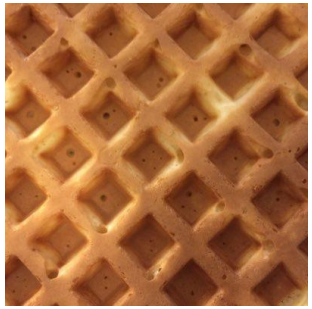
HoneyWaffle

第34回
世界コンピュータ将棋選手権
アピール文書
開発者　渡辺 光彦

開発者



氏名: 渡辺 光彦

職業: プログラマー

棋力: 将棋ウォーズで2級、ぴよ将棋でR900-1000程度の振り飛車党

Twitter: @shiroi_gohanP (https://twitter.com/shiroi_gohanP)

ニコ生の電王戦をきっかけにコンピュータ将棋を始める。

将棋連盟Liveやニコニコ生放送、AbemaTVの将棋中継が好き。

note書いています！ → https://note.com/honeywaffleshogi

文春オンラインのインタビュー記事 → https://bunshun.jp/articles/-/14921

HoneyWaffle (ハニーワッフル) 名前の由来（去年と同じ）

・四角いワッフルは将棋盤と似ている

・ゆるふわスイーツ的なスナック感覚の軽さを表現

　元々タブレット向けに開発していたので物理的に軽いこと、振り飛車の軽い捌きができるようになるといいなという想いから命名しました。

　コンピュータ将棋といえば、人名 +将棋と命名するのが格調高いと思っています（森田将棋とか）。私が有名ではなく、将棋界で渡辺といえば渡辺明先生なので、「渡辺将棋」とは命名すべきではないと思いました。ということで、渡辺がだめなら光彦 →みつ→Honey、Waffleは上記のとおり将棋盤の意味。 いい命名じゃないです か？ すごくないですか？

以下のリンク先で出せるものは公開しています。使い方がおかしいのはいつものこと。

https://github.com/32hiko

コンセプト

青字は使用予定ライブラリ

「毎日毎日練り続けた振り飛車定跡でいい将棋をお見せしたい」

（1）振り飛車定跡

ほぼ毎日4時間程度、floodgateでの実戦譜をもとに定跡を作成する作業をしています。複数台のミニPCで参戦していますが、検討にはAWS EC2のm6a.48xlargeを使用しています。1日約60局前後の棋譜を検討すると月に10万円以上かかります。（昨年8月からなので、高いマシンを買うのと同じくらいの出費が…）前回大会では先手は三間飛車、後手は四間飛車が中心でしたが、先手では初手56歩系、後手では角交換系の定跡を追加しています。

（2）評価関数とエンジンはBLOSSOMとやねうら王7.61を使用

無償で使用できる最高クラスの評価関数とエンジンを使用。（有償化に反対する意図は全くありません、念のため） こちらも自分自身で開発したい思いもありますが、全然手が回っていません。

（3）マシンは定跡の作業でも使っているm6a.48xlargeを使用（1台）

最後に

コンピュータ将棋では振り飛車の評価値が低く出る傾向なのは以前から言われていますが、飛車を振って下がった評価値をそれ以上下がらないようにキープすることは、評価値の割には難しくありません。（相居飛車で100の差がついたら大変なイメージがありますが、振り飛車ではそうでもない実感があります）

また、振り飛車を指すことで相居飛車の深い研究（特に先手が必勝に近い戦型）を必ず回避でき、その上で逆にこちらの深い研究に誘導できるのはメタ的に無視できない利点だと思います。

楽しくやれるうちはコンピュータ将棋と振り飛車を続けていくつもりですが、仮に今大会が最後になってしまっても悔いのないようにがんばります。

第 34 回世界コンピュータ将棋選手権 参加ソフト

ねね将棋 アピール文書

日高雅俊

2024/04/29

概要

iPhone 上で機械学習専用チップを活用し、深層学習ベースの将棋 AI を高速に動作させる。最新の iPhone 15 Pro を採用し、過去に使用していた iPad(第 9 世代)の約 5 倍の計算速度が得られる。探索速度は約 5000NPS (nodes per second)である。Floodgate でのレートは 3712（26 勝 24 敗、2024 年 4 月 29 日時点)。

図 1. 動作画面

技術要素

図 2. ソフトウェアの構造

ねね将棋は iPhone 上で動作するソフトウェアである。構造を図 2 に示す。GUI は、対局中

の盤面の表示や、対局サーバへの接続設定の受付を行う。UI フレームワーク SwiftUI により実装されている。通信部は、現在の盤面を把握し、思考部と USI プロトコルで通信し、対局サーバと CSA プロトコルで通信する。Swift 言語で実装された盤面表現を用いて、プロトコルの変換を行っている。思考部は指し手を決定する機構であり、C++言語で実装されたふかうら王（やねうら王の一種で、深層学習系評価関数を用いたモンテカルロ木探索機構を有する）を、iOS プラットフォーム向けに独自ビルドしたものを用いる。通信部との連携は、C++言語の標準入出力（通常、将棋所等とのプロセス間通信の手段となる）を Swift 言語のコールバック関数呼び出しに置換することによりプロセス内で完結させる。Core ML アダプタは、評価関数である深層学習モデルをふかうら王から呼び出すための機構である。Core ML は、Apple が提供する深層学習モデルの実行エンジンであり、機械学習専用チップ Neural Engine を利用することでモデルを高速に実行可能とする。Core ML アダプタは、Objective-C++で実装され、ふかうら王の一部としてビルドされる。評価関数には、dlshogi 系評価関数を使用（書籍「強い将棋ソフトの創りかた」サンプルコードにより学習した 20 層 192 チャンネルの CNN）する。PyTorch で学習されたモデルを、Core ML で使用される Apple 独自のモデル形式に変換して用いる。

対局サーバとの通信では、会場に用意される有線 LAN（イーサネット）を用いることで無線と比べ信頼性の向上を図る。iPhone を有線 LAN に接続することは一般的ではないが、USB 接続のイーサネットアダプタ ETX3-US2 (I-O DATA)により実現できる。

ねね将棋は iPhone の計算能力を最大限活用するものであるため、稼働中の発熱が大きい。長時間の対局を安定させ速度低下を避けるため、外付けファンによる強制空冷を実施する。

図 3. iPhone にファンを取り付けた状態

過去との差分

最新の iPhone 15 Pro を採用。ソフトウェアは、Swift 言語のみで思考部含め実装した WCSC32 と、ふかうら王を iOS 向けにビルドした WCSC33 (ただし USI と CSA の変換は Mac 上で動作する将棋所を利用)の実装を統合し、USI と CSA の変換機構を新規実装。

使用予定ライブラリ

- ふかうら王（やねうら王の一種。深層学習系評価関数により主に機械学習専用チップ Neural Engine で思考）
  - dlshogi 系評価関数を使用（書籍「強い将棋ソフトの創りかた」サンプルコードにより学習した 20 層 192 チャンネルの CNN）

ソースコードを公開しています。
https://github.com/select766/NeneShogi2024
各技術要素の詳細は、ブログをご覧ください。
https://select766.hatenablog.com/archive/category/%E3%82%B3%E3%83%B3%E3%83%94%E3%83%A5%E3%83%BC%E3%82%BF%E5%B0%86%E6%A3%8B

iPhone で将棋 AI を動かすノウハウ

単に将棋が指せるソフトができたあと、大会で安定して動作させるためのノウハウです。

- 機内モード・おやすみモードにする
  - 電話が着信すると困る。
- 有線 LAN へ接続する
  - USB 接続できる有線 LAN アダプタが存在する。
- プロビジョニングプロファイルの有効期限に注意
  - App Store で配布していない独自アプリは、一定期間で起動不能となる。
- 端末を冷却する
  - 端末が熱くなると計算速度が低下する。ゲーム向けの冷却ファンが使える。
- 充電器への接続を確認する
  - 接続忘れを見落としやすい。

第34回世界コンピュータ将棋選手権 アピール文書
プログラム名「タンゴ」
開発者　渡邊敬介
2024年3月30日

概要

実現確率探索による深い読みと、機械学習により最適化された正確な評価関数により、強力なコンピュータプレイヤの実現を目指します。

本プログラムの大きな特徴は、局面評価関数および実現確率用の着手確率にFactorization Machinesを使用している点です。2023年以前のコンピュータ将棋選手権でこのような手法を用いていたチームは、私の知る限りでは私のチーム以外に存在しません。

局面評価関数

本プログラムでは、駒の損得以外に局面評価関数の特徴量に下記の3つを採用しています。先後の対称性を保つため、先手から見た盤面と後手から見た盤面それぞれについて下記の特徴量を入力とするFactorization Machineによりスコアを計算し、算出された先後のスコアの差と駒の損得を合算してその局面の評価値としています。

1. 2駒の位置関係 (俗に言うKP + PP + KK)
2. 自玉周辺25マスの双方の効き
3. 手番

Factorization Machinesを使用することで、これらの入力特徴の組み合わせ特徴を取り込んだ極めて表現力の高い評価関数となっていると考えています。例えば、2駒の位置関係同士の組み合わせは4駒の位置関係(俗に言うKKPP + KPPP + PPPP)に相当します。

実現確率探索用の着手予測

Factorization Bradley-Terry モデル[1]を使用しています。通常のモデルと小規模で高速な簡易計算用モデルの2種類を用意し、探索中の末端付近ではこの簡易計算モデルを使用することで高速化を図っています。

また、Factorization Bradley-Terry モデルでは着手確率を計算するためにはその局面の全ての着手のスコアを計算した上でソフトマックス関数に通す必要があります。そこで、「駒の位置」や「王手がかかっている」などの盤面共通の特徴量を最初に計算し、各着手のスコア計算で再利用できるようにしています。さらに、駒の位置については差分計算可能なので、簡易計算モデルで使用される駒の位置の特徴量については差分計算を行っています。

追試可否

再現実験を可能とするため、対局用プログラムだけでなく学習に使用したデータセットも大会後1年間保管する予定です。

参考文献

[1]Xiao, Chenjun, and Martin Müller. "Factorization ranking model for move prediction in the game of Go." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 30. No. 1. 2016.

第34回世界コンピュータ将棋選手権

Ari Shogi and フレンズ　アピール文書

兵頭優空

概要

Ari Shogi and フレンズは、DL系のAI「Ari Shogi」とNNUE系のAIを組み合わせる事で「比較的低コストであらゆる状況に対応できるAI」を目指しているハイブリッド型の将棋AIです。

(12月末に今まで使っていたノートPCが壊れたので)1月末から新しいPCに変わり、計算資源にかなり余裕ができたので、NNUEの学習など今まで計算資源の都合で取り組めなかった事にも取り組みつつ、良い成績を目指したいと思っています。

目次

名前について

「Ari Shogi and フレンズ」という名前の「Ari Shogi」の部分は牡羊座(Aries)から、「and フレンズ」の部分は、複数のAIで協力するイメージからつけています。

牡羊座から名前を付けたのは、「自分の星座が牡羊座だから」と「某電気羊のように、何回か大きく進化していつか電竜になってほしいという願いを込めている」の2つの理由があります。(2つ目は後付けです)

アピールポイントなど

・DL系のAIの「終盤が弱い」など弱点を単体で克服するのを放棄し、苦手な部分はNNUE系AIに任せます。

・DL評価関数では、精度を上げるために複数の評価関数を使ってアンサンブルを行います。

第4回電竜戦(以降、DR4)でAri Shogi and フレンズは、「複数の評価関数の出力の平均を1つの評価関数の出力として扱う」という機能を大会直前に実装し、「自分が学習させた中で精度の高いモデル上位3つ」を搭載して参加していました。

この機能は、

1. かなり前から「複数の評価関数で行えば(速度は大きく落ちるものの)簡単に精度を上げられそう」という事を考えていた(実際に手を付けることはなかった)

2. 11月に「夏頃から新しく使い始めた入力特徴量」に大きなバグが見つかった。その特徴量を使って学習している途中だったモデルは良い感じの精度まで上がっていたが、結構酷いバグだったので新しいモデルを1から学習させる事になった。

3. 2のせいで予定が狂い、大会で使うモデルの精度が予定よりもずっと低くなってしまった。(これ関連で入玉モデル、進行度モデルの学習もほとんど行えなかったので、それらの出力による合議のモード切替は全然上手くいかなかった)

4. 大会直前に「今から精度を上げる方法はないか」と考えていたところ、アンサンブルの件を思い出したので、それ関連で実装が1番簡単そうだった「出力を平均する」というのを実装した。(当時は学習はGoogleColab無料版などの「ネットで無料で使えるGPU」に頼ってい

たが、使える分を全部使いきってしまった後だったのでそんなにできる事がなかった。当時のメインマシンのノートPCにはGPUが乗っていたが、GPUに負荷をかけるとバッテリー周りの挙動が怪しくなるようになっていたので、GPUに負荷が猛烈にかかる学習は怖くてできなかった。)

という経緯で実装された急場しのぎのものでしたが、評価関数の精度が(少しだけですが)ちゃんと上がり、デメリットの速度低下も

・探索部がPythonで書かれているので、C++で書かれているものに比べると元々かなり遅い

・(速度低下が1番悪影響を及ぼしそうな)中終盤はNNUE系AIに丸投げしている

・最善手をNNUE系AIでチェックし、NNUE系AIが考える最善手との評価値から大きく悪化している場合は、NNUE系AIの最善手を指すという機能がある

といった理由であまり気にならなかったので、割と上手くいった改造だったと考えています。

WCSC34バージョンのAri Shogi and フレンズは、DR4バージョンで上手くいった方針を発展させ、より強力な手法を採用等に取り組み、精度のさらなる向上を目指します。

また、アンサンブルに使うモデルはある程度多様性があったほうが良いと考えているので、今使っているResNet系以外の系統のモデルも学習させてアンサンブルに使う予定です。

(3月末の時点では、
https://tokumini.hatenablog.com/entry/2021/09/04/120000

を参考にVision Transformer(ViT)を実装するところまで終わっているが、計算資源が足りなくて学習は回せていない)

・NNUE評価関数を自作の学習部で学習させます

DR4までは、「NNUEの学習もやってみたいなー」とは思ってはいたものの、リソースの都合等から取り組む事はなく、公開されているNNUE評価関数をそのまま利用していましたが、今回からは自分で学習させた標準型のNNUE評価関数を使用する予定です。

学習については、普通に学習させるだけでなく色々な実験(DLモデル

からの蒸留とか)をしたいと考えています。(が、リソースの都合であまりできなさそう)

使い慣れたプログラムをベースにしている方が効率が良く実験できると考えたので、実験用プログラムのベースとしてpython-dlshogi2の学習部をベースにしたNNUE学習部を製作しました。

学習部は公開してほしいという要望があったため、テストで作った振り飛車評価関数と学習ログと一緒に
https://github.com/YuaHyodo/Ari_Shogi_NNUE_train

https://github.com/YuaHyodo/Haojian_nnue

で公開しています。(厳密には、非公開のものから分岐させて個人用のメモとかを消した後のものなので同じものではないです。また、元々公開する予定のなかったものなので、超絶汚いです)

上の学習部(のもとになった非公開の学習部)でHáoを水匠1000万ノードのデータで1epochだけ追加学習した"Haojian_240317"という評価関数が、CPU: i5-13400F, 探索部: やねうら王7.50,1スレッド, ハッシュ512MB, たややん互角局面集24手目～という設定で、

```
///
VS Háo / 1手1000ms
対局数5000 先手勝ち2606(52.7%) 後手勝ち2336(47.3%) 引き分け58
engine1
勝ち2558(51.8% R12.1 +-9.6) 先手勝ち1349(27.3%) 後手勝ち1209(24.5%)
宣言勝ち8 先手宣言勝ち3 後手宣言勝ち5 先手引き分け33 後手引き分け25
engine2
勝ち2384(48.2%) 先手勝ち1257(25.4%) 後手勝ち1127(22.8%)
宣言勝ち3 先手宣言勝ち2 後手宣言勝ち1 先手引き分け25 後手引き分け33
///
VS Háo / 1手5000ms
対局数5000 先手勝ち2478(50.0%) 後手勝ち2475(50.0%) 引き分け47
engine1
勝ち2630(53.1% R21.4 +-9.7) 先手勝ち1315(26.5%) 後手勝ち1315(26.5%)
宣言勝ち15 先手宣言勝ち10 後手宣言勝ち5 先手引き分け21 後手引き分け26
engine2
```

勝ち2323(46.9%) 先手勝ち1163(23.5%) 後手勝ち1160(23.4%)

宣言勝ち6 先手宣言勝ち3 後手宣言勝ち3 先手引き分け26 後手引き分け21

///

VS BLOSSOM / 1手1000ms

対局数5000 先手勝ち2486(50.2%) 後手勝ち2470(49.8%) 引き分け44

engine1

勝ち2914(58.8% R61.2 +-9.8) 先手勝ち1461(29.5%) 後手勝ち1453(29.3%)

宣言勝ち22 先手宣言勝ち13 後手宣言勝ち9 先手引き分け22 後手引き分け22

engine2

勝ち2042(41.2%) 先手勝ち1025(20.7%) 後手勝ち1017(20.5%)

宣言勝ち1 先手宣言勝ち0 後手宣言勝ち1 先手引き分け22 後手引き分け22

///

VS BLOSSOM / 1手5000ms

対局数5000 先手勝ち2597(52.5%) 後手勝ち2346(47.5%) 引き分け57

engine1

勝ち2876(58.2% R56.7 +-9.8) 先手勝ち1502(30.4%) 後手勝ち1374(27.8%)

宣言勝ち9 先手宣言勝ち4 後手宣言勝ち5 先手引き分け30 後手引き分け27

engine2

勝ち2067(41.8%) 先手勝ち1095(22.2%) 後手勝ち972(19.7%)

宣言勝ち1 先手宣言勝ち0 後手宣言勝ち1 先手引き分け27 後手引き分け30

///

VS 水匠5 / 1手1000ms

対局数5000 先手勝ち2580(52.5%) 後手勝ち2331(47.5%) 引き分け89

engine1

勝ち2548(51.9% R12.9 +-9.6) 先手勝ち1338(27.2%) 後手勝ち1210(24.6%)

宣言勝ち56 先手宣言勝ち30 後手宣言勝ち26 先手引き分け47 後手引き分け42

engine2

勝ち2363(48.1%) 先手勝ち1242(25.3%) 後手勝ち1121(22.8%)

