



HoneyWaffle

第29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

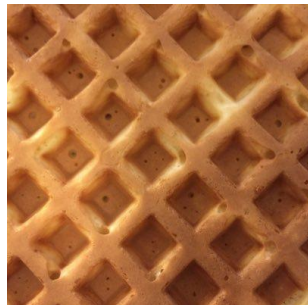
職業: プログラマー

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やニコニコ放送、AbemaTVの将棋中継が好き。



HoneyWaffle (ハニーワッフル) 名前の由来

- ・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う思いから命名しました。

※表紙やTwitterアイコンのワッフルはうちで焼いたものを使用しています。

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>

戦績ふりかえり(1)

2016年、Go言語でオリジナル開発版

- ・第26回世界コンピュータ将棋選手権 出場 一次予選2勝5敗
- ・第4回将棋電王トーナメント 出場 予選リーグ3勝5敗

盤面データを複素数平面で表現しているのが特徴。Go言語なのにゴルーチンを使いこなせず、シングルスレッド動作。

戦績ふりかえり(2)

2017年、やねうら王ライブラリ使用

- ・第27回世界コンピュータ将棋選手権 出場 決勝リーグ7位
- ・第5回将棋電王トーナメント 出場 決勝トーナメント初戦敗退

ここから振り飛車をテーマに。当初は、振り飛車にすれば対抗形になってじわじわとした渋い展開になる →手数が伸びる→最初から時間を温存しつつ時間攻め、という戦略がメイン。電王トーナメントでは、振り飛車をひいきした教師局面生成を行い、それを学習した評価関数を通常の評価関数とブレンドすることで振り飛車評価関数を作成。

戦績ふりかえり(3)

2018年、Apery、やねうら王ライブラリ使用

・第28回世界コンピュータ将棋選手権 出場 決勝リーグ8位

定跡、評価関数を振り飛車仕様にした上で、玉の美濃囲い落とし穴(28玉型にボーナス加点)を採用。

シードで二次予選からの参加。5勝3敗1分けでギリギリ決勝リーグに進出。決勝リーグでは1勝6敗も、唯一の勝ち局は振り飛車党の方にはぜひ御覧いただきたい。

定跡の出来がいまいちで、評価値つきで見ると見どころのなかった将棋が結構あったのが反省点。

ハードの面でも、AWSで選べる一番いいやつにするべきだった。

ポエム(1) WCSC28以降の状況(個人の感想)

NNUE型の評価関数のレート向上がすごいことに。

(参照: <https://www.qhapaq.org/shogi/>)

また、NNUEkaiシリーズのたややん氏の、振り飛車評価関数もすばらしい。これこそが期待されていたものなのでは。

→評価関数界限はとてつもないレッドオーシャン。海の生き物の名前も多いし。

→何か自信のある理論の裏付けがあるわけでもなく、機械学習のリソースも豊富ではないので、正直まともにぶつかり合うことすら難しい。仮にうまくいったとしても、評価関数だけで決定的な差をつけられるのか？

→というわけで評価関数は前回の KPPTのままで、それ以外のところで何とかできないか考える。

ポエム(2) AlphaZero完全に理解した(個人の感想)

将棋の知識なしでも膨大なリソースをうまく使うことで elmoを超えた、どやぁ(物量で殴ってるだけのようにも見える)。

千田七段(当時六段)の叡王戦パラダイス 12月号での棋譜解説は必見。電王盤の Ponanzaが拗ねたように起動しなくて、電王ぽんぽこと交代したのが、うける。

(<http://live.nicovideo.jp/watch/lv316751884>)

モンテカルロ? だからか、詰みロジックがない? からか、勝勢になっても詰ましにいかない。

先手なら初手 26歩固定なのが興味深い。

なお論文は全く読んでいない。

→学習の結果、NNの中に、現在の将棋ソフトにはない「あるもの」が実現されているように見える。

ポエム(3) 振り飛車について(個人の感想)

最近のソフトは平手初期局面の開始直後の数手で振り飛車は損だと見ていると、認めざるを得ない。対抗形にすると、勝敗問わず相居飛車よりも手数がかかりがちというのが影響しているのでは。仮に相居飛車が100手、対抗形が130手なら、例えば飛車先の歩を突く→勝ちまで1/100の価値の手、飛車を振る→勝ちまで1/130の価値の手。その数字に基づいて手を選んだ結果、自然と相居飛車の将棋になる、みたいな。手数が長いと不確定要素が増えて、短い将棋よりは勝ちにくくなるのでは？というも。

ただ、飛車を振っただけで一気に-200とか、そういった数字ははっきりとおかしいと思う。ソフトとしては、いくつかある手のうちでわずかでもいい手を選んだ結果が居飛車だと思うので、振り飛車の手は居飛車の手よりも評価値が1とか10とか低いだけであるはず。そもそも-200は振った飛車を戻せ、振り飛車指すよりも2手損した方がまだマシなことだから、単にそのソフトが振り飛車を指せない、振り飛車の局面をまともに評価できてない、というだけのことで、振り飛車という戦法が悪いわけではない。と強く主張したい。

コンピュータ将棋での振り飛車のメリットは、こちらはほぼ必ず対抗形になると分かっているところ。相居飛車の定跡にはまることはなく、逆にこちらの土俵に引きずり込むことができる。まあ居飛車党のソフトからしたら、わざわざ自分から評価値を下げる相手だと思っているのだろうけど。少数派の振り飛車対策をしている暇は正直ないだろうし、学習の時にわざわざ振り飛車の局面も作るの大変だろうし、そういうところを突くのが楽しみ。

正直なところ、振り飛車をコンピュータ将棋でやるのは、個性派揃いの振り飛車党棋士の方々の領地を重機で荒らすような、無粋な行為だと思って後ろめたい面があった。また、振り飛車をコンピュータで調べていって、振り飛車のすばらしさを追求するつもりが、逆にこういう理由で振り飛車はだめなんだよという結果になったらイヤだなあというものがあった。ただよく考えれば、自分ごときがちょっとやったところで何の影響もないし、そもそも将棋はそんなに浅くないだろう。ということでやれることは何でもやっていきたい。

コンセプト

・「自分の戦い方で、最強の振り飛車党ソフトを目指す」

今回、評価関数のバトルには参加しない。

AlphaZeroそのものの再現や実装はしない。

AlphaZeroが実現したであろう「あるもの」は振り飛車にとっても重要なので、そこだけピンポイントで実装を試みる。

「あるもの」は決勝リーグ進出した場合に出す文書で種明かしできたらいいな、ということで。

ハードはAWS EC2インスタンス1台(極力高性能なものを選択)の予定です。今回こそ一番いいのを。

構成(予定)

■開発部(探索)

WCSC28で使用したやねうら王^[1]ベースに、前述の「あるもの」を追加する^[2]予定

■評価関数

WCSC28で使用した、Apery^[3]ベースの振り飛車評価関数、進捗状況によってtanuki^[4]のNNUE型に手を出すかも？

■定跡

WCSC28で使用したファイルをベースに、WCSC28後から5か月くらいかけて手動で作り込んだものを使用

■ライブラリ採用理由

[1]やねうら王: 最高の探索、フレームワーク2017年から使用している。[2]python-shogi: python使うので。

[3]Apery: KPPT最高クラスの評価関数。振り飛車との相性も悪くない。[4]tanuki-: 最新最強の評価関数。

最後まで読んでいただき、ありがとうございます。