

WCSC28 elmoアピール文書

1. elmoについて

elmoは主に評価関数に改良を加えたソフトです。
Apery/やねうら王を主に利用しています。

<以前の内容>

WCSC26: 自己対局の勝率に基づいて評価値生成
→とても弱かった

WCSC27: 自己対局時の勝敗と深く読んだ時の評価値を用いて評価値を更新、
大規模(50億局面)学習
→優勝：やねうら王/Apery等で採用！

2. 評価関数

昨今のコンピュータ将棋は500点程度差がつくと逆転するのは難しいので、
序盤を含めた評価値の近い「競っている局面」を手厚く学習する方向で進めています。
学習部分の変更も色々試しているのですが、良くも悪くもあまり変わらず悩ましいです。

3. 定跡生成

定跡抜けたら負け確定でしたとかツライじゃないですか。
勝敗への影響が大きい割に今までなおざりにしていたので、
評価関数と同じような方法で作るようにしました。
(自己対局→勝敗+評価値を利用して評価)

4. 利用ライブラリ

やねうら王:

対局時と定跡生成で利用しています。定跡に互換性が無いので少し手を入れています
採用理由：探索部分が優秀なため。既存拡張機能の流用

Apery:

評価関数生成と定跡生成で利用しています。

また、評価関数もAperyのものをベースに更新しています。

採用理由：評価関数(および学習部)が優秀なため。既存拡張機能の流用

Qhapaq/tanuki-:

評価関数を混ぜて使おうかと思っています(多分使います)。

採用理由：評価関数が強い

5. その他

ビッグウェーブに乗れませんでした

コンピュータ将棋ソフト 「大合神クジラちゃん」

第 28 回世界コンピュータ将棋選手権 アピール文書

2018/04/11 鈴木雅博、大賀一貴

1. 大合神クジラちゃんの紹介

マスタ／スレーブの構成を取り、マスタがスレーブを管理しながらクラスタリング探索を行います。スレーブを動かすのは主にニコニコ生放送や YouTube 視聴者のパソコンを考えています。前回の選手権ではパソコン 400 台以上のクラスタで動かしました。

クラスタの基本的な手法は GPS 将棋を参考にしています。マスタが各指し手をスレーブに展開することでより読みを深くしています。また、冗長性を持たせるため同じ手を 2 つのスレーブに探索させています。

2. 独自の工夫

独自の工夫として、探索を探索ツリーの末端ノードだけでなくすべてのノードで行うことで、スレーブの性能差を気にすることなく安定した棋力向上を可能にしました。過去のシステムでは末端でないノードでは探索させず、末端ノードのみに指し手を展開し、展開できなかった手はそれ以外の手を探索する「その他ノード」に探索させるようにしていました。しかしながらその手法では、ある指し手とそれ以外の指し手の探索量が大幅に違った場合、なにが最善手であるか判断するのが極めて難しいという問題がありました。具体的には、評価値が 1000 点・探索深さ 14 という手と、評価値が 900 点・探索深さ 20 という手があった場合、どちらが良い手なのか判断できないという問題がありました。

現在のシステムでは、末端以外の親ノードでも探索を行うようにしこの不安定性を少し減らすことに成功しました。例えば上記の例であれば、末端ノードの探索結果（指し手 A：評価値が 1000 点・探索深さ 14、指し手 B：評価値が 900 点・探索深さ 20）に加え、親ノードの探索結果が加わります。例えばその結果が評価値 950・探索深さ 18 で指し手が A だったとすると、探索深さが深くなった場合でも指し手 A が有力ということがわかります。

またこのシステムはビンゴマスターさんのアドバイスをもとに作られたものなので、ビンゴマスターさんに深く感謝申し上げます。

ディープラーニングによる指し手探索もやろうと考えていますが、現状（4/11）全く手つかずです。もし間に合いそうであればクラスタの指し手の初期探索部分を DL で行いたいと考えています。

3. その他

- 探索部分に YaneuraOu、評価関数は既存のものを追加学習したバージョンを使用しております。
- 非公開協力者として、まふさんに定跡ファイルを作って頂く予定です。
- ディープラーニングによる指し手探索もやろうと考えていますが、現状 (4/11) 全く手つかずです。もし間に合いそうであれば指し手の初期探索部分を DL で行いたいと考えています。