宣言勝ち1 先手宣言勝ち0 後手宣言勝ち1 先手引き分け42 後手引き分け47

///

VS 水匠5 / 1手5000ms

対局数5000 先手勝ち2515(51.1%) 後手勝ち2402(48.9%) 引き分け83

engine1

勝ち2596(52.8% R19.1 +-9.6) 先手勝ち1323(26.9%) 後手勝ち1273(25.9%)

宣言勝ち56 先手宣言勝ち29 後手宣言勝ち27 先手引き分け34 後手引き分け49

engine2

勝ち2321(47.2%) 先手勝ち1192(24.2%) 後手勝ち1129(23.0%)

宣言勝ち2 先手宣言勝ち1 後手宣言勝ち1 先手引き分け49 後手引き分け34

///

計測はTanuki Coliseum(https://github.com/nodchip/TanukiColiseum / 使ったのは古いバージョン)で行った。

いずれもengine1がHaojian_240317

FV_SCALEは、水匠5とBLOSSOMが24、Háoが20、Haojian_240317は軽く計測して1番強そうだった値に設定。

対BLOSSOMは設定をミスっているのか、相性の問題なのか、開始局面の問題なのか、異様に成績が良い。

という結果になっていて、floodgateでの計測でもHáoより少しだけレートが高いという結果で、少なくとも弱くはなっていさそうなので学習部に致命的レベルのバグはないと思います。多分

Haojianという名前は、ベースになっているHáoと、某ゲームに出てくる災いの剣からつけています。

・自動定跡生成時の局面評価で、「方策出力のみで指すDL系AIによる連続対局の結果」と「浅い探索のNNUE系AIによる連続対局の結果」と「長い時間探索したAIによる連続対局の結果」を組み合わせる予定です。

「方策出力のみで指すDL系AIによる連続対局」には「GPUを用いる事でそこそこの棋力で同時に何千局も行える」という大きなメリットがあるため、定跡生成や教師データ生成の一部で使用できると考えています。

詳細は未定ですが、

「方策出力のみで指すDL系AIによる連続対局の勝率」と「浅い探索の

NNUE系AIによる連続対局の勝率」が大きく違なる場合、局面の難易度が高いと判断し「長い時間探索したAIによる連続対局」を多めに行う

といった事も行いたいと考えています。

本番までに余裕があったらやる事

・持ち時間管理部を改造する

形成に大きな差がついてからは持ち時間に差があっても逆転する事は難しいと思うので、それを念頭に置いた持ち時間管理を行いたいと考えています。

詳細は決まってないですが、DR4バージョンの序盤モードの持ち時間管理をベースに作ろうと今のところは考えています。

[DR4バージョンの序盤モードの持ち時間管理(ざっくり)]

1.事前に決めておいた「中終盤のために残しておく時間」を現在の残り時間から引く

2.「現在の局面の終局までの推定手数」から、事前に決めておいた「終局までの推定手数がこの値以下になった時に中終盤モードに移行する値」を引いたものを「中盤までの推定手数」とする

3.残り時間を「中盤までの推定手数」で割ったものに、加算時間や秒読みを足したものを「今の手番で使う時間(のベース)」に設定

4.一定の条件を満たした場合は予定の時間より早く打ちきったり、逆に延長したりする。(時間は余裕を持って計算しているので足りなくなる事はあまりない)

ちなみに、本番では「終局までの推定手数」の精度が低かったせいで思うように行きませんでした。

~~・局面生成AIを教師データ生成などに応用する~~

~~DR4では簡単な局面を生成できるところまではいったものの、色々あってGPU資源がなくなってしまったために結局応用する事はできませんでした。~~

~~いつかは覚えてないですが、DR4のアピール文書~~
~~(https://drive.google.com/file/d/16P0Gcv4K0yo8v4GGH2zo2QQ290EnWeHd/view)に書いていた「評価関数の弱点の克服方法」について、より良い案を思いついたので余裕があったら実装したいと考えています。~~

=> 間に合いそうにないので今回はやりません。

ただ、個人的には局面生成AIにゼロからの強化学習くらいロマンを感じるので、いつかリベンジする予定です。

使用予定のライブラリ

・python-dlshogi2(https://github.com/TadaoYamaoka/python-dlshogi2)

シンプルな構造でそれなりに強く、かつ使い慣れているため、Ari ShogiやNNUE学習部のベースとして採用しています。

・やねうら王(https://github.com/yaneurao/YaneuraOu)

自分が知っている「NNUEが利用できる探索部」のなかで最強なので、NNUE系の探索部として採用しています。

・dlshogi(https://github.com/TadaoYamaoka/DeepLearningShogi)

探索部のパラメータ調整スクリプトや、教師データの変換スクリプトが便利なので、その部分だけ利用しています。

・KomoringHeights(https://github.com/komori-n/KomoringHeights)

USIプロトコルで利用できて、かつ性能が良さそうなので、詰め将棋エンジンとして採用しています。

使用予定のデータ / 公開モデル

・AobaZeroのデータ(http://www.yss-aya.com/aobazero/)

・floodgateのデータ(http://wdoor.c.u-tokyo.ac.jp/shogi/index.html)

・水匠1手200万手のデータ
(https://drive.google.com/file/d/1R9kI3xDKeoIjyFPDORS6K-1wwck

o75fr/view ）

・水匠1000万ノードの公開データ
（ https://drive.google.com/file/d/1VyP4MX_AuQhvy8sesymgPVf9sUnQoGPl/view ）

・dlshogi_with_GCTのデータ
（ https://tadaoyamaoka.hatenablog.com/entry/2021/05/06/223701 ）

・書籍「強い将棋ソフトの創りかた」のデータ

いずれのデータも、質も量も十分以上なため採用しています。

・Qhapaq Pretty Derbyのデータ
（ https://qhapaq.hatenablog.com/entry/2021/11/23/220251 ）

採用理由：振り飛車評価関数を簡単に作れるから

振り飛車をコンセプトにするわけではないですが、振り飛車評価関数を一部で採用する予定なので、その振り飛車評価関数の学習に利用します。

~~・shogi_suisho5のデータ_depth9（ https://huggingface.co/datasets/nodchip/shogi_suisho5_depth9 ）~~

~~・shogi_hao_depth9のデータ（ https://huggingface.co/datasets/nodchip/shogi_hao_depth9 ）~~

~~ゼロからNNUEを学習させる事ができる量/質である事が分かっているデータなので、ゼロからNNUEを学習させる場合は採用します。~~

SSDの容量が足りないし、ゼロからNNUEを学習するのを断念したので多分使いません。

・Háo（ https://github.com/nodchip/tanuki-/releases/tag/tanuki-.halfkp_256x2-32-32.2023-05-08 ）

・BLOSSOM（ https://twitter.com/senninha_a/status/1654675205464395

9872 ）

・水匠5(https://github.com/mizar/YaneuraOu/releases/tag/v7.5.0 ）

・Lí(https://github.com/nodchip/tanuki-/releases/tag/tanuki-wcsc33-2023-05-04 ）

~~ゼロからのNNUE評価関数を学習させるのを断念した場合は、NNUE評価関数のベースとして採用する予定です。(単体でベースにする以外にも、キメラ化したものをベースにする可能性もある ）~~

~~特にHáoは、3月末の時点で1番強い評価関数(Háoよりレート10～20ほど強い)のベースになっているので、採用される確率が1番高いです。~~

採用理由：強いから

3月末の時点の状況だと、NNUEをゼロから学習させるのは無理そうなのでこれらの評価関数を組み合わせて作ったキメラ評価関数をベースにする予定です。

標準型NNUEを作るのに標準型以外のNNUEを素材として使用しり予定になっているのは、ネットワーク構造がある程度違うNNUEを使ったキメラも作れる方法を採用する予定だからです。

その他

・本当は会場に行って参加したいですが、家から遠いので今年もオンライン参加です。

・アピール文書は後で追記したものに差し替える予定です

以下、おまけなので文章とかより一層雑です

おまけ1：モデル精度について色々

モデル名	パラメータ数	方策正解率	価値正解率	備考
WCSC33のモデル	1687306	0.443736	0.7185835	テストデータの一部が学習データに含まれている
23年11月モデルA	3213186	0.445989	0.7198876	テストデータの一部が学習データに含まれている
23年11月モデルB	3213186	0.4552351	0.7279302	テストデータの一部が学習データに含まれている
DR4で使ったアンサンブルモデル	8113678	0.463778	0.7344486	WCSC33のモデル + 23年11月モデルA + 23年11月モデルB
SplatBrella_231219のアンサンブルモデル	12296782	0.475275	0.7415995	WCSC33のモデル + 23年11月モデルB + python-dlshogi2のモデル
WCSC34に向けて育成中のモデルの1つ	7164218	0.463992	0.727845	学習途中でまだまだ伸びそう
python-dlshogi2のリポジトリのモデル	7396290	0.4785397	0.7409203	
GCT電竜	?	0.46163161	0.73495564	データは https://tadaoyamaoka.hatenablog.com/entry/2021/09/11/155938 より引用
dlshogi with GCT	?	0.48964214	0.75278598	データは https://tadaoyamaoka.hatenablog.com/entry/2021/09/11/155938 より引用
dlshogi dr2_exhi	?	0.52322546	0.76564239	データは https://tadaoyamaoka.hatenablog.com/entry/2021/09/11/155938 より引用
tanuki-wcsc28	32117089	-	0.66378727	方策出力が無いので価値正解率のみ
水匠5	32117089	-	0.68430669	方策出力が無いので価値正解率のみ
BLOSSOM	32117089	-	0.68488171	方策出力が無いので価値正解率のみ
Háo	32117089	-	0.68491119	方策出力が無いので価値正解率のみ

・データはGCTの公開ノートブック
(https://tadaoyamaoka.hatenablog.com/entry/2020/11/26/203912)で使われているテストデータと同じ

・パラメータ数は
https://rheinmetall.hatenablog.com/entry/2021/05/14/000000 のコードで測定

・「23年11月モデルA」は「入力特徴量がバグっていたのに気づいて途中で学習を止めたモデル」

・「python-dlshogi2のリポジトリのモデル」は
https://github.com/TadaoYamaoka/python-dlshogi2/blob/main/checkpoints/checkpoint.pth のモデルの事。精度を比較したり、アンサンブルの効果を確かめたりするために利用している。

・比較のために、GCT電竜、dlshogi with GCT、dlshogi dr2 exhiのデータを https://tadaoyamaoka.hatenablog.com/entry/2021/09/11/155938 から引用している。

・昔の記録を発掘して、そこに書いてある数値を確認せずにコピペしているだけなので、間違っている可能性がある

・SplatBrella_231219は、モデルとパラメータを軽く弄って(ついでにいらない機能をOFFにして)floodgateに放流していた Ari Shogi and フレンズ。ペタショック定跡に(有利な先手番でだが)1発入れた時のもの。

http://wdoor.c.u-tokyo.ac.jp/shogi/view/2023/12/19/wdoor+floodgate-300-10F+SplatBrella_231219+YOV800-peta4M-3700X+20231219190001.csa

SplatBrellaという名前は、某イカのTPSゲームで私がここ数年愛用している傘の英名からとっています。

[感想とか考察とか]

・DR4のモデルでは、単体での精度が1番高いモデルから正解率が方策/価値ともに上昇している。探索速度は落ちたものの、DR4での構成では探索速度の低下によるデメリットをあまり感じなかったので、大会直前にこれを実装したのは成功だったと思う。

・SplatBrellaのモデルは、単体での精度が1番高いモデルからほとんど変わっていない。(むしろ若干下がっている)

1番精度の高いモデルと2番目以降のモデルの精度に大きな差があったので、足を引っ張ってしまったのかもしれない。(が、より複雑な方法でアンサンブルを行った場合は、これでも精度が上がる可能性があると自分は考えている。この辺は余裕があったら実験する。)

=>(追記)「1番精度の高いモデルと2番目以降のモデルの精度に大きな差があったため、足を引っ張って精度が上がらなかった」という説を確かめるための実験を行った。

モデル名	パラメータ数	方策正解率	価値正解率	備考
WCSC33のモデル	1687306	0.443736	0.7185835	テストデータの一部が学習データに含まれている
23年11月モデルA	3213186	0.445989	0.7198876	テストデータの一部が学習データに含まれている
23年11月モデルB	3213186	0.4552351	0.7279302	テストデータの一部が学習データに含まれている
DR4で使ったアンサンブルモデル	8113678	0.463778	0.7344486	WCSC33のモデル + 23年11月モデルA + 23年11月モデルB
SplatBrella_231219のアンサンブルモデル	12296782	0.475275	0.7415995	WCSC33のモデル + 23年11月モデルB + python-dlshogi2のモ
WCSC34に向けて育成中のモデルの1つ	7164218	0.463992	0.727845	学習途中でまだまだ伸びそう
python-dlshogi2のリポジトリのモデル	7396290	0.4785397	0.7409203	
GCT電竜	?	0.46163161	0.73495564	データは https://tadaoyamaoka.hatenablog.com/entry/2021/09/11/155
dlshogi with GCT	?	0.48964214	0.75278598	データは https://tadaoyamaoka.hatenablog.com/entry/2021/09/11/155
dlshogi dr2_exhi	?	0.52322546	0.76564239	データは https://tadaoyamaoka.hatenablog.com/entry/2021/09/11/155
tanuki-wcsc28	32117089	-	0.66378727	方策出力が無いので価値正解率のみ
水匠5	32117089	-	0.68430669	方策出力が無いので価値正解率のみ
BLOSSOM	32117089	-	0.68488171	方策出力が無いので価値正解率のみ
Háo	32117089	-	0.68491119	方策出力が無いので価値正解率のみ
WCSC34に向けて育成中のモデルの1つ(その後)	7164218	0.4717394	0.7350589	伸びしろはもうそんなに残ってなさそうだが、まだ少しは伸び
2024年3月30日の実験モデル	14560508	0.4842954	0.7458868	pydlshogi2のモデル + 育成中のモデル(その後)

(表の赤色のところが新しく調べた場所)

「2024年3月30日の実験モデル」は、「WCSC34に向けて育成中のモデルの1つ(その後)」と「python-dlshogi2のリポジトリのモデル」の2つのモデルの出力を単純に平均しただけのものだが、ちゃんと正解率が上昇している。

なので、「SplatBrellaのモデル精度が、python-dlshogi2のモデル単体からほとんど上がらなかったのは、他のモデルが足を引っ張っていたから」という説は多分あっている。

ただ、あるモデルが"有望ではない"としてほぼ0%を出力した指し手でも、他のモデルがある程度の値を付けた場合は探索される可能性が出てくるため、実際に探索に組み込んだ際はテストデータでの精度以上に違いが出る(良い方向でも悪い方向でも)のではないか、と考えている。(試せてはない)

・DR4で使った3つのモデルは「2023年に作ったDL評価関数ランキングtop3」だが意外と精度が出ていない。

「WCSC34に向けて育成中のモデル」は、学習させ始めてから3日くらいしか経ってないのに、DR4のアンサンブルモデルと同じくらいの精度が出ている。(DR4のモデルはテストデータの一部が学習データに含まれているというハンデがあるので、実際の精度は育成中のモデルのほうが高い可能性もある)

(2024年1月の末にPCが新しくなるまでは)GoogleColab無料版などの「無料で利用できるGPU」を使ってチマチマと学習を進めていた(一応、メインのノートPCにはGPUが載っていたが、酷使しすぎたせいか、GPUに負荷をかけるとバッテリー周りの挙動が怪しくなるようになっていたので、あまり学習には投入できなかった)のが、新しくなってからはRTX4060Ti(メモリ16GB)がほぼ24時間フル稼働、と環境が大きく変わったのはあるが、それにしても「3日学習させたモデル >= 去年1年間の集大成ともいえるモデル」というのは少し残念。

おまけ2: DR4バージョンのAri Shogi and フレンズのfloodgateでの計測

ハードウェア含め本番と同じ構成で計測。対局数が少ないのでレートはあまり信頼できないが、少なくとも4000は超えていると思う。

asa35	4257	61	35	0.635	2023-08-08	4168
SV-037	4254	75	47	0.614	2023-09-03	4165
test_13900k	4252	39	15	0.712	2023-06-29	4163
b22sd	4242	45	30	0.598	2023-12-05	4153
ECLIPSE-20230523_R9-5900HX	4241	24	9	0.728	2023-07-25	4152
GloriousMoment	4240	22	5	0.799	2023-11-28	4151
FAREWELL_R9-5900HX	4234	221	136	0.619	on line	4145
Sagittarius	4232	104	68	0.603	2023-12-02	4143
VW201	4229	16	7	0.691	2023-06-18	4140
Ari_Shogi_and_Friends_DR4	4221	12	4	0.735	2023-12-05	4132
S.novi-belgii	4219	16	1	0.933	2023-10-15	4130
GGG	4217	52	29	0.640	2023-09-14	4128
ECLIPSE-20230717_R9-5900HX	4215	31	16	0.659	2023-07-25	4126
ECLIPSE_Ryzen9-5900HX	4214	26	14	0.648	2023-07-03	4125
dltest	4210	14	4	0.758	2023-10-17	4121
KANROJI_MITSURI	4209	31	15	0.662	2023-11-03	4120
Hao-s-book_black-i5-13400	4204	21	2	0.913	2023-12-06	4115
nnue-tanuki-dr3_5800u	4203	51	32	0.616	2023-11-22	4114
SV-q044	4200	222	36	0.859	2023-10-19	4111
B1112	4199	343	240	0.588	on line	4110
HoneyWaffle-37-003	4195	879	666	0.569	2023-12-02	4106

引用元：http://wdoor.c.u-tokyo.ac.jp/shogi/x/rating/players-floodgate-20231207.html

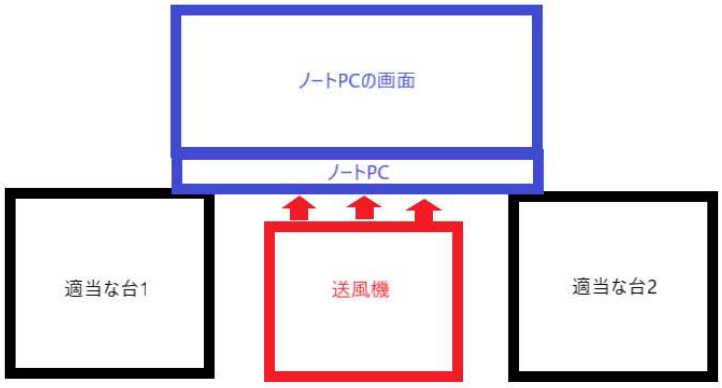
おまけ3. 第4回電竜戦本戦の感想とか

・大会前は「モデル精度が低い」「モード切替機能が微妙」「定跡がない」「探索部や合議パラメータを本来は連続対局の結果を元に自動調整する予定だったが、リソース不足でちゃんとできなかった」「無理な運用を続けたせいでハードウェアが劣化してきている」といった理由から、「あまり良い結果にはならなさそう」と予想していた

