて思考時間一手１分～５分で思考させる。２手目４通りの開始局面からベストラインを長く登録している。

ベストラインからの枝は短く評価値が(先手に)振れたら打ち切っている。

一本道の局面では打ち切らず、同程度の評価値の候補手が複数現れたら打ち切る。

-以上-