4. ライブラリの選定理由について

- YaneuraOu
 - ✧ クラスタのスレーブ部分の基礎として使用。非常に強く探索部分も優秀なため使わせて頂いています。
- Apery
 - ✧ クラスタのマスター部分の将棋ライブラリとして使用。探索は行わず合法手の確認や指し手生成に使っています。YaneuraOu に変えてもいいのですが、昔から使っているのでも今も使用しています。
- 人造棋士 18 号
 - ✧ 関連ツールを使用させて頂く可能性があるため登録しています。
- dlshogi
 - ✧ マスターの初期探索部分をディープラーニングで行おうかと考えているので、念のため登録しております。ただし現在は全く手がついておりません。

5. 過去の戦績

- 第 23 回コンピュータ将棋選手権 27 位 (独創賞を獲得)
- 第 24 回コンピュータ将棋選手権 16 位
- 第 25 回コンピュータ将棋選手権 19 位
- 第 27 回コンピュータ将棋選手権 4 位

6. 完全オリジナルの GUI もあります。

- コンピュータ将棋を動かすための GUI も自作しています。
- クジラちゃんを擬人化し、GUI 内に取り込みました。
- URL: <http://garnet-alice.net/programs/whalewatcher/>
- ニコニココミュニティ: <http://com.nicovideo.jp/community/co516151>

コンピュータ将棋の概念を打ち砕き
進化させるべく三駒関係を封印する

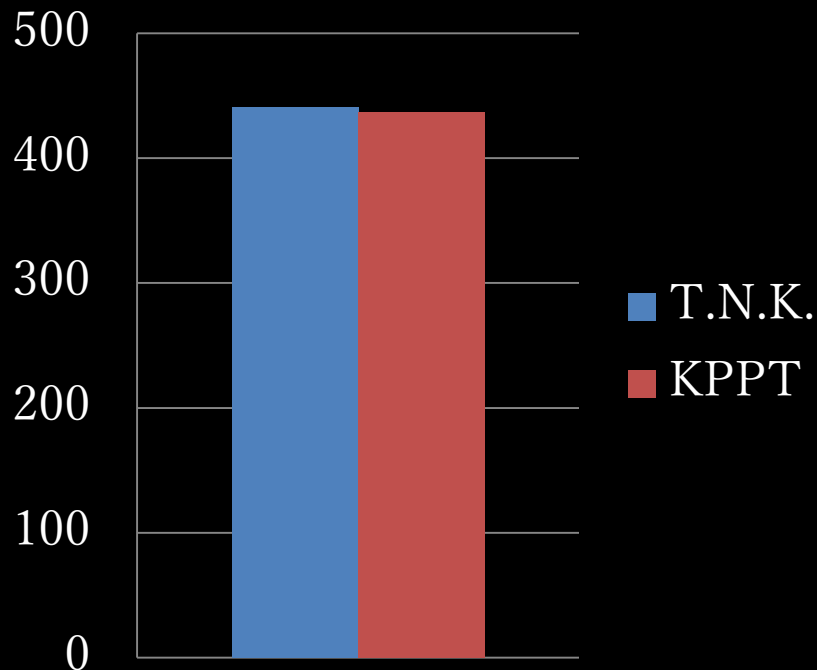
高速な差分計算を特徴とし
三駒関係と同等の NPS を実現する
ディープラーニング評価関数を搭載

the end of genesis
T.N.K.evolution turbo type D

ザイオソフト
コンピュータ将棋サークル
野田久順 岡部淳 鈴木崇啓
那須悠 河野明男

ベンチマーク

(万NPS)