・特にハードウェアは、大会前/大会中に故障しないかヒヤヒヤしていた。長期間無理な運用をしてきたせいなのか、GPUに強い負荷をかけると「バッテリーが認識されなくなる」や「電源に接続されているはずなのにバッテリーで稼働中(電源に接続されていない)と表示される」といった現象が発生したし、比較的軽めの負荷でも長期間稼働させると「いつの間にか電源が落ちている」や「画面が真っ暗になり、一切の操作を受け付けなくなる(が、

本体は高温になっているしファンも稼働しているので恐らく電源はついている)」といった現象が発生した。

・大会中も、ずっと電源に接続しているのにも関わらず「バッテリーが検出されません」みたいな表示と「充電中」という表示が高速で切り替わる、という事が起きていた。多分そのせいで、大会中は下の画像のようにしてPCを冷やしていたのに、CPUのクロックがかなり落ちてしまっていた。



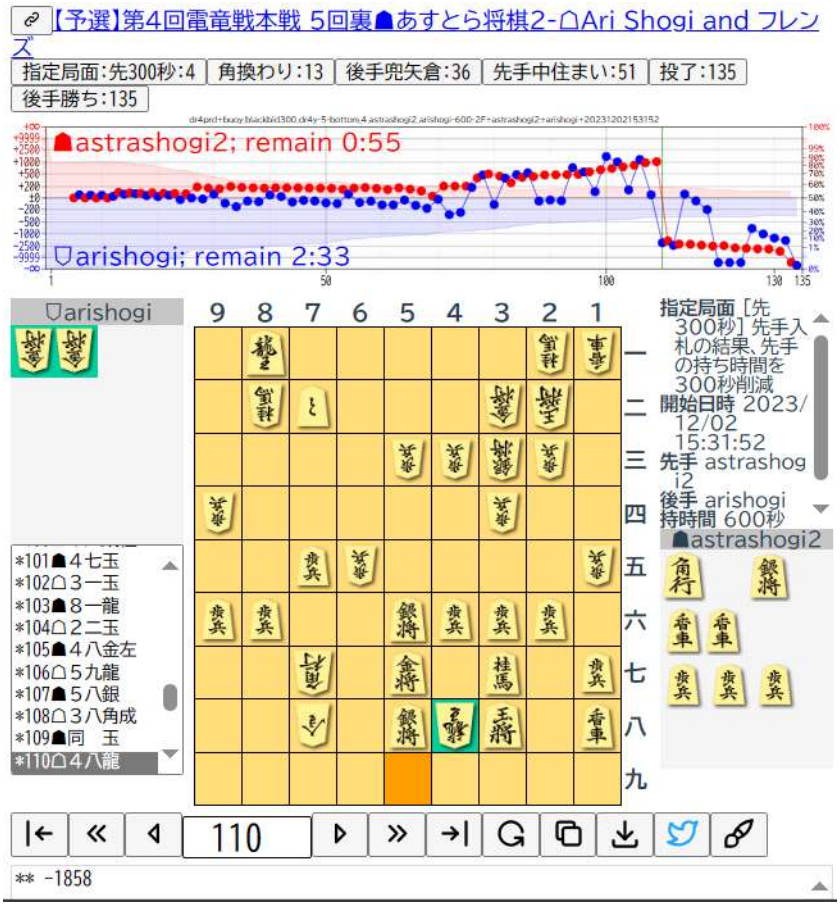
・大会で使ったPCは、大会後の12月の末に「動作中に異臭がする」という事が起きてしまったので封印する事になった。一応起動もできるし、データは無事で取り出す事もできるが、火事になるのは絶対に避けなければならないので、(データを取り出すなどの)理由がない限りは起動しないようにしている。

PCが壊れたせいで、1ヵ月くらい開発速度が大きく低下し、モチベーションも激減したが、「気づかないうちにPCから出火して家が火事になる」とか「大会前/大会中に壊れる」とか「(バックアップ取ってないのに)データが全部消える」とか、そういう事が無かったのでまだ良かったと思う。

・それでもB級に進出し、しかもB級最下位を回避したのは、非常に運が良かったからだと思う。(実際、C級上位陣や、2日目に参加しなかったQhapaqチームやアストラ将棋チームのほうが強いと思う)

WCSC33も強運で1次予選を突破していたし、2023年は将棋AIに関しては非常に運がよかった。(2023年の初詣で引いたおみくじの結果はたしか"凶"だったけど)

・個人的に1番嬉しかった勝ちは1日目5回裏の対あすとら将棋2戦。相手が格上だったので、5回表の先手千日手で満足していたのだが、終盤に相手が読み抜けしたので、2戦1.4勝0.6敗という想像もしなかったレベルの非常に良い結果に終わった。



引用元: https://denryu-sen.jp/denryusen/dr4_production/dist/#/dr4prd+buoy_blackbid300_dr4y-5-bottom_4_astrashogi2_arishogi-600-2F+astrashogi2+arishogi+20231202153152/110

・個人的に1番悲しかった負けは2日目の5回表の対ECLIPSE戦。後手番で600以上まで行ったのに、そこから逆転して負けた。

引用元：https://denryu-sen.jp/denryusen/dr4_production/dist/#/dr4prd+buoy_blackbid300_dr4b-5-top_4_eclipse_arishogi-600-2F+eclipse+arishogi+20231203141035/91

第34回
世界コンピュータ将棋選手権
なのはアピール文書

2024年3月31日 川端一之

■**なのは**ってなんだよ

- 熱血魔法バトルアクションアニメ「魔法少女リリカル**なのは**」シリーズの主人公**高町なのは**を由来にし、さまざまな称号を冠する彼女のような強さを盤上で実現したいという願いを込めています。

- よく「名前の割に強い」という声をいただきますが、その認識は逆で、「名前負けしている」や「名前の割に弱すぎる」というほうが妥当な評価です。

■作者はどんな人？

- 静岡県出身 愛知県在住
- とあるメーカーに勤務(ちょっとAWS使う)
- 好きな食べ物は焼肉、しゃぶしゃぶ、寿司
- 好きなアニメは魔法少女リリカルなのは、りゅうおうのおしごと！、痛いのは嫌なので防御力に極振りしたいと思います。、くまクマ熊ベアー、とある科学の超電磁砲、ラブライブ！、五等分の花嫁

  最近は「お隣の天使様にいつの間にか駄目人間にされていた件」がお気に
- 将棋ウォーズ 1級
- 囲碁は日本棋院 初段(アマチュア)

■開発環境

- こんなPCで開発しています

  Lenovo Thinkpad T14

  CPU：AMD Ryzen 7 PRO 4750U

  プレゼントでいただきました

- 出場PC

  Minisforum UM790Pro

  CPU：AMD Ryzen 9 7940HS

  RAM：32GB

  OS：Windows 11 Pro

■なのはの構成

- MSYS2のg++で開発
- 手生成では歩、角、飛の不成も生成
- 探索はStockfish使用
- Bitboard未使用(盤情報は配列)
- 定跡部は実戦での出現数および勝率を考慮して手を選択
- 評価ベクトルは3駒関係(KPPT)

…と、数年前に見たような平凡な構成

■今回の変更点

- 出場PCのパワーアップ(Ryzen 7 4750U → Ryzen 9 7940HS)
- [予定]探索部の強化(Stockfish 8改→ Stockfish 15.1改)
- [予定]評価関数の強化(配布教師データでの学習)

■意気込み

- 現地会場から参加！
- 0次予選突破！
- 出来れば勝ち越したい！
- 詰めルーチンを間に合わせたい！

■今後の野望(?)

- Ryzen AIの活用
- 評価関数のNNUE化
- ミクロコスモスを解く
- 非コピーレフト系ライセンスを採用して開発

■使用ライブラリについて

- なのはmini　Stockfishをベースに独自に将棋化しました。

■使用データについて

- 評価ベクトルの学習には、一般に流布された教師データを使う予定
  - やねうらおさん配布の110億教師データ
  - nodchipさん配布の教師データ

■最後に

- **なのは**アピール文書は以上です
- 最後まで読んで頂きありがとうございます

絵： COCOさん

■参考文献

- 小谷善行、他:「コンピュータ将棋」, サイエンス社, 1990.
- 松原仁 編著:「コンピュータ将棋の進歩」, 共立出版, 1996.
- 松原仁 編著:「コンピュータ将棋の進歩2」, 共立出版, 1998.
- 松原仁 編著:「コンピュータ将棋の進歩3」, 共立出版, 2000.
- 松原仁 編著:「アマ四段を超えるコンピュータ将棋の進歩4」, 共立出版, 2003.
- 松原仁 編著:「アマトップクラスに迫るコンピュータ将棋の進歩5」, 共立出版, 2005.
- 池泰弘:「コンピュータ将棋のアルゴリズム」, 工学社, 2005.
- 金子知適, 田中哲朗, 山口和紀, 川合慧:「新規節点で固定深さの探索を併用するdf-pnアルゴリズム」, 第10回ゲーム・プログラミングワークショップ, pp.1-8, 2005.
- 脊尾昌宏:「詰将棋を解くアルゴリズムにおける優越関係の効率的な利用について」, 第5回ゲーム・プログラミングワークショップ, pp. 129-136, 1999.
- 保木邦仁:「局面評価の学習を目指した探索結果の最適制御」 http://www.geocities.jp/bonanza_shogi/gpw2006.pdf
- 岸本章宏:「IS 将棋の詰将棋解答プログラムについて」, http://www.is.titech.ac.jp/~kishi/pdf_file/csa.pdf, 2004.
- 橋本剛, 上田徹, 橋本隼一:「オセロ求解へ向けた取り組み」, http://www.lab2.kuis.kyoto-u.ac.jp/~itohiro/Games/Game080307.html

■参考Web

- やねうら王 公式サイト: http://yaneuraou.yaneu.com/
- 千里の道も一歩から: http://woodyring.blog.so-net.ne.jp/
- 小宮日記: http://d.hatena.ne.jp/mkomiya/
- State of the Digital Shogics [最先端計数将棋学]: http://ameblo.jp/professionalhearts/
- ながとダイアリー: http://d.hatena.ne.jp/mclh46/
- 毎日がEveryday: http://d.hatena.ne.jp/issei_y/
- Bonanzaソース完全解析ブログ: http://d.hatena.ne.jp/LS3600/
- aki.の日記: http://d.hatena.ne.jp/ak11/
- FPGA で将棋プログラムを作ってみるブログ: http://blog.livedoor.jp/yss_fpga/

※読めなくなったサイト含む

Python-dlshogi2 に二つの重要な機能を追加しました。

定跡ファイルの読み込み: この機能を用いて、定跡に基づく手が選ばれると、プログラムは考慮せずに即座にその手を指します。これにより、定跡に沿った局面では迅速に手を進めることが可能になります。

詰め将棋エンジン「komoringheights」の統合: 詰め将棋エンジンが相手の王に詰みを発見すると、自動的に詰め将棋エンジンに切り替わり、指し手を行います。これにより、詰みの局面を効率的に活用することができます。

さらに、学習モデルに関しては、「強い将棋ソフトの創り方」の第 7 章に基づく学習データを使用したモデル「checkpoint.pth」に、Floodgate の棋譜を用いて追加学習を行いました。具体的には、2023 年度、2022 年度、2019 年度の棋譜を使って、モデルを 3 回学習させました。

[WCSC34]

きのあ将棋について　1

方針

→実行は低コストになるように

→局面単位のスポット思考ができるように

→設定ファイルで多態性を確保できるように

[WCSC34]

きのあ将棋について　2

いわゆるフルスクラッチ

→他将棋ライブラリ、学習データなど未使用

→棋譜のみ他のものを利用

→評価値や読み筋などは使わない

[WCSC34]

きのあ将棋について　3

おおよその構成

→探索　　：PVS、反復進化をカスタマイズしたシステム

→局面評価　：深層学習に影響を受けた独自多段階評価

→候補手評価：キレーティング（QR）によるパターン評価

特に「キレーティングによるパターン評価」に

今回から力を入れたい。

[WCSC34]

キレーティングによる候補手評価

- 候補手を各種パターンに分類
- その各種パターンごとにより候補手の評価値を算出
- 先年に囲碁AIにて簡易実験

  →単純な確率よりも性能が出る可能性がある結果

→より工夫して将棋AIに適応させるのが今回の目標

[WCSC34]

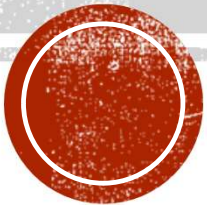
キレーティングについて

- キレーティングを利用した評価方法の詳細については
  うまくいけば「GI」か「GPW」にて発表予定。

- キレーティング自体の研究については「GI-51」において
  発表したため、その資料を次ページ以降に掲載。

[GI-51]

新しいレーティングシステム と その活用方法



2024/03/09

山田 元気

[GI-51]

レーティングとは

チェスにおいて、

アルパド・エムリック・イロ（Arpad Emrick Elo）により考案されたイロレーティング（Elo rating）が代表格。

対局結果から競技の強さを算出でき数学的に応用可能で、将棋や囲碁といったボードゲームやスポーツなどで幅広く応用可能。

[GI-51]

イロレーティングの課題 1

実力差のあるレートが大きく離れた対局において
高レート側がハイリスク/ローリターンなのに対し
低レート側はローリスク/ハイリターンとかなり不公平。

→ これの何がどのように問題なの！？

[GI-51]

イロレーティングの課題 2

たとえばある将棋サイト。

自分よりもレートが大きく高い相手との対局すると…

- 勝つと30点、31点ゲットもゲット
- 負けても2点、1点しか減らない

→ レート低い側はお得！！

[GI-51]

イロレーティングの課題 3

たとえばある将棋サイト。

自分よりもレートが大きく低い相手との対局すると…

- 勝つと2点、1点ゲットしかゲットできない
- 負けても31点、30点もゴリゴリ減る

→ レート高い側はつらい

[GI-51]

イロレーティングの課題 4

このレートの差による問題は対局内容にも影響が…

レート低い側は、破れかぶれの作戦や嵌め手作戦が有利。なぜなら実力差があっても対応を少し間違えるとつぶれてしまうから。

レート高い側は、無理攻めや嵌め手に対し、うっかりミスが出ないように慎重に指す将棋になりやすくつまらない。

→ 対局の内容までおかしくなる

[GI-51]

イロレーティングの課題５

その他の改善したい課題として、

- 多人数の勝敗ではイロレーティングは基本計算できない。
  （MM法など拡張はあるが複雑である）
- レーティングが地味（ドッカンバッカン差をつけたい）
- せっかくなので精度も改善したい。

[GI-51]

新レーティングについて

新レーティングの仕組みは単純。キレーティング！！

ゲームの参加者は、それぞれの持ち点の x%を場台に出す。

→場台に出た点数の合計を勝った人が総取り

（複数いた場合は分配してゲット）

→引き分けは無勝負で持ち点は元に戻る

（平均して分配のほうが適切か！？）

[GI-51]

キレーティング(QR)の計算例 1

下記ケースにて 0.02を係数として対局。

→プレイヤ Aは「QR1000」　　プレイヤ Bは「QR2000」

*プレイヤ Aを 勝利*とし、略式計算（勝ったら点数はもどってくるので）

[A の新レート] = [A の旧レート] + ([B のレート] * [係数])

[B の新レート] = [B の旧レート] - ([B のレート] * [係数])

→プレイヤ Aは「QR1040」　　プレイヤ Bは「QR1960」 となる。

[GI-51]

キレーティング(QR)の計算例 2

下記ケースにて 0.02を係数として対局。

→プレイヤ Aは「QR1000」　　プレイヤ Bは「QR2000」

*プレイヤ B を勝利*とし、略式計算（勝ったら点数はもどってくるので）

[A の新レート] = [A の旧レート] - ([A のレート] * [係数])

[B の新レート] = [B の旧レート] + ([A のレート] * [係数])

→プレイヤ Aは「QR980」　　プレイヤ Bは「QR2020」 となる。

[GI-51]

キレーティング(QR)は機能するの？

仮想の5人のプレイヤで簡易な確率ゲームを実施

1.ランダムで5プレイヤから1人を挑戦者として抽出

2.残りのプレイヤからランダムで対戦相手を選び対戦

3.各プレイヤ番号(1～5)に、乱数(0以上、1未満)を乗算し持ち点

4.両者の持ち点を比較し大きいプレイヤを勝ちとする

→これを4000回繰り返す

→浮動小数点を許容して計算

[GI-51]

