第３４回世界コンピュータ将棋選手権
「あすとら将棋」アピール文書

令和６年４月

恒岡 正年

1. 全体の構成

dlshogi の探索部、学習部を改造して利用。独自構造の model を採用。

2. 独自に実装した部分

Network 構造、学習方法、探索部の改造、model の生成の工夫等。

3. 開発動機

深層学習の勉強を兼ねて dl 系将棋 AI の model の学習を始めた。

WCSC33 に参加して決勝戦で全敗。上位ソフトと良い勝負をしたい。

4. 主な開発内容

独自構造の model の作成・学習方法

棋譜生成：5kpo～20kpo 程度の物を中心に千日手・持将棋の物を除いて 約 60 万個の棋譜作成

学習には、上記生成した棋譜以外に GCT の学習用データを利用

探索パラメータの調整(optuna 利用)

５. 定跡生成

前回は手入力の定跡だったが、今回は条件を満たす棋譜を集め定跡を生成した。

条件付きで選別した自己生成棋譜及び Floodgate の棋譜から作成した。

千日手の棋譜を含めるべきかどうかを検討した結果、Draw_Value_Black/White を調整すれば千日手の棋譜を含めなくても良いとの結論になった。

この方法は手軽な割に効果が大きい。定跡の有無でレートが R100 位変わる。

「2 段仕込み定跡」を作成した。これは、計算量と引き換えに上記の手法で作成した定跡に含まれている可能性のある悪手を排除する手法である。この手法の問題点は上記で登録された「対振り飛車定跡」が削除されてしまう点である。

「2 段仕込み定跡」と上記定跡を混ぜた「ブレンド定跡」も用意した。

対振り飛車には「ブレンド定跡」を使う予定であったがその機会はなく、全対局「2 段仕込み定跡」を使った。

6.思考時間制御
　手数と推定勝率に応じて思考時間を最大５倍まで変化させる様にした。

7.探索部
　Hybrid 探索
　２つの異なる学習を行った model(異なる極小値へ収束した可能性が高い)を用意し これらを A,B とする時 A=>A=>A または B=>B=>B の様に単独で使用し探索するよりも、A=>B=>A=>B の様に交互に探索する方が読み抜けが減る事を確認した。

V235(k=256,36b)と V252(k=384,24b)の２つの model を使って Hybrid 探索を行う。

　バッチサイズとスレッド数は、将棋 AI 用に用意した複数の次の一手問題を複数回解かせて、20sec の正答率が最も高くなる様に調整した。

8.高速化
　network 構造の工夫、及び GPU が rtx4090・rtx3090 の場合に最適化した探索部の工夫により、同じパラメータ数を持つ ResNET 構造の model と dl 将棋標準の探索部の組み合わせに対して約２倍の高速化を達成した。
　今回２次予選では model:V235 と model:V252 の Hybrid 探索を使用した。探索速度よりも探索精度を優先したパラメータを用いて約 44000nps 程度の探索速度だった。
(nps 優先の設定だと 50000nps 出る。)

9.予想レート
　以上の工夫により WCSC33 では R4200-R4300 程度だったが、今回は同じハードウェアで R+250 程度向上したと推定している。

以下は少し細かな内容

10. Network 構造
　　Network を次の３つに分ける。
　　(1)データ入力部　(2)ResNET 部　(3)結果の出力部(Dual Head 部)
　1）データ入力部
　　入力データの構成は標準と同じ。構造はオリジナルとは異なる。
　2）ResNET 部
　　標準の ResNET とは異なる構造を採用。（ 以下 AstraNET と称する。）
　　WCSC33 では、カーネル数を k, ブロック数を b とする時、(k=256,b=5) +
　　(k=224,b=20) の２段構造の通常の ResNET(合計 25 層)を使用した。
　　今回は ResNET を AstraNET に置き換えた。また 20b~36b まで種々のサイズの model を作成し強さを比較した。

k=256 で 20b 及び 24b の model では思考時間を延ばしても R4400-4500 のソフトに読み負ける事を確認。これよりも重い Network が必要。