キレーティング(QR)は機能するの？

仮想の5人のプレイヤで簡易な確率ゲームを実施

1.ランダムで5プレイヤから1人を挑戦者として抽出

2.残りのプレイヤからランダムで対戦相手を選び対戦

3.各プレイヤ番号(1～5)に、乱数(0以上、1未満)を乗算し持ち点

4.両者の持ち点を比較し大きいプレイヤを勝ちとする

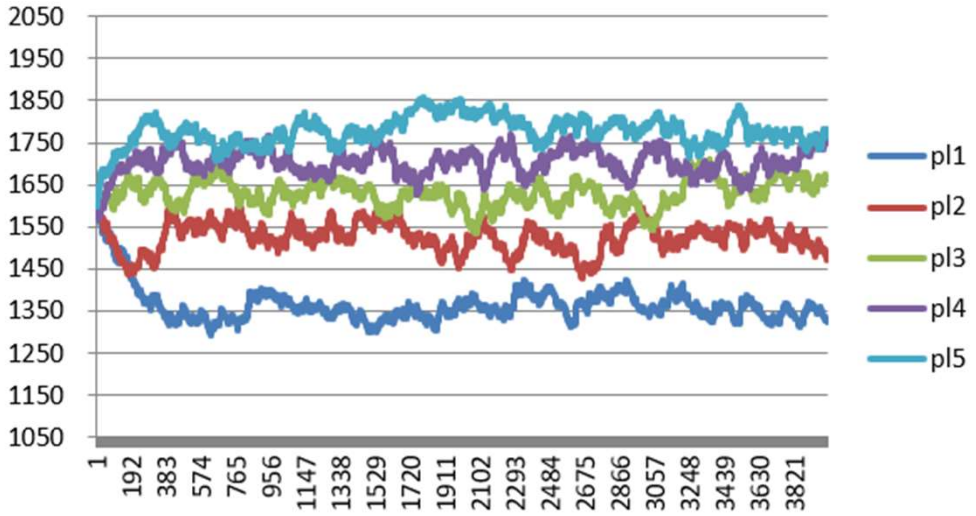
→レーティングの変動をグラフにして観察

→イロレーティングとキレーティングを比較

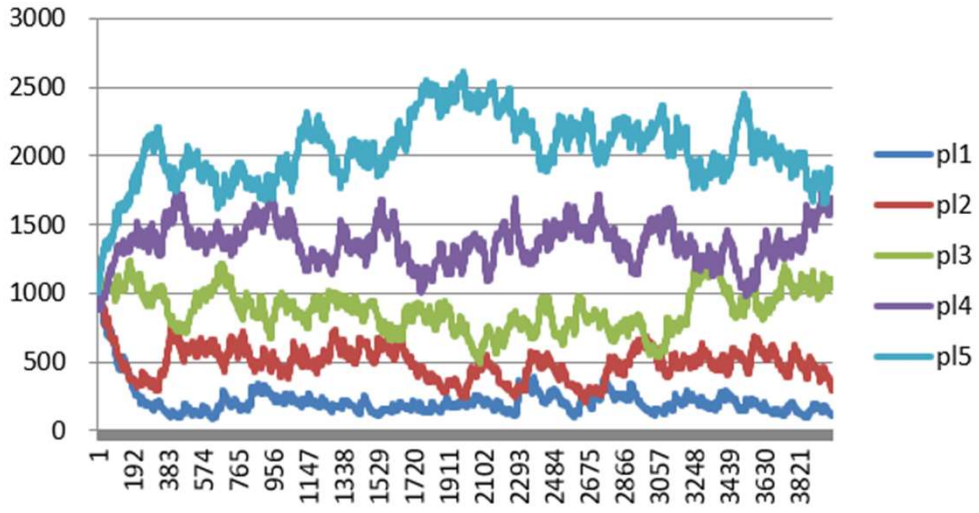
[GI-51]

各レーティングの比較 1

イロレーティング（係数16）



キレーティング（係数0.03）

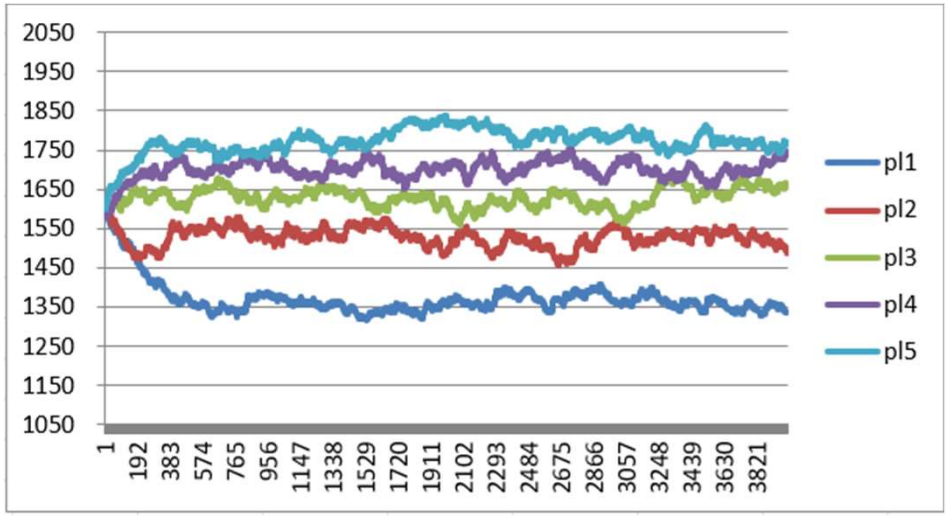


論文の図は、イロレーティング（係数16）は（係数10）、キレーティング（係数0.03）は（係数0.02）のものを誤って掲載

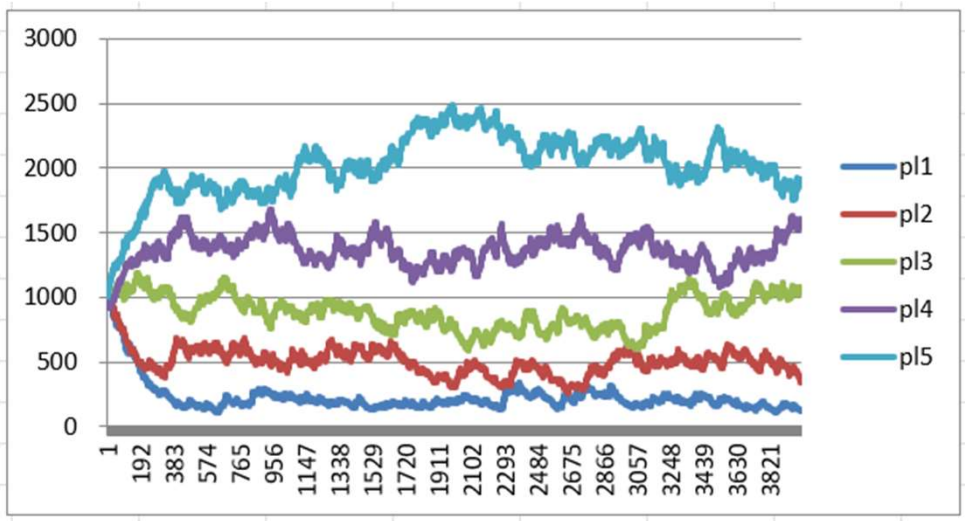
[GI-51]

各レーティングの比較２

イロレーティング（係数10）



キレーティング（係数0.02）

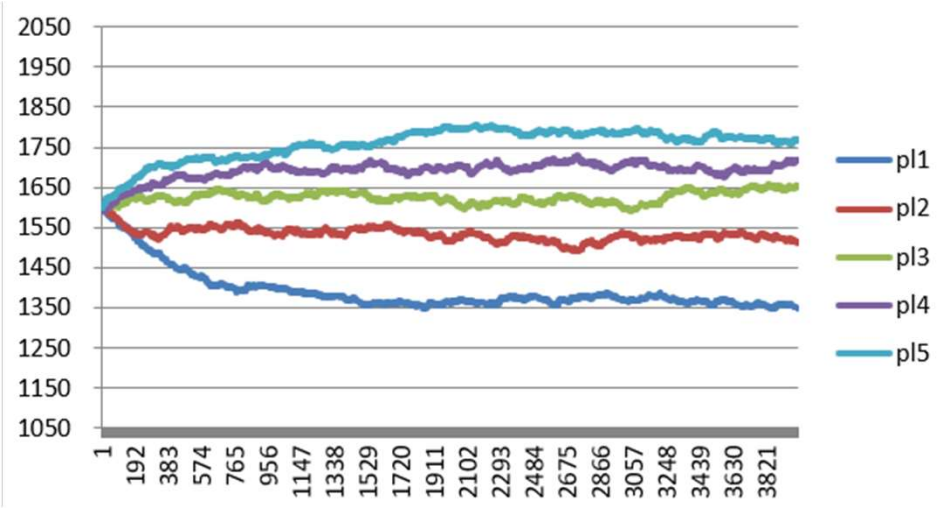


係数は大体同じくらいの意味になるのかなというところで、アドホックに設定

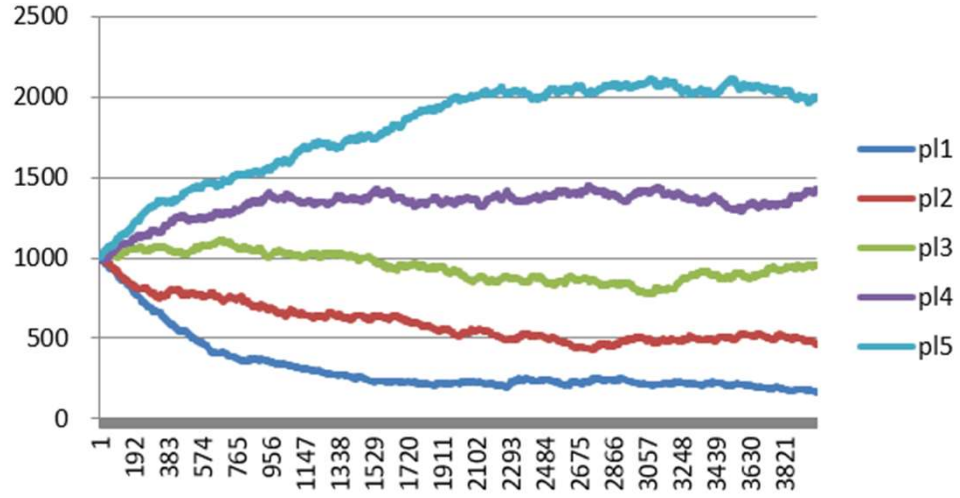
[GI-51]

各レーティングの比較３

イロレーティング（係数4）



キレーティング（係数0.005）

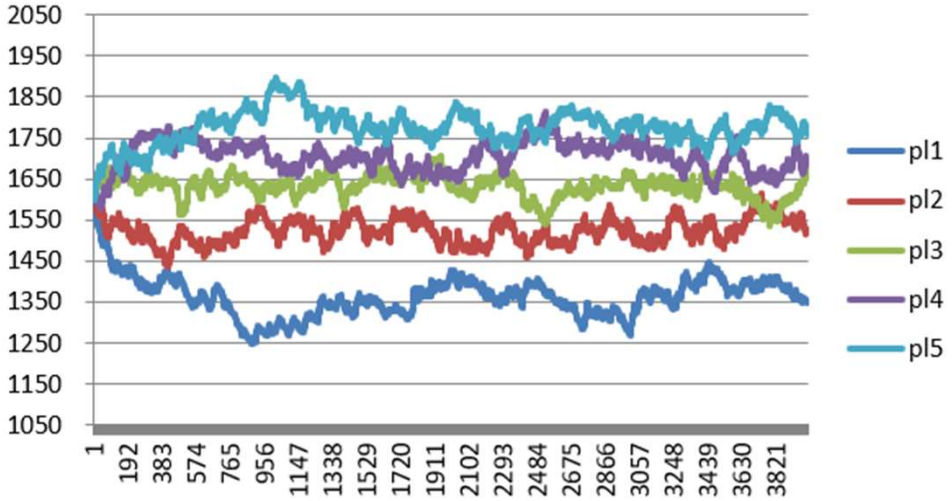


係数は大体同じくらいの意味になるのかなというところで、アドホックに設定

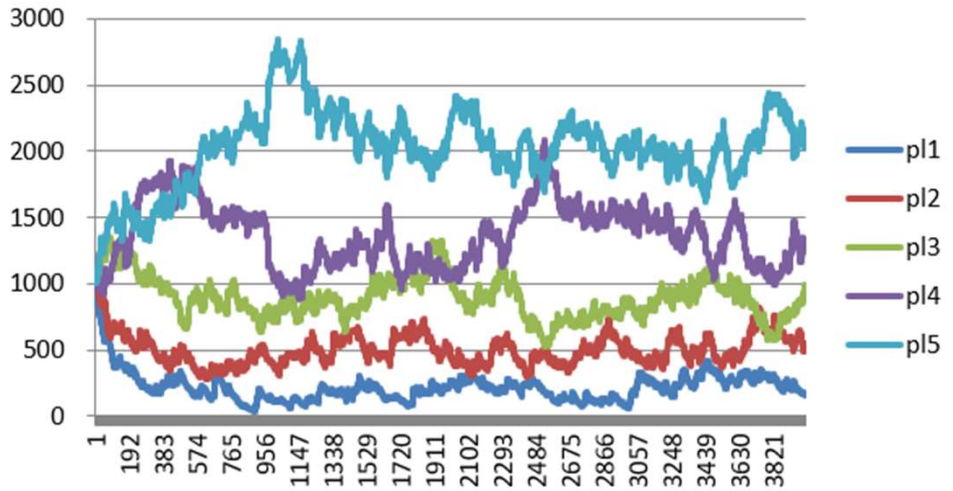
[GI-51]

各レーティングの比較４

イロレーティング（係数16）



キレーティング（係数0.03）

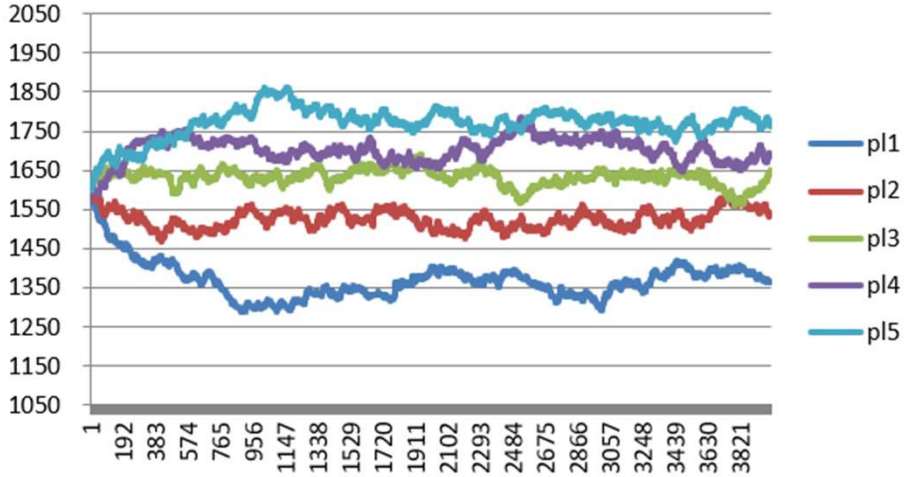


各レーティングの比較１，２，３とは別のデータセット

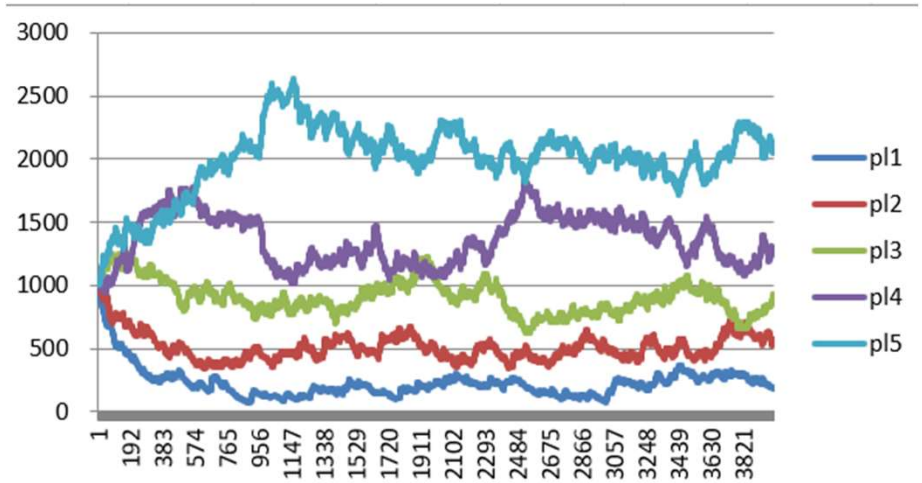
[GI-51]

各レーティングの比較５

イロレーティング（係数10）



キレーティング（係数0.02）

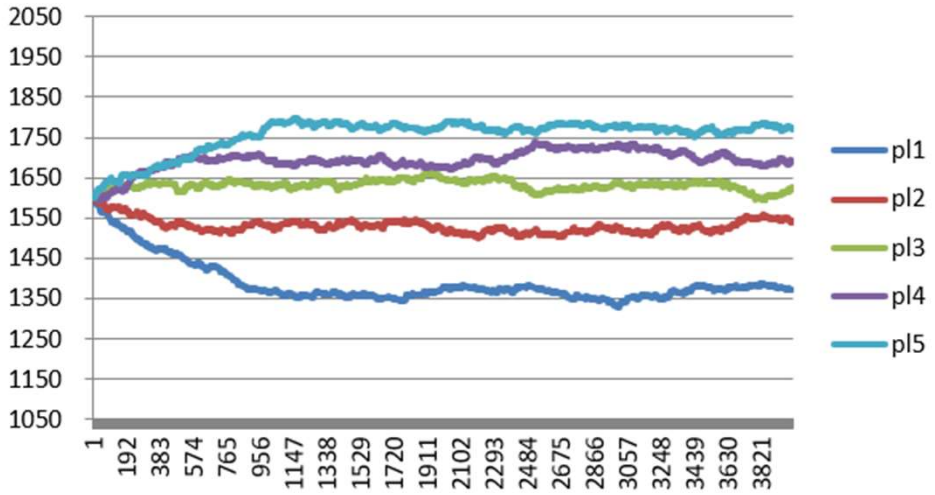


各レーティングの比較１，２，３とは別のデータセット

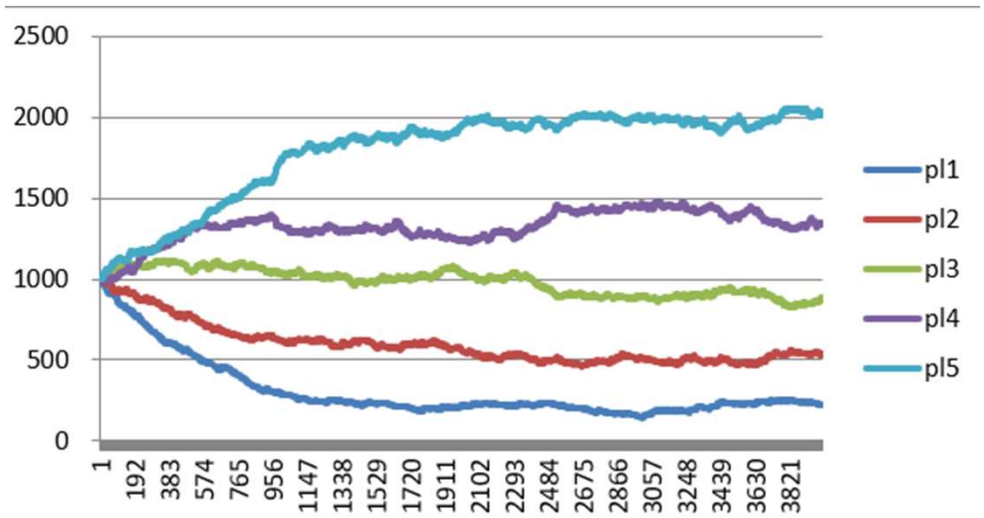
[GI-51]

各レーティングの比較６

イロレーティング（係数4）



キレーティング（係数0.005）



各レーティングの比較１，２，３とは別のデータセット

[GI-51]

各レーティングの考察と発展 1

→イロレーティングとキレーティングともに係数を小さくすることで収束は遅くなるものの精度を向上することを確認

→イロレーティングよりもキレーティングのほうが各プレイヤの分離が若干明確である一方で、収束は遅いように感じられた。（あくまで感覚によるものであり、係数にも依存するため錯覚かも）

→キレーティングの数字の違いがはっきり出るのは強さの指標が指数表現になっていないため。これはキレーティングのメリットでもありデメリットでもある。

[GI-51]

各レーティングの考察と発展2

→レーティング差が大きいときに、負けた時のリスクを両プレイヤともに一定の割合にすることにより同程度と感じられる設計に成功。この実際の人間の感じ方や、プレイスタイルにあたる影響程度は追加の研究が必要である。

→多対多の適用として簡易検証では、きのあ囲碁において12近傍のパターンをもとにした着手確立を利用するよりもレーティング評価でのほうが思考が強化され勝率が高い可能性が確認されたのでより研究したい。（本資料には未記載）

手抜きについて

手抜きチーム *

2024 年 3 月 31 日

手抜きは CSA プロトコルで対局を行うコンピュータ将棋プログラムです。開発者らが将棋プログラムの仕組みを理解するために作っています。
リポジトリ：https://github.com/hikaen2/tenuki-d

- NNUE
- α β 探索
- D 言語

使用ライブラリ

- 『どうたぬき』(tanuki- 第 1 回世界将棋 AI 電竜戦バージョン) の評価関数ファイル nn.bin[*1]

TODO

- 差分評価
- 千日手チェック
- 詰将棋
- ponder

おまけ：有用かもしれない過去のアピール文書

- 第 31 回世界コンピュータ将棋選手権, nn.bin の構造について（NNUE の解説文書）, https://www.apply.computer-shogi.org/wcsc31/appeal/tenuki/tenuki31.pdf
- 第 33 回世界コンピュータ将棋選手権, 私のための Minimax 入門（Minimax の解説文書）, https://www.apply.computer-shogi.org/wcsc33/appeal/Tenuki/tenuki33.pdf

* 鈴木太朗 (https://mastodon.ttg6.net/@hikaen2), 玉川直樹 (https://twitter.com/Neakih_kick)

[*1] https://github.com/nodchip/tanuki-/releases/tag/tanuki-denryu1

「技巧」アピール文書

2024 年 3 月 31 日

出村　洋介

1. Gumbel AlphaZero を用いた効率的な強化学習

AlphaZero [1]の学習を効率化した Gumbel AlphaZero [2]を用いて，既存の棋譜を使わずに強化学習でゼロから学習を行なっています.

オリジナルの AlphaZero では難しかったような軽量な実験条件（1 手 16〜64 シミュレーション程度）でも Gumbel AphaZero では比較的安定した学習ができるため，学習に要する計算資源が小さく済むようになり，各種実験を行いやすくなりました.

選手権用には，さらに学習時間を増やしたバージョンで参加予定です.

2. 強化学習時の初期局面を多様化

元々の AlphaZero では強化学習時の初期局面は 1 種類（平手初期局面）のみであり，手元の実験では自己対局で似たような展開になりやすい傾向が見られました.

そこで，今年の技巧では，自己対局時にさまざまな局面を積極的に経験させるため，学習時の自己対局で用いる初期局面の多様化を行なっています. 具体的には，2 手ランダムに指した局面（下図（b））や，駒の配置を一定の制約のもと並べ替えた初期局面（下図(c)）を初期局面として強化学習を行なっています（下図(c)の並べ替えでは駒の配置が左右対称を除き約 81 万通りあるため，「チェス 960」[3]にならって「将棋 81 万」と呼んでいます[4]).

このように強化学習時の初期局面を多様化することにより，実験では平手初期局面のみでの強化学習と比較して同一対局数で一定の勝率向上が確認できています. 学習順序としては，学習初期には「将棋 81 万」（下図(c)）で幅広い形を学習させてから，2 手ランダムの初期局面（下図(b)）で学習を行う方法が調べた中では効果的でした.

(a) 平手初期局面　(b) 2手ランダムの初期局面(例)　(c) 将棋81万の初期局面(例)

3. Rust と Python による独自実装

主要な開発言語には Rust を利用しており，既存の将棋ライブラリを使用しない独自実装

となっています．実装上は以下のような工夫がされています．

・Rust 組込みの 128 bit 整数型を用いた Bitboard

・利き数を保持するデータ構造

・利き情報を活用した高速な 1 手詰関数［5］

・ニューラルネットワークへの入力に将棋独自の特徴量を追加

また，学習部の開発は主に Python を用いて行っており，高速化や安定性向上のために次のような実装上の工夫がされています．

・PyTorch Lightning の混合精度学習による学習の高速化［6］

・Torch-TensorRT を用いた推論の高速化［7］

・Python 側と Rust 側のデータの受け渡し時に安全な NumPy 形式で転送［8］

参考文献

[1] D. Silver, et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science, 362(6419): 1140–1144, 2018.

[2] I. Danihelka, et al. Policy improvement by planning with Gumbel. In International Conference on Learning Representations, 2022.

[3] Wikipedia, https://ja.wikipedia.org/wiki/チェス 960.

[4] 出村，金子．将棋 81 万: 強化学習のための多様性を持った将棋初期局面集．第 28 回ゲームプログラミングワークショップ, pp. 111–118, 2023.

[5] 金子，田中，山口，川合．新規節点で固定深さの探索を行う df-pn の拡張．情報処理学会論文誌, 51(11): 2040–2047, 2010.

[6] PyTorch Lightning, https://lightning.ai/docs/pytorch/stable/common/precision_intermediate.html.

[7] Torch-TensorRT, https://github.com/pytorch/TensorRT.

[8] Rust-numpy, https://github.com/PyO3/rust-numpy.

CGP アピール文章

2024/3/31
大熊　三晴

主な特徴

・無駄に一から作成
・非ビットボード型
・無駄に高 NPS を目指してるけど最近この部分はさぼり気味
・局面構造体に各マスへの利きの状態を保持
・局面構造体に評価関数の演算途中結果のうち変化の頻度が少ないものを中心に保持
・評価関数も自力で学習
・AVX-512 命令をはじめとした拡張命令を活用
・現在のところ去年からの変更点は反復深化における探索深さの決定手法と
探索パラメータの微修正です。
・大会までにはアピール文書を書き直すくらいに開発が進んでて欲しいです。
・一般に流布している定跡データや、一般に流布している「局面と評価値のセット」、
読み筋等は使用しておりません。無駄なこだわりだとは思いますが。

・１から作成

強さをあまり考えずに高 NPS を目指して自作したプログラムをベースとして
おります。並列化手法は現在は LazySMP を微修正したものです。

・非ビットボード型，利き等を保持

非ビットボードだとビットボードに比べ遅くなる処理もありますが、複雑な
情報を持てることにより速く処理できる可能性もあります。ビットボードに比べ
遅い処理をうまく避けるために利きを保持したり、局面構造体の配置をビット
位置を含めて工夫しております。

また利きの保持以外にも、演算途中のデータを保持することによりメモリ
アクセス待ち時に演算を回す事により高速化を狙っております。

・評価関数

現在は手番付き KPP です。ただし持ち駒周りの評価を一般的な 3 駒関係より拡張
しております。

・SIMD等の活用

高速化のためSIMDを活用しております。SIMDは現在評価値の算出、指し手の生成、オーダリング、構造体のコピーが主な使用箇所です。ただしAVX-512F、CDまでの対応でAVX-512VBMIなどの対応はしておりません。

動いていることが奇跡で、特段アピールすることなんてないのですが、
アピール箇所がないとアピール文書リジェクトとなることを（確か）2015年に実証しておりまして、
運営の方に迷惑をかけるので、とりあえず前回のアピール文書から技術的なところをコピペしておきます。

・探索はαβ中心で、LMRとかFutilityとかの至って普通の技術を使ってます
→なんか評価関数をいじったらLMR効かなくなったのでどうするか（20180331）
→バグをとったら効いたので入ってます(2018大会時点)
→なんか評価関数をいじったら明確に弱くなったどうしよう(20190301)
→なんか静止探索をいじったらまれに動かなくなったどうしよう(20190301)
・いわゆるボナンザメソッドを使ってます、もうちょっと改造できないかな
・オンライン学習を勉強してみたかったので、平均化パーセプトロンを試しています
→今更ながらこれ本当に平均化パーセプトロンなのか・・・？って気もしてきた（20180331）
→怪しかったので試すのをやめました(20190301)
・なんと打ち歩詰めのバグが未だにあったので修正しました(20200330)
・なんと王手されている時に受ける手に未だにバグがあったので修正しました(20200330)
→めったに起きないので強さにあまり影響がない、というかむしろ探索が遅くなって弱くなりました。。。なぜ。。。けどさすがに外せない
・探索がもうちょっといい感じになる予定です(20200330)
・今年は正直何も手を入れられていないです・・・健康はいいぞ(20210331)
・2020年バージョンから探索を丸っと作り直し、評価関数も学習しなおしたら、ちょっとだけ強くなりました。
もっと強くなると思ったのになぜ同じくらいに収束するんだ・・・(20220329)
・今回は誤差レベルでしか強くなってない・・・正直まったく強くなってないと思う・・・(20230331)
・今回はバタバタしていて、何もできてないですすみません・・・(20240401)
以上です。

アピールについては以上なので、思い出話と参考URLを貼っておきますね。

といいつつ、過去のアピール文書をコピーしたところも多いですが。。。

毎年なんだかんだ起きるので、参考URLは増える一方です。

■始めての参加

始めての参加は第17回の時で、選手としてではなく、アルバイトとして小谷研から徴集されました。

場所はかずさアークでして、近くのホテルに泊まるとバイト代から足が出るという訳の分からない状態だったのですが、

世界で初めて？パズルで博士を取ることになる先輩にそそのかされ（なぜか今同僚）、

君津のネカフェに3泊4日し、毎朝となりのマックでご飯を食べてました。

初日は全く寝ることができず、すごくつらい思いをしたことを覚えています。

もう二度とネカフェで3泊4日はしたくありません。

第17回大会

<http://www2.computer-shogi.org/wcsc17/>

かずさアーク

<http://www.kap.co.jp/>

自遊空間君津店

<https://jiqoo.jp/shop/9931865/>

マクドナルド君津店

<https://map.mcdonalds.co.jp/map/12031>

■始めての出場と、（恐らく大会史上初の）プログラム名リジェクト

実は当初プログラム名は、「(´・ω・`)」にしていたのですが、

運営の方から「読めません」と言われリジェクトされ、今のとても強そうな名前になりました。

最近の事例を見る限り、読み方をつけておけばリジェクトにはならなかったぽいので、ちょっと後悔をしています。

デビュー戦はなかなかに熱かったです、白砂将棋さんと対局させていただきました。

あの負け方（王手千日手負け）は二度と忘れることは無いでしょう、悔しかったw
実は二次予選に行って20位でしたすげえ。

第21回大会
<http://www2.computer-shogi.org/wcsc21/>

■始めての賞罰

・独創賞受賞（解説記事の生成）
・（恐らく大会史上初の）アピール文書リジェクト

2012年に独創賞いただきました、ﾔｯﾀｰ
2017年にどう考えてもポナがDLぶん回していて独創賞取れる状況にも関わらず、
「優勝しなかったから独創賞対象なし」という恐ろしい裁定が下されていたので、早いうちに取っておいて本当に良かったと思いました。

第22回大会
<http://www2.computer-shogi.org/wcsc22/>

■会場に行かずに近くで遊ぼう
有名なマクドナルド定跡(関東人の意地としてマクド定跡とはよばない)について、毎年試している割には全然勝ち星が増えないので、もしかすると効果がないのではないかと最近思い始めました。
まだ試行回数が足りないと思うので、今年も行きたいと思います。
また、再びかずさアークでの開催となった時期から、なぜかいつも二日目が暇になっていたので、
将棋を見るのではなくて、どうせだからどこかに遊びに行くとかそんなことやってました。
行った場所は下のほうにまとめておきました。
2016年には罰ゲームでうまるちゃんやりました。

その際は、（確か菅井先生だったかな）プロ棋士の方が「なんか変な人いるんですけど大丈夫なんですか？」と運営の方に相談されていたそうです、すみません、ぼくは大丈夫な人です。
また、千田先生からは「ちょっと身長が高すぎるかもしれませんね」とご指導いただきました、ありがとうございました。
この話を最近（2018年夏ころ？）小谷研の後輩さんにお話ししたところ、「マジっすか、ご褒美じゃないですか！？」と言っていただいたので、今後ぼくの人生におけるご褒美イベントの一つとして大切にすることにします。
ところでコスプレは罰ゲームだと思っていたのに、たぬきさんとか自発的にコスプレされているので、コスプレは罰ゲームじゃなかったんだなあと思いました。
もうやりたくありませんが、とりあえず一式は取ってありますので、うまるちゃんになりたい人がいればお貸しすることは可能です。
2018年は永瀬先生のお父様のお店である川崎家に行き、ネギチャーシューラーメンを食べました、辛みがアクセントになって非常においしかったです。
家系ラーメンは大好きですし、きっと二日目は暇になるので（え、今年もぜひ行きたいと思います、もしよかったら皆さん行きましょう！
(20200330追記)
去年はラーメン食べている間に対局が始まってしまい会場入りが間に合わなかったのですが、きちんと対局がスタートしていました。
自動化されているって素晴らしいなって思いました、もう会場いかなくてずっと川崎家でラーメン食べてもいいですね。
あと今年は二次予選に進出しないと二日目に川崎家いけないので、是が非でも初日に川崎家に行っておきたいと思います。
(20220329追記)
地元のおいしいレストラン（とある弁護士事務所のそばだったりする）でランチを食べることを楽しみに対局してました。
だけど5/3は月曜日で確か空いてなかったのよな・・・。5/5に食べに行った記憶がある。
今年は川崎家だ・・・行きたい・・・
(20240401追記)
川崎家美味しかったですね・・・

無限に川崎家食べたい・・・・

喜楽飯店(担々麺)

<https://tabelog.com/chiba/A1206/A120603/12012396/>

東京ドイツ村

<http://t-doitsumura.co.jp/>

マザー牧場

<http://www.motherfarm.co.jp/>

東京湾観音

<http://www.t-kannon.jp>

食事処やまよ

<http://www.yamayo7240.com/>

うまるが家でかぶってるアレ [干物妹！うまるちゃん]

<http://cospa.co.jp/detail/id/00000065274>

マクドナルド 川崎ソリッドスクエア店

<https://map.mcdonalds.co.jp/map/14707>

川崎家 榎町店

<https://tabelog.com/kanagawa/A1405/A140502/14013754/>

停車場

<https://tabelog.com/chiba/A1202/A120202/12000477/>

■まさか自宅から大会に参加することになるとは・・・

2020年はコロナの影響もあり、まさかのリモートでの大会となりました。

割と暇な時間が長かったので、大会中にずっとゲームをやっていたのは内緒です。

下記のゲーム、渋滞を起こして街が死ぬってパターンが多い気がする。

シティーズスカイライン
<https://www.spike-chunsoft.co.jp/cities_skylines/>

■送りバントはいいぞ
自分のプログラム名について一応拾っておきますか。
2019年ですが下記の試合を観に行ってました。送りバントからのサヨナラ。送りバントはいいぞ。
ただ今年(2021年)、日テレ系プロ野球中継で、「イニング得点確率」を予測する人工知能があるのですが、バントしたら確率下がってたんですよね・・・送りバントはいいぞ。

【ハイライト】7/3 劇的なサヨナラ勝ちの巨人が今シーズン3度目の4連勝！【巨人対中日】
<https://www.youtube.com/watch?v=eb9etHJK-2o>

野球のイニング得点確率や作戦成功確率を予測するAIを開発日本テレビ系プロ野球中継で、「AIキャッチャー」に次ぐ進化系AI施策としてサービス提供が決定
<https://www.datastadium.co.jp/news/6655>

アピール文書を書きながらジャイアンツの開幕戦（2023年）を観ているのですが、村神様が初打席でホームラン打ったそうです。
WBCでは味方になってくれましたが、敵になった村神様は恐怖でしかないですね、早くFAでジャイアンツに来ないかな。

■久しぶりの賞罰(20240401)
2023年の大会で、久しぶりに表彰していただきました！
卒論で上手くいかなくてゆうちゃん倒したかっただけなのに、なんか10回も出場してましたか・・・
ということは10回もGWを室内で過ごしてるんだひええ・・・
あと、フロムスクラッチで5位なのが意外すぎる・・・、もっと順位下だと思ってた・・・
フロムスクラッチで作るのもかなり面白いし、プログラムがかわいく思えるのでとってもおすすめで

す！

・フロムスクラッチ表彰 5位

・10回出場 34勝35敗

■将棋が強くなりたい(20240401)

そもそも自分が将棋できないので、初段とやらを取れたらカッコいいなと思って勉強しているのですが、まったく強くなる気配がありません。

いい方法があれば教えてほしいです。。。

■目標

まったりゆうちゃんを倒して師匠超えしたいです。

最近は直対すると勝てるケースが多いのですが、順位で勝てないケースがたまにあるので、ぜひ勝ちたいですね。

あとできれば勝ち越ししてみたいですし、久しぶりに2次予選にも進出してみたいです・・・！

去年(2021年)は二日目暇になることを見越して、あらかじめ桃鉄やる約束を取ってしまってました。

その時他のメンバーにボコられて悔しかったので感想戦をやったのですが、「1年目の5月のムーブが悪い」と言われました、なんて厳しいゲームだ。

桃太郎電鉄 ～昭和 平成 令和も定番！

<https://www.konami.com/games/momotetsu/teiban/>

ー昨年は惜しかったんですよね・・・王手ラッシュして逃げられるっていう負け方しました。。。あの対局勝ってれば上に上がれたらしい・・・。

同じ方に去年当たったのですが、レベルめっちゃ上がってて、手も足も出ませんでした・・・。

今年も去年同様に、CSA運営の皆さんも頭を抱えられたことと思います。

まだ予断を許さない状況でもありますが、川崎での開催に向けて進めてくださっている運営の皆様には心より御礼申し上げます。

それでは今年も、参加者の皆さん、CSA運営の皆さん、よろしくお願いします。
がんばるぞー。

まったりゆうちゃんのアピール文書 2024

今回は、去年とまったく同じシステムである。改良することができなかった。可能な限り出場続けたいとと思っている。以下は前回と同様の説明である。

複数のシステムを動作させて、その間でデータをやり取りして並列動作をする方法を検討している。以下は前回とほぼ同様の説明である。

1990 年過ぎから開発を始めた。4 半世紀にわたって開発しているシステムである。当初のコードもたくさん含んでいる。完全独自開発であり、他のシステムを参考にしていない(考え方は参考にしている)。アイデア的にも独自工夫をしている。