今回は k=256,36b 及び k=384,24b の寸胴型の物を使った。NVIDIA 製 GPU のハードウェアの構造上 2 段構造にしても探索速度へのメリットは小さいと判断し寸胴型にした。

3）結果の出力部 (Policy/Value Network の分岐後)

全結合層の大きさを除いて dlshogi と同じ構造。

4）その他

AstraNET の活性化関数は主に ReLU を、全結合層には ELU を使っている。

(去年は全結合層に SiLU を使った)

11. 学習方法

1）学習率(lr)：0.2 から始めて 0.000003 まで等比数列的に減少させ学習した。減少させる割合は 0.80～0.95。d 値(=Loss-AverageAcc) の下がり方を見ながら調整した。

2）weight decay を 0.00010～0.00003 まで振って評価した。0.00003 は小さすぎる可能性がある。学習初期は大き目の物を使って学習終盤に小さくするのが良さそう。

3） 学習の途中で AstraNET のブロックの順序の入れ替えを適宜行っている。（去年も同様）

12. model の「追加工」

学習後に model の「追加工」をおこなった。

必ずしも Loss 値や d 値の最も小さい物、PolicyAcc,ValueAcc,Entropy の最も大きい物がベストな model とは限らない。

学習終盤では上記の指標を参考に良さそうな model を選択しベンチを実施。ベンチの結果の良かった複数の model のブレンド(例えば、3:2:1、100:1、100：-1 等)や乱数による揺らぎを与え派生 model を作成した。これらの中から最もベンチ結果の良かった物を最終的に採用した。

ベンチは棋譜生成も兼ねており次回の学習に使用する為 10k po~40k po で行った。

「追加工」により R+60 程度向上する場合も有った。

10. Optuna での探索パラメータの調整

去年の model は手作業である程度絞り込んでから Optuna を使った。

今回は 7 つある探索パラメータから以下の様なグループ A､B､C､D を作り、A=>B=>C=>A(不十分な場合 B=>C=>A を追加)=>D=>E の様に調整した。

1 回の Trial 数は 8～15 程度。全てを合計しても 100Trial は超えていない。

A: Softmax_Temp., root パラメータ 計 4 個

B: Softmax_Temp., non-root パラメータ 計 4 個

C: Softmax_Temp., C_fpu_reduction, C_fpu_reduction_root 計 3 個

D: 全パラメータ 計 7 個

E:Softmax_Temp., C_fpu_reduction, C_fpu_reduction_root を個別に振って妥当性を確認

一回の調整毎に上位 2~3 個のベンチを取り、ベンチ結果の最も良い物を次の中心値とした。Optuna での相手は Hao(3.5e6 nodes),ベンチでの相手は Hao(~6.0e6 nodes)と Tanuki-dr4(~5.0e5 nodes)を使った。評価される側の勝率が 50~65%になる様に po 数を調整した。1 Trial の games 数は 250 とした。

11.中終盤の棋力向上

WCSC33 では、80~100 手付近で読み負けるケースが多かった。その対策として以下の 3 点を行った。

1) 自己生成棋譜は、去年は意図的に 40～80 手を重複させ 30 手～120 手の局面を学習データとした。121 手以降は未使用であった。今回は、意図的な重複を排した 30～150 手の範囲の局面を使用した。

2) 学習終盤では全学習データ内の重複局面を削除して学習した。これにより序盤～中盤初期の棋譜の割合が減る。

全データを毎回全て学習するのは時間が掛かりすぎるため、学習時間が 12 時間を超えると次の学習条件に移る様にした。経験的に学習条件を変更した直後から 3 ～ 4 時間の間に有望な model が得られる事が多く、逆に 8 時間以上同一条件で学習を続けても良い model が得られる事は少なかった。

3) 持ち時間の使い方を変更し、70～95 手の思考時間を延ばした。また思考時間を延長する場合、従来 2 倍までだった物を条件によって最大 4 倍（元の時間と合わせて 5 倍）まで延長するようにした。

-以上-