今日、AI というとディープラーニングをはじめとしてパラメータ学習に基づくものが多い。コンピュータ将棋での駒価値学習もそうである。しかしそうでない進化論的計算などの方式を試そうとしている。またディープラーニングと多量パラメータ学習の中間的なメカニズムを考えたいと思っている。

実現できるかどうかわからないが、全面的に書き換えることを考えている。今日からみれば、適切でないコーディングもある。それを直すことで、新しい並列方式が実現できると考えていて、少しとりかかっている。

開発者の年齢がもっとも高いのではないか。可能な限りやめないで続けたいと思っている。コーディング能力がいつまで続くか試したい。

去年と全く同じ。https://www.apply.computer-shogi.org/wcsc33/appeal/Katsudon_Shogi/wcsc33appeal.txt

【第34回世界コンピュータ将棋選手権】

コンピュータ将棋ソフト『元気もりもりニンニクパワー』アピール文

Vtuber 都賀町えいだ

【第34回世界コンピュータ将棋選手権】
『元気もりもりニンニクパワー』アピール文(1)

・Vtuber 都賀町えいだが作成したソフト

・名前は公募＋アンケートによって命名
　名付け親は牛車さん(@gyuusha3)

【第34回世界コンピュータ将棋選手権】
『元気もりもりニンニクパワー』アピール文(2)

- 「強い将棋ソフトの創りかた」
  著：山岡忠夫、加納邦彦
  これをもとに言語(C#)で作成し
  python版を超える棋力になること

基本方針はWCSC33と変わりなし

【第34回世界コンピュータ将棋選手権】
『元気もりもりニンニクパワー』アピール文(3)

●WCSC33、第4回電竜戦からの改善点

主に詰み部分関連の追加

学習する棋譜を厳選する

→ 第4回電竜戦では2008年以降のfloodgate棋譜で学習したため、微妙な棋譜もあった

→ 今回はfloodgate以外の棋譜も使用予定

主にたややんさん公開のもの

【第34回世界コンピュータ将棋選手権】
『元気もりもりニンニクパワー』アピール文(4)

・定跡は今回もなしの予定
　トップレベルの評価関数でないので
　単純な棋力向上を優先

3/31時点では第4回電竜戦バージョンより強くなっていないので、下手したら電竜戦バージョンかも

山田将棋について（2024 年）

全体像

クラシカルな構造・アルゴリズムとなっています。

データ構造

配列による盤駒表現、駒背番号制、利き数、
飛び利き方向ビットの OR 値、
利いている駒背番号ビットの OR 値、
囲いへ誘導するための落とし穴表、
玉位置からの距離に応じた評価値を納めた表、
pinned と cover の概念、置換表、
8 近傍利き位置を納めた表、8 近傍合法移動先を納めた表、
など

アルゴリズム

αβ探索、反復深化、局面が静かでない場合の探索延長、
手調整した仮評価による手のオーダリングと前向き枝狩り、
null move pruning、late move reduction、
killer move heuristic、pass move、
YSS 式指し手の反復生成、Crafty 方式の並列探索、
反復深化による詰め将棋ルーチン
root ノードでの簡易必死検出、
leaf ノード付近での簡易一手詰み検出、
予測読み、フィッシャークロック対応、

など

評価関数

駒割、玉の安全度、囲い、駒への当たり、大駒の働き、
盤上の利き（SIMD を用いた合算）、
そっぽ、金駒へのひもなどの、手調整した評価値の線形和

未採用

多重反復深化、影の利き、SEE、持ち駒の優劣表現、
bitboard、実現確率探索、評価関数評価値の学習、など

こまあそび
アピール文書

2024/03/31

探索部

・基本アルゴリズムは$\alpha\beta$法。
・王手、王手回避手、駒をとる手などを延長している。
・延長深さの制限を先手番、後手番別々に持っている。
　たとえば先手番Max4手、後手番Max4手の場合、
　先手4手延長＋後手4手延長=計8手延長はOKだが、
　先手5手延長＋後手3手延長=計8手延長はNGなど。
・手を読む広さは探索深さによって変えている。

評価部

・評価関数は学習は使わず手でチューニングしている。
・駒組みは落とし穴方式で行っている。
・銀桂は敵陣に近くの手の点数を高くしている。
・金は上部に出る点を低くしている。
・金は角と筋違いの位置の点数を高くしている。
・中盤、終盤は角と金の価値がほぼ同じにしている。
・竜王、飛車の価値を高めに設定している。

爆裂駒拾太郎
アピール文書

作成日：2024年3月31日

最近の将棋AIは
入玉が苦手らしいですね？

つまり爆裂駒拾太郎が
最強ってコト！？

目次

1. 爆裂駒拾太郎について
2. WCSC33からの変更差分まとめ
3. 主な使用言語・ライブラリ
4. 定跡部
5. 探索部
6. 評価関数部

爆裂駒拾太郎について

- 開発方針
  - 詰将棋が意味ないことを証明する
  - 詰みではなく基本的に入玉宣言による勝ちを目指す
  - 思考に大きな影響を与える他者の創作物は使用しないフロムスクラッチ将棋ソフトとして開発する
    ※指し手生成はライブラリ使用

- GitHub
  - URL:https://github.com/burokoron/Yolts

目次

1. 爆裂駒拾太郎について
2. WCSC33からの変更差分まとめ
3. 主な使用言語・ライブラリ
4. 定跡部
5. 探索部
6. 評価関数部

WCSC33からの変更差分まとめ

- 探索部
  - Null Move Pruningを実装しました
  - Late Move Reductionsを実装しました
  - 置換表を連想配列から動的配列に変更し、エンジン起動直後にメモリ確保するように変更しました
  - 指し手並び替え用配列を動的配列から静的配列に変更しました
  - 置換表を世代管理し、前回の探索結果を活用するように変更しました
- 評価関数部
  - KKP型評価関数(第7世代)からPP型評価関数(第2世代)に変更しました
  - 局面の評価を差分計算するように変更しました
- 評価関数学習部
  - 学習に使用するライブラリをTensorFlow 2.5.0からPyTorch 2.1.1に変更しました
  - 学習データ生成部をJITコンパイルするように変更しました
  - スパーステンソルを用いて学習するように変更しました
- 学習棋譜生成部
  - 千日手手順は選択しにくくなるように変更しました
- その他
  - 評価関数ファイルが読み込まれていない場合のみ、isreadyコマンドで読み込むように変更しました

目次

1. 爆裂駒拾太郎について
2. WCSC33からの変更差分まとめ
3. 主な使用言語・ライブラリ
4. 定跡部
5. 探索部
6. 評価関数部

主な使用ライブラリ(対局時に使用)

- プログラミング言語
  - Rust 1.73.0
    - 対局エンジンの作成に使用

- ライブラリ
  - yasai 0.5.0 [1] ベース
    - Rustで利用できる将棋ライブラリ
    - 指し手生成、局面管理に使用
    - 使いやすいように一部改変

[1] yasaiのURL : https://github.com/sugyan/yasai

主な使用ライブラリ(開発時に追加使用)

- プログラミング言語
  - Python 3.11.5
    - 学習データおよび評価関数の作成に使用
- ライブラリ
  - cshogi 0.7.5 [2]
    - Pythonで使用できる将棋ライブラリ
    - 学習データおよび評価関数の作成における局面管理に使用

[2] cshogiのURL : https://github.com/TadaoYamaoka/cshogi

目次

1. 爆裂駒拾太郎について
2. WCSC33からの変更差分まとめ
3. 主な使用言語・ライブラリ
4. 定跡部
5. 探索部
6. 評価関数部

定跡部

- 学習データの生成時の副産物として作成
- 学習データの棋譜から各局面の勝ち数、負け数、引き分け数を算出し、Thompson Samplingアルゴリズムに基づいて指し手を選択する
- ただし、訪問数が10未満の局面となるような指し手は選択されない

R	時間	深さ	ノード数	評価値	読み筋
11				48	▲5八金(69)
					6i7h move_rate=48.83% black_wins=250 white_wins=278 draw=22
10				48	▲7八金(69)
					5i6h move_rate=48.92% black_wins=583 white_wins=564 draw=44
9				48	▲6八玉(59)
					7i7h move_rate=48.98% black_wins=191 white_wins=215 draw=17
8				48	▲7八銀(79)
					1i1h move_rate=49.26% black_wins=323 white_wins=335 draw=29
7				49	▲1八香(19)
					9g9f move_rate=50.02% black_wins=325 white_wins=342 draw=32
6				50	▲9六歩(97)
					2h4h move_rate=50.25% black_wins=792 white_wins=747 draw=72
5				50	▲4八飛(28)
					5i5h move_rate=50.86% black_wins=816 white_wins=782 draw=60
4				50	▲5八玉(59)
					4i3h move_rate=51.43% black_wins=242 white_wins=271 draw=24
3				51	▲3八金(49)
					4g4f move_rate=51.45% black_wins=517 white_wins=508 draw=38
2				51	▲4六歩(47)
					2g2f move_rate=55.06% black_wins=189809 white_wins=146995 draw=46527
1				55	▲2六歩(27)

目次

1. 爆裂駒拾太郎について
2. WCSC33からの変更差分まとめ
3. 主な使用言語・ライブラリ
4. 定跡部
5. 探索部
6. 評価関数部

探索部

- 探索アルゴリズム
  - αβ探索
  - 反復深化探索
  - 静止探索(駒取り、王手回避のみ最大3手延長)

- ムーブオーダリング
  - Hash Move (前回の探索結果も活用)
  - Killer Heuristic
  - MVV-LVA
  - History Heuristic (piece-to)

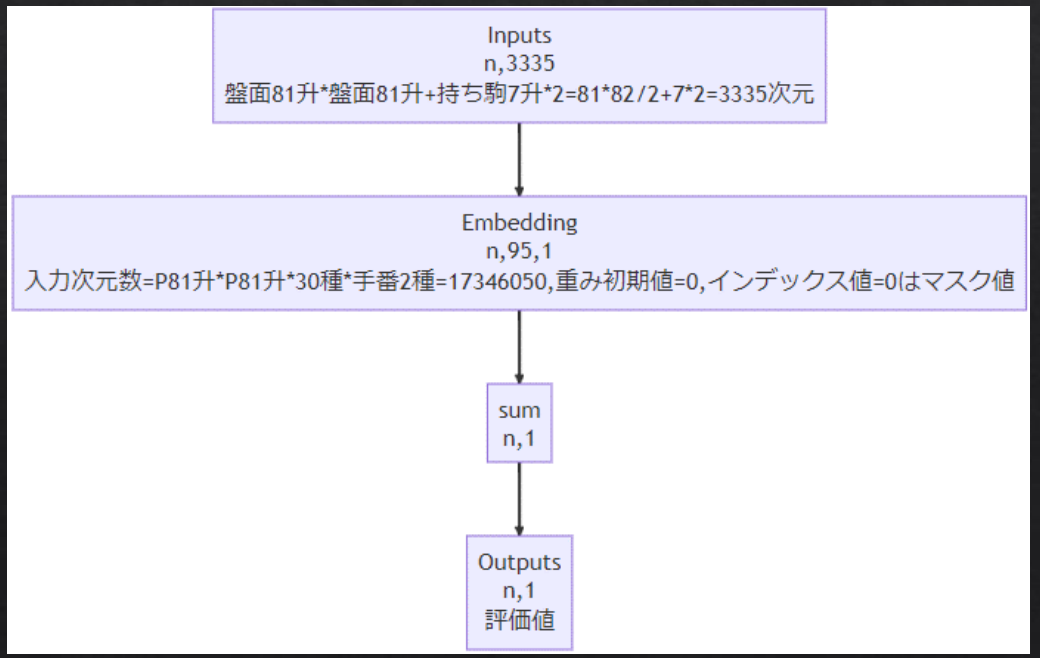
- 前向き枝刈り
  - Mate Distance Pruning
  - Null Move Pruning
  - Late Move Reductions

目次

1. 爆裂駒拾太郎について
2. WCSC33からの変更差分まとめ
3. 主な使用言語・ライブラリ
4. 定跡部
5. 探索部
6. 評価関数部

評価関数部(1/2)

- 方針
  - 詰みではなく入玉を目指すような評価関数を作成する
- 学習時と推論時の違い
  - 推論時は1手進んでも局面があまり変化しないため、変化した特徴量のみ差分計算する
- アーキテクチャ
  1. 入力層(2駒関係特徴量)
     - 3335次元
  2. 埋め込み層(2駒関係特徴量)
     - 17,346,050次元⇒1次元
  3. 出力層
     - 合計



評価関数部(2/2)

- パラメータの調整
  1. 探索部により強制的に相手を詰ますと負けになるようにし、
     入玉でなければ勝てないようにした爆裂駒拾太郎を作成する
  2. 通常の爆裂駒拾太郎と対局を行い棋譜を作成する。
     棋譜をばらけさせるために1度以上出現した局面については、
     勝ち数、負け数、引き分け数を算出しておき、
     Thompson Samplingアルゴリズムに基づいて指し手を選択する
     ※ただし、引き分け数が出現回数の半分以上を占めている場合は選択率を下げる
  3. 棋譜のうち1で作成した爆裂駒拾太郎の手番の局面と評価値のみで
     評価関数の学習を行う
  4. 3で作成した評価関数を新たな爆裂駒拾太郎とし、1へ戻る。
     ただし、2では過去の評価関数との対局も行う。
     このときの対局割合は勝率の高い評価関数を小さく、
     勝率の低い評価関数を大きくする
- 初期値は小駒100点、大駒500点とし、敵陣に近いほど1点加点(最大9点加点)
  する評価関数とする

・alpha-beta法で7-9手先を読んでいます。

・末端では、1-2手読むのに相当する静的評価関数を使っています。

・強さには直接関係がないのですが、GUIに多少力を入れており、

盤面を3Dで斜め上から見た絵にしています。

リアルタイムで動かせるのですが、そこに一所懸命になって、

相手の手を受信し損ねるのがこわいので、当日は動かさない予定です。

JHBR将棋のアピール文章

2024年

JHBR将棋

アピール文章

JHBR将棋のアピール文章

JHBR将棋は、cshogiライブラリを使用して、ミニマックス法による探索を行います。ノードの評価は駒得で行います。ミニマックスによる探索は、量子コンピュータのライブラリ(Qiskit)を使って行なっています。ちゃんと動けば１試合くらい実機を使いたいと思っています。
よろしくお願いします。

第３４回世界コンピュータ将棋選手権

きふわらべ using cshogi
アピール文書

２０２４年０３月３０日　高橋智史

MURI
無理

NANNIMO
何にも　やってない

開発者　高橋　智史

「 今年も　何にも手を付けてないな……
そうだ、去年に　電竜戦のワン・ファイル・マッチの練習会に出した
きふわらべ　があったな、アレで出るか」

（←左の画像は著作権上、問題ありません）

コンピュータ将棋エンジン　きふわらべ

「 cshogi のランダム・ムーブだろ」

（←左の画像は著作権上、問題ありません）

ひよ子

「 どうせ　ライブラリを使うのなら　やねうら王を　使えばよくない？」

（←左の画像は著作権上、問題ありません）

「 ディープ・ラーニングも　評価関数も　探索も
わたしには合わないしな。
指し手生成部を　cshogi　にやってもらおうというわけだぜ」

ライブラリ使用宣言

思考部に大きな影響を与える、他者の作成したプログラム・データ等を利用します。

OTON

お父んは　何をやるんだぜ？

SASHITE　SEISEI

指し手　生成　部しかやったことないのに……

「ディープ・ラーニングと　評価関数と　探索が合わなくて、
指し手生成部を　cshogi　がやってくれるんだったら、
お父んは　何をやるんだぜ？」

「デターミニスティックな　ルール・ベース　の思考部の作成だろ」

「その詳細をこのＰＲ文書に書きなさいよ」

「目の前の局面で、次の１手は　０～５９３個　しか無いのだから、
およそ　６００本のくじがある　くじ引きで
常に１番良いくじを引けるような　運の良さ　を上げたら
いいんじゃないか？」

「目の前を見ろだぜ」

「でも　あんたのお父んは　将棋を見る目が無いのだから、
盤を見て　考えても意味ないのよ。
くじ運　を上げるしかないのよ」

「ランダム・ムーブより　くじ運がいいことを　示せだぜ」

MINAI

盤を見ない～

MITEKURERU

盤は cshogi が 見てくれる のだから

「 多分、 cshogi が合法手の一覧を返してくれるだろ」

「 `board.legal_moves` 配列に入ってんじゃないかな」

「 じゃあ もう 将棋盤を見るとか、 相手が何を考えてるか察するとか、
そういうこと 全部止めようぜ？

`board.legal_moves` 配列が 全てで、
その中から １本のくじ を引くことに集中しようぜ？」

「 貧すれば鈍ずるの わらう」

「 利きとか 駒割りとかも 見ないのかだぜ？」

「 将棋を指さない。 くじを引く」

「 １１５手ぐらいで 対局が終了すると仮定したら、
くじを ５７～５８本 引くだけでいいのだから、
１局面の平均合法手数を ８０手 と仮定すれば、
８０の５８乗は １０の１１０乗よ」

「いや、考えなければならないのは
合法手の　セットの数だぜ。

自分のくじ引きと　相手のくじ引きは　深く考えるのを放棄して　独立。
マルコフ過程だぜ」

「サブセットを考え出したら　べき集合　にならないかだぜ？」

「平手初期局面の合法手は３０手と言われているけど、
駒落ちの初期局面は　例えば左香落ちなら２９手ですしね」

「あっ、お父ん　待てだぜ。
コンピューター将棋の指し手は　移動元、移動先、成り、打　だぜ。
自分がどの駒を動かしているかという情報は持っていないぜ？」

「むしろ　計算しやすくなったぜ。
将棋盤は８１マス、持ち駒は７種類、成りは難しいが
どこでも成れると仮定して
（８１＋７）×８１×２　＝　１４２５６

もしかすると　指し手の符号は　１４２５６　個も無いんじゃないか？」

「そのべき集合の要素数は　２の１４２５６乗　ぐらいあるんじゃないの？」

「１０の４２９２乗ぐらいか。
あれっ？　１０の１１０乗より多いぜ？」

「その、べき集合　とか言うの　止めようぜ？
深く考えるの、禁止！」

「じゃあ　１４２５６個の要素を　全順序に並べなさいよ」

「将棋の指し手が　全順序に並んでいるとは　思えない……」

「全順序だが、その全順序を決めるシード値が
セットの組み合わせによって決まればいいんじゃないか？」

「それが　べき集合　になると言ってる。
全数調査は無理だろ。標本抽出にしろだぜ」

「標本かあ。いいなあ」

「全射であり、単射ではないわねえ」

「おみくじって　そんなもんだろ。
引きにくる客の数より　おみくじの番号の数の方が少ない」

「じゃあ　お父んが　やるのは　シャッフルだけだぜ。
くじを　かき混ぜて　一番手前の　くじ　を引くだけ……」

IRANAI

盤の表示は要らない～

KUJI

くじ を引くだけなのだから

「 あれっ？　じゃあ　選手権に出る前に　いつも作ってる
将棋盤の表示のようなものも　要らないのかだぜ？」

「 おみくじに　盤なんて　無いわよ」

「 よーし、わたしが　やるべきことが　はっきりしてきたな。
シャッフル・アルゴリズム　を作ることだぜ」

「 全自動麻雀卓みたいだな」

>>> Dad, stir up
the lottery well

2024年アピール文書

プログラムの基本は初回参加時から使いまわしていて

一般的な手法であるαβ法、ハッシュテーブル、null move等を使っています。

３駒間評価値を去年作成した学習データを元に強化学習をさせています。

申し込み者：北川博隆（HN:雨宮一也）
グループ名：重力団
プログラム名：重力場計算法

2024 年　第 34 回大会提出アピール文

重力場計算法を引き続き開発中です。
目に見える成果が出るのは次回以降となりそうです。

****

2023 年　第 33 回大会提出アピール文

重力場計算法を引き続き開発中です。
目に見える成果が出るのは次回以降となりそうです。

****
2022 年　第 32 回大会提出アピール文

盤面評価を重力場計算法を用いて計算します。

重力場計算法 PV
将棋の数学的解明に必要となる哲学と理論物理学、そして開発者に課せられる心構えについて謳った動画。

ニコニコ動画
https://www.nicovideo.jp/watch/sm40237301

YouTube
https://youtu.be/m3nkswy9X3g

****

2021 年　第 31 回大会提出アピール文

■「参加プログラムは、主要な開発者が思考部に技術的に何らかの明示的な工夫を施したプログラムであること。」
　を満たすことをアピールしていただくための文書

このプログラムの思考部は「盤面評価を最高精度で算出する事で最善の一手が特定できる」との趣旨で構成されています。
その為以下の様な特徴があります。
・通常の「深い読み」を行わない。
・「定跡・棋譜」を一切利用しない。解析データも利用しない。
・将棋のルールと評価設定のみを入力する。
・盤面判断は重力場理論（簡易版）を使用して計算を行う。

将棋はお互いに一手ずつ指すことで競技が進行しますが、手番、非手番に与えられる条件は都度同じです。
従って将棋の本質とは与えられた盤面に対する最小手数範囲を計算できれば良いのであり、それ以上の処理は重複作業になるため不要と言えます。
結果、根拠のない「深い読み」を意識的に行わないプログラムとなります。

又、将棋のルール内に過去の「棋譜・定跡」の使用を義務付ける項目はないし、それらによって競技ルールに何かしらの影響が発生する事もありません。
従って「棋譜・定跡」をプログラム内に入力することは無駄であると言えます。

盤面判断を行う上で最も困難とされていたのは「個々の設定の解析」と「異なる単位を正確に比較する手法」です。
その為今までの盤面評価プログラムは精度が悪く、進歩が止まりました。
しかし、この二つの課題は重力場理論によって克服することが出来ます。

私たちの制作したこのプログラムは異なる単位の設定であっても、重力場理論（簡易版）によって計測処理を行い、上昇値の合計を比較する事で「最高精度の盤面評価」が理論上達成されます。

重力場理論に基づく究極のアルゴリズムは「神の一手」を特定できるのです。
つまり「将棋の数学的解明」はこの理論によって完成されるでしょう。

■「このプログラムの思考における工夫や独自性について」

従来のプログラム思考技術や人間同士の対局で培われた知識や成果を基にした「工夫」は行われていません。
なぜならこのプログラムはそれらを全否定する所から構成が行われているからです。

真逆の方向性を目指して作られている上に、将棋のルールと評価設定をプログラムに入力しているだけの代物であり、面白味は何もありません。
只、淡々と盤面向上を追求する一手を計算しています。
それこそが究極の将棋なので、人為的努力の「工夫」は不要です。

しいて言えば、「人間の知恵と技術と歴史と情熱を皆無にする事」が既存のプログラムにない「独自性」なのかもしれません。

対象物を”鏡に映しただけ”の作業は工夫もなければ独自性も発生しません。
しかし、世界で初めてその行為を行った場合は「イノベーション」として評価されます。
そして現実社会に多大な影響と貢献をもたらせば「究極 AI 産業革命」として歴史に記録されます。
”将棋を鏡に映しただけ”なのに……ね。

第34回世界コンピュータ将棋選手権「水匠」アピール文書

令和６年３月３１日

たややん

第１　定跡作成プログラム

１　ＷＣＳＣ３３においては、以下の定跡作成手法を採用していました。

⑴　指定局面から連続対局させ、指し手を全て定跡として登録する。

⑵　任意の局面が定跡に登録されていた場合、ミニマックス法で定跡データ内を探索し、勝ちの枝があるか、負けの枝しかないか判別する。

⑶　勝ちの枝があれば、その手を指し、負けの枝しかない、又は定跡に登録されていない局面であれば、探索エンジンで思考させる。

２　これに対し、ＷＣＳＣ３４における定跡作成手法は、以下の手法により、ミニマックス法で探索する必要なく、前項⑴～⑶と同様の定跡が作成できる仕組みを採用しました。

⑴　対局が終わった際、勝った側の指し手は全て定跡データベースに登録する。

⑵　負けた側の指し手は、末端の局面から、定跡データベースに今回選択された指し手と別の手が登録されている局面が現れるまで、定跡データベースから削除する。

⑶　定跡データベースに指し手が登録されていればその手を指し、登録されていなければ、探索エンジンで思考させる。

３　コードは公開されています。

https://github.com/tayayan/HiraganaSuisho/blob/main/makebook2.py

第２　評価関数の学習

第３２回世界コンピュータ将棋選手権に出場されたマメットブンプクの評価関数からファインチューニングをした評価関数を使います。

学習手法での工夫は、以前とほぼ同様（学習させる局面の評価値に閾値を設ける、学習時FV_SCALEの調整、勝敗項の教師信号の低減等）です。

教師データとして、日本AMD株式会社様よりお借りした、AMD EPYC™ 9654プロセッサーやAMD Ryzen™ Threadripper™ PRO 7995WXプロセッサーを活用し、水匠の1手1000万ノードで対局して作成したデータを使用しました。

当該教師データも一部（約１億局面）公開しています。

https://drive.google.com/file/d/1VyP4MX_AuQhvy8sesymgPVf9sUnQoGPl/view?usp=drive_link

第３　評価関数リレーの採用

上記学習方法を採用することによって、入玉宣言が苦手な評価関数となっているため、一定の評価値を超えたら水匠5にスイッチする仕組みを採用します。

第４　探索部の改善

やねうら王を使用し、Stockfishに実装された探索部の変更部分の採用及びパラメータ調整をしています。

以上

「習甦」

【探索】

1スレッドはdf-pnにより詰みを確認，他のスレッドはStockfishから取捨選択したシンプルなα-β探索

【評価関数】

玉の位置に対する各駒の位置と全升目の利き数（先後別3まで）を特徴量とするニューラルネットワーク

【機械学習】

自己対戦において初手〜4手目までMultiPV24〜16に飛車を振る手を入れることによりオールラウンダーをめざす

ゼロベースから評価関数を効率良く学習するため，初期段階では入玉も考慮した駒割り関数の値を報酬に加算

Argo（アルゴ）WCSC34 アピール文 2024.3.31

ソフト名/soft-name：Argo（アルゴ）

開発者/developer：市村豊（いちむらゆたか/Yutaka Ichimura）

X（旧Twitter）:@argonworks

Blog：http://blog.livedoor.jp/argon1/

2024.1.31時点

WCSC32のときの評価関数をそのまま使って参加させて頂こうかと思っています。

WCSC32の作文に記載した内容を以下に転載します。

＊＊＊＊＊＊＊＊

「水匠 3 改」を元にして以前やねさんが配布していた 150GB の教師データを 3 つ使って学習させたものが、水匠 5 に対して 50 回くらい対局させて勝率が 40%くらい（20 勝 30 敗くらい）になりました。現状はそれでやろうかと思っています。

使用ライブラリとそれを使用した理由

やねうら王：使いやすくて改造をしやすいから。

水匠 3 改：色々と評価関数を作っていてこれを元にしたのが現状で一番良さそうだったからです。

＊＊＊＊＊＊＊＊

**********以上です。以下はおまけです。

この文章を読んでいただきましてありがとうございます。あなた様の大変貴重な時間を使ってこの文章を読んでいただけること、感謝しております。

以下の文章は無理に読む必要はありません。個人的にはコンピュータ将棋選手権のアピール文を読む人にとって興味深く読めそうな作文を以下に書いたつもりでおります

2024.1.31時点 気が向いたら2024.3.31までにもうちょっと加筆修正するかもしれないです。
以下は、アピール文のおまけの作文の草稿です。現状の分を載せておきます。
読みたくなければ無理をして読まなくて良いです。

この作文を読んでいるということはあなたは頭が良い人だということです。
コンピュータ将棋の大会のアピール文を読んでいるという、そんなことをしている時点であなたは頭が良い人だということです。
あなたが、自分が頭が良いということを自覚しているのならばそれで良いのですが、自覚していないのならば自分は割りと頭が良い人間であるということを自覚してもらって良いです。

> 人生は他人との競争ではない話。
囲碁の名人と将棋の名人は争っていない。それと同じように私やあなたは他人とは争っていない。人生は他人との競争ではない。将棋の名人が囲碁の名人よりも劣っているということがないように、あなたが他人よりも劣っているということはない。

仮に競争であるとしたら、一番になるよりも最下位になる方が良いのではないかと言う風に思う。一番になるということは「動物園のサルの飼育員」とか「幼稚園の先生」みたいなもので、周りの人の面倒を見ないといけなくて非常に面倒でしかない。

そもそも大体の人は一番ではない。
「一番以外は一番ではなく、大体の人は一番ではない。なので、一番ではないことは標準的なことであり一番の人が例外的」ということ。
あなたが今どういう状況かは知りませんが、「自分は今現在すごく良い状態で勝ち組です」というのでしたらそれは良いことだと思いますが、そもそも世の中の大体の人は多分勝ち組ではありません。だから私やあなたが勝ち組ではないとしてもそれは全くもって標準的な状態であるということです。
私がこれまで根拠なく誤解していたことですが、大体の人は負け組であって、勝ち組のほうが例外的な少数派ということです。だから私やあなたが負け組の状態にあるとしてもそれは単に人間として標準的

な状態にあるというただそれだけの話しです。人間的に立派であるとか、能力が高いとか、価値があるとか言うことは大体の人はない。だから劣等感を感じて苦しむ必要はまったくない。

もちろん可能であるならば勝ち組になる方が良いとは思います。立派であるとか能力が高いとか価値があるとかいう人間になるために努力する方が良いとは思います。最終的に死ぬまでずっと負け組で終わるとしても、それでも努力するのは意味があることだと思う。というか、負けても負けても闘い続けるというその事それ自体が十分に立派なことだと思います。結果的に勝つかどうかというのは副次的なことだと思います。

具体例として明治維新のときの新選組の土方歳三という人はかなり人気がある人だと思います。ご存知のように明治維新において新選組は最初から最後までずっと負け続けます。土方歳三は敗走を重ねながら最後までずっと戦い続けた人です。それだから人気なのです（「ゴールデンカムイ」という人気のマンガがありますが、読んでもらえばわかりますが土方歳三が影の主人公の話です）。

日本のアニメやマンガのストーリーは「序盤から中盤にかけて負けているが最後に勝利する」という内容のものが多い印象を受けますが、実際のところは、「大体の人は、人生は最初から最後までずっと負け続ける」ものなのではないかと私は思っています（それなので、私やあなたが人生において最初から最後までずっと負け続けるとしてもそれは人間として標準的であるというだけであって、標準よりも劣っているということはない）。それでも結果的に勝つかどうかはともかくとして、勝ちを目指して挑戦を続けることそれ自体に意味があることだと思います。やったことがある人はわかるでしょうが、そもそも勝つことよりも負けても負けても闘い続けることのほうが難しいことです。勝っているから続けるというのは技術的には難しくとも精神的にはそこまで難しくはない。負け続けているのに闘い続けることのほうが技術的には難しくなくとも精神的には難しいことです。それなので、負け続けているのに闘い続けている人は勝っている人と同じくらいに立派なのです。

他人の不幸を願う理由がない

「他人の幸福を自分の幸福のように喜び、他人の悲しみを自分のことのように悲しむこと」それが人間にとって最も重要なことである、というのが「ドラえもん」の結婚前夜のエピソードに描かれている藤子・F・不二雄のメッセージです。そして、それができない状況のときというのは、大体は自分の側に問題があることが多い。そのことが自覚できているのならばまだ大丈夫だと個人的には思っています。

他人に対する評価というのは自分に対する評価の反映だと思います。やたら他人に文句を言っている人

というのは、本人が問題を抱えていることが多いです。そもそも悪口というのは自分が言われたら嫌なことを言うわけです（それなので悪口というのはそれを言われている人ではなくて、それを言っている人の欠点を表していることが多い）。本人が自分に満足していたら他人に文句を言わないと思います。我ながら、そもそも他人の批評という生産性があまりないことにリソースを投じている時点で負けている感じがします。興味関心は自分の未来に向けたほうが良いと思います。過去と他人にこだわるべきではない。変えることができるのは未来の自分だけなので、自分の未来のことに意識を集中させたほうが良いと思います。

自分が得をすることをする。

現在の日本において経済学と言ったら大体は新古典派経済学を指すと思うのですが、その基本的な考え方は「人間は自分が得をすることをする（人間は自分が損をすることはしない）」「人間は常に合理的に振る舞う」ということを仮定して社会現象を分析するものだと個人的には思っています（社会全体を大きな将棋盤だとみなして、人間一人一人が駒であると考えて盤上の動きを予想すると考えると、経済学というのは将棋の研究と本質的には同じことのように感じます）。この「人間は常に自分が得をすることを目的に合理的に振る舞う」という仮定は、当然ながら実際とは多少ズレています（実際の人間は道徳心や利他心を多少は持っているし、いつも合理的に振る舞うわけではない。それなので、人間はいつもいつも合理的に振る舞うわけではない、として分析する行動経済学の考え方があってこれはこれですごく面白いです。行動経済学は、経済学に心理学を組み合わせて社会現象を分析する感じ）。

その「人間は自分が得をすることをする」という考え方ならば、「他人が不幸になることで自分が得をするならばそれをする」。しかし、それは例外的なことだと思う。多くの場合は他人が不幸になっても自分は得をしない。それなので、他人の不幸を願う理由はない。

そして、他人が幸福になると自分も幸福になるのであれば、他人の幸福を願う理由になる。

社会科学的に考えて、他人がパフォーマンスを上げると自分も得をするのではないか？ ということを私はなんとなく思っている。

他人が全部やってくれるならば自分は何もしないで毎日ゴロゴロと寝て暮らすことができる。それなので、他人がパフォーマンスを上げることは歓迎するべきことだと思う。

他人がパフォーマンスを上げることを邪魔をする理由はない。

「将棋」というゲームは「二人零和有限確定完全情報ゲーム」、ゼロサムゲームです。それなので、相手が得をするとその分だけ自分が損をするというゲームになっていますが、それは将棋というゲームが特殊なのであって、私たちが今いるこの社会というのは「他人が得をするとその分だけ自分が損をするというゲーム」ではないと思います（このことを学術的に論証してみたらどうなるのかというのは気になります。社会科学なのか、ゲーム情報理論なのかどのジャンルになるのかもよくわかりませんが）。

私は生まれてから現在まで一貫して勤労意欲がない人間です。
日本国憲法第二十七条一項で規定されている「勤労の義務」を果たす意志が皆無という、存在そのものが憲法違反な私です（ついでに言えば日本国憲法第三十条で規定されている「納税の義務」も可能ならば果たしたくないとも思っています。義務というものが嫌いな割に権利意識はやたら高いという、我ながら私は模範的な日本国民なのです）。
一体何をどうすれば「勤労意欲」などという気の利いたものが湧いて出てくるのかというのが不思議でしょうがないと私は心から思います（最も勤労意欲がある人からすれば「何でそんなに働くのが嫌なんですか？」と心から思っていることでしょうが）。
しかしながら、若さ故の過ちから7年9ヶ月の間、私は勤労と納税をしてしまいました。実際に勤労をしてみて、自分に勤労が向いていないことを再認識することができました。
そんなわけで、いい感じに無職になった私は、少し調査をしてみて市議会議員は基本的に一ヶ月に一回駅前のゴミ拾いをすることさえしていれば大体の日はのんびり過ごせそうだと思うようになりました。
そんなわけで、市議会議員だったら私のような勤労意欲がない人間でもできるだろうと思って、市議会議員になろうと2023年4月の統一地方選のときに行われた広島県呉市の市議会議員選挙に立候補してみたのですが、30万円の供託金を没収される結果になったのは我ながら遺憾に思います。
そんなわけで、市村さんは会社を解雇されていい感じの無職になって市議会議員選挙に立候補したら落選したどころか30万円の供託金を没収されたりしているけど今のところ人生を前向きに生きているので、あなたが今どういう状況かは知りませんが少々のことで人生を後ろ向きになる必要は全くないのです。

もし「私が市村さんの分まで二人分の勤労と納税をするので、市村さんは勤労と納税をしなくても良い

のですよ」と、言ってくださる方がいるのでしたら、私はその方に大いに感謝すると同時にその方を応援したいと心から思います。

他人が勤労をして納税することを妨げるべきではない。

余談ですが、市議選に立候補してみて、「選挙公報」という新聞みたいなのが発行されるのですが、それの広告代だと考えると30万円は十分にペイするなと思いました（供託金を没収された私が言うのもなんですが、市議選の場合は供託金没収ラインを超えるのは一般的にはそこまで難しくはないと思います）。自分が選挙に立候補するまでは東京都知事選挙に立候補されている独立系候補の方々について、「この人たちは一体何なんだ？」と思っていたのですが、市議選に立候補してみると、都知事選に立候補するのは300万円であれだけ広告が打てると考えるとむしろ「安い」と思うようになりました。都民向けの商品を持っていて都民向けに広告を打つことが有効な人は都知事選に立候補するのは全然ありだと思います。

> 囲碁・将棋のような競技の有益性。
「人生は他人との競争ではない」と私は思っています。
「『人生は他人との競争ではない』ということを前提にするから、将棋のような競技を安心して行うことができる」というのが私の認識です。個人的には別になんの役に立たなくたって良いじゃないか、とは思っていますが、幸か不幸か将棋のような競技はおそらく相当になにかの役に立っているのだろうという感じがします。
オリンピックとか、サッカーのワールドカップとか、あれは「代理戦争」なのでしょう。人が死なない戦争をしている。それをすると人が死ぬ戦争が発生する頻度を減らすことができるのだろうと私はなんとなく思っています。
個人的に私はサッカーが好きという以上にJリーグが好きなのですが、それは「地域間の代理戦争」をしているところが見ていて面白いというふうに思っています。バスケットボールのBリーグとかプロ野球とかもそういう「地域間の代理戦争」という側面があるような感じがします。私が好きなモータースポーツについて言えば、自動車のレースというのは自動車メーカー同士の代理戦争なんですね。

ゲームセンターを運営されていた方がTwitterで「ゲームセンターは年齢や性別や社会的地位に関係なく対等な立場で交流することができる稀有な場所」という投稿をしていて、私は「なるほどー」とすごく納得しました（現在はクレーンゲームが多いかもしれませんが、かつての格闘ゲームの筐体がおいてあった頃のゲームセンター）。囲碁とか将棋もそういう「年齢や性別や社会的地位に関係なく対等な立場で交流することができる稀有な場所」なのだろうなと。そういうのは非常に重要だし、非常に有益だと思います。

なんというか、戦争をしているときの「中立地帯」な感じがします。言い方を変えると、サードプレイスですね。個人的にそういう場所は非常に重要だと私は思っている人なのでそれは大事にしたいと思います。

学校とか会社とかでしんどくても、将棋の大会で上位に入賞したというようなことがあれば、それを支えにして人は生きていけると思うんですよ。そういうことを思えば、おそらくは囲碁や将棋のような競技はマイナスよりはプラスのほうが大きいのだろうと思う。

> ゲームの有益性について。

ゲーム雑誌「ファミ通」連載のデジタル技術のコラム、西川善司「ゲームのムズカシイ話」において「日本においてはゲームは子供が遊ぶもの、というふうな認識のされ方をすることが多いが、ゲームを『その時点におけるコンピュータ・サイエンス技術の集大成』と捉えている文化圏は意外と多い。特に最近はゲーム技術を産業応用することが行われるようになってきていて、学術界からも注目されている」というふうなことが書いてあった気がします（注）。たしか2023年の5月頃の号だった気がする。

今日の日付の新聞に書いてあるようなことを含めて「歴史」というのは「社会実験の記録」だと私は思っています。

そして「SFは未来についての思考実験」という言われ方をする事があるのですが、それを言えばSFに限らずアニメ・漫画・小説・ドラマ・映画、のようなフィクションの物語というのは、「思考実験の論文」だと私は捉えています（注）。

ゲームは現実世界のシミュレーターという風に私は捉えています。わかりやすい例で言えば、将棋と本質的には同じゲームである「大戦略」とか「ファミコンウォーズ」のようなゲームは戦争のシミュレー

ションとして行われていた兵棋演習を娯楽にしたもののようです（注）。2023年に公開された映画「グランツーリスモ」は、レースゲームのチャンピオンから本物の自動車のレーサーになったヤン・マーデンボロー選手の実話の物語です。

日本はもうダメなのではないか、という話をたまに聞くことがありますが、「日本がこれからの30年間でほぼ間違いなく世界ナンバーワンを取れる分野が一つだけあって、それが「マンガ・アニメ・ゲーム」のポップカルチャーの分野だ」という話を聞いたことがあります（どこで聞いたかは覚えていないのですが）。
個人的には、そう簡単にトヨタ・ホンダ・日産・スズキ・マツダ・SUBARUが自動車産業において敗北するだろうかと思いますが、仮に自動車産業でトヨタ・ホンダ・日産・スズキ・マツダ・SUBARUとかの日本メーカーが負けるにしても「マンガ・アニメ・ゲーム」がある限りこれからの30年間は日本は大丈夫だという風に私も思っています。

注：ゲーム技術の産業応用
https://xtech.nikkei.com/atcl/nxt/column/18/00001/07867/
> 物流や工場にデジタルツイン、AmazonやBMWが活用
> 現実空間の物体や環境を仮想空間に再現した「デジタルツイン」を利用して、最新の大規模施設の設計や構築、検証を行う動きが活発になってきた。先行するのは海外勢だ。米Amazon.com（アマゾン・ドット・コム）は物流施設、ドイツBMWは電気自動車（EV）工場を仮想空間で構築し、生産性向上やコスト削減を図る。

https://blogs.nvidia.co.jp/2021/05/10/nvidia-bmw-factory-future/
> NVIDIA と BMW、現実世界と仮想世界が融合された未来の工場を実演

注：「SFは未来についての思考実験」
松本健太郎「スマホアプリはなぜ無料？ 10代からのマーケティング入門」河出書房新社 P161-162
＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊

特にインターネットが浸透した今の社会では、デジタルでしか実現できない体験を作ることが、大きなお金を生み出すことに直結します。ですから大人たちは、デジタルでイノベーションを生み出すのに必死です。

ただ、先ほどお話ししたように、イノベーションが起こったあとの未来を、イノベーションが起こる前に具体的に想像するのは、なかなか難しいわけです。

こうした状況で、大人たちは一体どんなことをしていると思いますか？

その答えは、「SF思考」です。これまでは物語の一ジャンルであったSFのような考え方を元にして、新しい体験を作ろうとしています。

（中略）

といっても、素人がSFの未来の世界を描くのは簡単ではありません。そこで大人たちは、すでに世の中に出ているSF小説やSF映画、SFマンガも参考にしながら、頭を柔らかくして、まだ世の中にない新しい体験を生み出そうとしているのです。

こうした背景もあって、14歳の皆さんが、今からたくさんSFものを見たり読んだりしておくのは、SF思考を身につけるという意味でも、将来的に役立つんじゃないかと思います。

そう言われても「SFってなんだか難しそう」と敬遠してしまう人もいるかも知れませんね。一見とっつきにくそうなジャンルですが、先に上げた「ドラえもん」もSFです。アニメなら「機動戦士ガンダム」シリーズや「サマーウォーズ」など、親しみやすい作品もたくさんあります。

＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊

注：兵棋演習

> チェスター・ニミッツは第二次大戦時の対日戦は海軍大学校で学んだ兵棋演習の再演であり、予期できなかったのは神風攻撃のみだったと語っている。

https://ja.wikipedia.org/wiki/%E5%85%B5%E6%A3%8B%E6%BC%94%E7%BF%92

アピール文 240331追記

> 囲碁・将棋ソフトの競技会は実質的にLLM（大規模言語モデル）の性能を競っている。
大雑把に書きます。

「囲碁ソフトや将棋ソフトは、割と容易にLLM（大規模言語モデル）に転用が可能ではないか？」と現在の私は思っています（「容易」かどうかは主観の問題ですが）。棋譜の代わりにウィキペディアのテキストのような文章を大量に学習させたら、それはLLMになるのではないか。

　それなので、囲碁・将棋ソフトの競技会は実質的にLLMの性能を競っているのに近い状態だと思う。そしてそれをやっていたら最終的にはAGI（汎用人工知能）に到達するのではないか？　と私は思っているわけです。
　オープンソースプロジェクトとしてLLMやAGIの開発をしているのが囲碁・将棋ソフトの競技会、という風な捉え方をしてみたらどうかということです。
　OSについて、Windowsに対してのリナックスOSがオープンソースプロジェクトとして開発されたように、LLMやAGIにおけるオープンソースプロジェクトなのではないかという風に思う。
　それなので、これを続けていたら、OSにおけるLinuxのようなポジションに、LLMやAGIにおいて囲碁・将棋ソフトの競技会がなるのではないかと思っているわけです。

> 個人的に、知っておくと何かの役に立つかもしれないと思っていることを書いておきます。

　スマートフォンで、音声入力をすると、テキストを簡単に入力できます。

　LLM（大規模言語モデル）で、検索というか調べ物をすると良いです。
　マイクロソフトのCopilotとグーグルのGeminiを（無料で使えるために）私は使っているのですが、分からないこととか困ったなと思ったときにはCopilotとGeminiに自然言語で質問文を入力すると、割と有

益な答えが返ってくるのですごく良いと思います。

　例として「会社を解雇されて途方に暮れているけど、これからどうしたら良いですか？」「市議会議員選挙に立候補して当選するためには何をするとよいですか？」というふうな文章を入力すると割と有益な答えが返って来ます。LLMは種類によって返答の内容が違うので可能ならば複数の種類のLLMに同じ質問をして比較してみると良いです。

　個人的にはマイクロソフトのCopilotとbingのアプリをスマートフォンに入れておいて、いつでも使えるようにしておいたほうが良いと思っています。

・マイクロソフトのbing。ここからCopilotが使えます（ここからだとGPT-3.5じゃないかと思いますが、スマートフォンのアプリからだとGPT-4が使える）
https://www.bing.com/
・GoogleのGemini
https://gemini.google.com/app

　2045年にシンギュラリティが実現してもおかしくない状況になってきたと思っていますが、現在でもカメラからの映像をリアルタイムで処理して、話し出すようになればそれは殆ど人工知能と言ってよい状況だと思うのですが。

　元はゲームで2018年4月にアニメが放送された「シュタインズ・ゲート ゼロ」という作品のヒロインがそういう感じです。ヒロインのクリスは人工知能でスマートフォンの中にいて、スマートフォンのカメラから外界の映像を得て、主人公と会話をする。それが近い将来に実現すると思います。

　端的に言えば、男性の場合で言えば、「AI彼女」が近い将来に実現するということです。

　個人的に良いと思っているサービス。

・無料で音楽が聞ける
Spotify（スマートフォンで聞くとシャッフル再生しかできないけど、パソコンだとプレイリストの順番

通りに聞ける。無料アカウントだと広告がたまに入ります）

https://open.spotify.com/

・無料で利用できるインターネット・テレビ（知っている人も多いでしょうけど）

ABEMA

https://abema.tv/

TVER

https://tver.jp/

・無料で漫画が読める

ニコニコ静画

https://seiga.nicovideo.jp/manga/

コミックNEWTYPE

https://comic.webnewtype.com/

少年ジャンプ+

https://shonenjumpplus.com/

マガジン

https://pocket.shonenmagazine.com/

サンデー

https://www.sunday-webry.com/

カドコミ

https://comic-walker.com/

　スマートフォンの漫画アプリ。マンガ・アプリのランキングの上位10個のマンガアプリを全部入れてもよいです。大雑把に言えば、一つのマンガアプリにつき「（マンガを）一日に4話無料で読めます」というのならば、マンガアプリを10個スマートフォンとかタブレットに入れておくと、（マンガを）一日に40話無料で読めるようになります。

・国立国会図書館デジタルコレクション
https://dl.ndl.go.jp/
　著作権が切れた書籍を公開している。現在は書籍だけだけど、近い将来に雑誌も公開対象にすることを予定しているみたいです。
　個人的には、仏教についての重要な経典（漢訳された大蔵経を日本語に翻訳したもの）は恐らくは全部ここに無料で公開されていると思うのでそれがありがたいです。

> 大蔵経(だいぞうきょう)とは？ 意味や使い方 - コトバンク
仏教のすべての典籍を集録したもので，一切経・蔵経・三蔵聖経ともいう仏教の経（仏の教説集）・律（仏弟子の生活規範）・論（インド仏教学者による経の解釈）の３部（三蔵）を含む。 サンスクリット語系とパーリ語系に分かれ，前者には漢訳・チベット訳，後者にはビルマ訳・タイ訳などがある。 後者は『南伝大蔵経』として伝わる。

・無料ではないけど低価格で良いと思うサービス。

　新しかったり人気だったりする漫画を低価格で読む方法として、
レンタル・コミック
漫画喫茶・インターネットカフェ
　で読む。

ブックオフ・オンラインで注文した本とか雑誌とかをブックオフの店頭受け取りにすると一冊でも送料無料です。ブックオフ・オンラインは、書籍だけではなくて雑誌も扱っているので中古の雑誌が安くなっているのを近くのブックオフの店頭受取で送料無料で購入するのは結構良いと思います。あと、ブックオフはゲーム・ソフトとかDVD・ブルーレイディスクとかの映像ソフトとか、音楽CDも取り扱っています（個人的には10年くらい前に発売されたアニメのイベントとかライブのDVD・BDが500円くらいで売っていたりするのをたまに買います）。

・予測市場のポリマーケット

https://polymarket.com/

Wisdom of Crowds（群衆の叡智）・集合知による未来予測にして、「保険」を中央管理者なしで実現している（「過去のことはGoogleに聞け、未来のことはAugerに聞け」とインターネットで見たことがありますが、そのAuger（英語で「占い師」）の互換サービスです。）。

英語のサイトですがGoogleのクロム・ブラウザで開くとブラウザの標準機能で日本語に翻訳してくれます。

Wisdom of Crowds（群衆の叡智）については、ジェームズ・スロウィッキー著『「みんなの意見」は案外正しい』（角川書店）という本がすごく面白いです。

> 予測市場 (よそくしじょう、prediction market) とは、将来予測をするための先物市場である。

https://ja.wikipedia.org/wiki/%E4%BA%88%E6%B8%AC%E5%B8%82%E5%A0%B4

WizOdds 2024 アピール

by david wada

---

1. randomized parallel best first search

探索木を2手進めた状態から展開

単なる最良探索にランダム因子を加えてました。

末端ノードで少しだけ探索

2. evaluation function

apery wcs30 の予定

3. opening book

数年前しようした「やねうら王」から拝借

4. マシンは i9 14900K 単体の予定

nshogi アピール文章

中屋敷 太一

概要

- nshogiはAlphaZeroの手法をベースとしています．
- AlphaZeroの手法に加え，いくつかの工夫を行っています．
- 指し手生成や探索，評価関数の学習など，主なコンポーネントはフルスクラッチで実装しています．
- 中屋敷は去年までTeam Noviceのメンバーで出場していました．
  - 今年は，新たに，Noviceとソースコードを一切共有しないコンピュータ将棋ソフトウェアを実装しました．
  - それをnshogiとして，Noviceとは別にエントリーしています．
  - （技術的な交流などはTeam Noviceのメンバーと引き続き行っています．）
- 学習データにWCSC 33までのNoviceのデータを使用しているため，フルスクラッチ申請を行いません．

探索

- AlphaZeroと同様，pUCTを用いたMonte-Carlo Tree Searchを行っています．
- ゲーム木のリーフ局面の評価に，ニューラルネットワークによる評価に加え，Depth First Searchによる5手詰みの探索を行っています．
  - 探索スレッドとは別のCPUスレッドで詰み探索を行います．
- AlphaZeroがゲーム木内の各ノードの勝率を保持していることに加え，nshogiは各ノードの引き分け確率を保持しています．

評価関数

- 評価関数にはニューラルネットワークを用いています．
- 30 Block 256 Filter のResidual Networkの構成のニューラルネットワークで局面を評価しています．
- 現局面を入力に取り，その局面の勝率，次の一手の選択確率分布，引き分け確率，現局面の利きの4つを出力します．
  - 現局面の利きは学習時の補助タスクとしてのみ用い，対局時には用いません．
- 学習に542,709,386局面（重複あり）を用いています．

定跡

- floodgateの棋譜から生成しています．
- 勝率などの指標をもとに，定跡として登録する手を選んでいます．

その他

- 探索におけるメモリ確保/解放が頻発したため， segregated free listを用いたメモリ管理を行うことで，メモリ確保/解放のオーバーヘッドを減らしています．
- リーフ局面を展開する際に行われるメモリ確保を，ニューラルネットワークによる局面の評価中に，別スレッドにより行います．

Last updated 2024-03-29 20:25:01 +0900

恭祐 アピール文書 WCSC34

開発者 ペンギンクミマヌ

2024年3月10日　作成

プログラム情報

プログラム名:恭祐

プログラム名の由来:知人がどうしても自分の名前を冠して欲しいと言ってたので

特徴: dlshogiベース

初参加:WCSCは初参加　第4回電竜戦でコンピューター将棋大会デビュー

使用ライブラリ: dlshogi,cshogi “強い将棋ソフトの創りかた”に準じる

採用している手法: “強い将棋ソフトの創りかた”に準じる

探索部:ふかうら王

コメント

第4回電竜戦のときは,強い将棋ソフトの創りかたの付属データとAobaZeroの棋譜から評価関数を作成しました。強さは定跡なしでfloodgate 3600-3800程度でした。電竜戦前は、定跡で棋力をカバーできるかと思っていましたが、本戦で棋力の必要性を痛感しました。ネットで公開されている棋譜からこれ以上の棋力向上に限界を感じたので、電竜戦終了後から自前で教師局面生成に取り組んでいます(1日1万局面しか作れず😂)。4月頃からその教師局面を混ぜて学習させ,WCSC34で使う予定です。目標は2次予選進出です。

参考文献

・強い将棋ソフトの創りかた

・やねうらおうwiki

開発者情報

開発者名:ペンギンクミマヌ　(本名)山下公誠

SNS: X note

【はるか wcsc34 アピール文章】

自分自身が将棋プログラムを作ってみたいという興味の下で作成しています。そのためライブラリは使用せずにプログラムを作成しています。

評価部は手調整で作成し、探索部は$\alpha\beta$法を用いる予定です。

初出場なので楽しんで参加したいと思っています。

2024年3月 小林 航太

python-dlshogi2のニューラルネットワーク部に改変を加えました。

層を減らし、チャンネル数を192->384->768->1536に順々に増えていきます。

さらにそこにGhostModuleを加えました。入力データには全く同じデータのチャンネルが多く存在するのでGhostModuleと相性がかなり良いと思います。