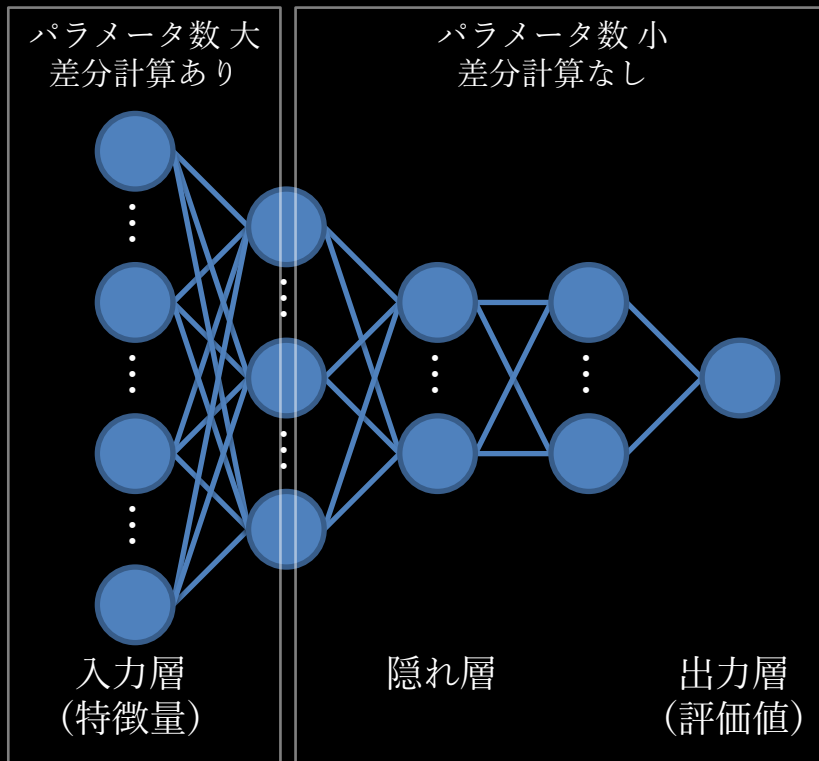
- 測定環境

- やねうら王ベンチマーク
- CPU: Core i7 6700K
- 置換表サイズ: 16GB
- スレッド数: 8
- トーナメント版
- NPSはやねうら王ベンチマークに収録されている3局面のNPSの平均値

CPUによる演算

- T.N.K. のディープラーニング評価関数はGPUを使わず、CPUで1局面ずつ評価します
 - $\alpha\beta$ 探索ベースの探索ルーチンにそのまま組み込むことができます
- パラメータを整数化し、SIMD演算で高速化しています

差分計算



- KPに相当する特徴量を入力とする全結合ニューラルネットワークです
- 入力のアフィン変換を差分計算で高速化しています
 - 二乗関係の差分計算をベクトルに拡張して適用しています

使用ライブラリ

- やねうら王

- 用途

- エンジンの基礎部分として使用

- 選定理由

- レーティングの高さ
 - 改造のしやすさ

- Apery

- 用途

- 学習データ生成時の評価関数として使用

- 選定理由

- レーティングの高さ

高速に差分計算可能なニューラルネットワーク型将棋評価関数

那須 悠[†]

[†] ザイオソフト コンピュータ将棋サークル

2018 年 4 月 28 日

概要

現在のコンピュータ将棋ソフトの多くで使用されている評価関数である三駒関係は、高速な計算が可能であるが、非線形な評価を行うことができない。本稿では、CPU で高速に動作するニューラルネットワーク評価関数、「NNUE 評価関数」(Efficiently Updatable Neural-Network-based evaluation functions) を提案する。NNUE 評価関数は、CPU での演算に最適化した設計と、差分計算を始めとする高速化技術により、三駒関係と同等の実行速度で動作する非線形評価関数である。NNUE 評価関数を使用する初めての将棋ソフト『the end of genesis T.N.K.evolution turbo type D』は、第 28 回世界コンピュータ将棋選手権に参加を予定している。

NNUE: Efficiently Updatable Neural-Network-based Evaluation Functions for Computer Shogi

Yu Nasu[†]

[†] Ziosoft Computer Shogi Club

April 28, 2018

Abstract

Most of the strongest shogi programs nowadays employ a linear evaluation function, which is computationally efficient but lacks nonlinear modeling capability. This report presents a new class of neural-network-based nonlinear evaluation functions for computer shogi, called NNUE (Efficiently Updatable Neural-Network-based evaluation functions). NNUE evaluation functions are designed to run efficiently on CPU using various acceleration techniques, including incremental computation. The first shogi program with a NNUE evaluation function, *the end of genesis T.N.K.evolution turbo type D*, will be unveiled at the 28th World Computer Shogi Championship.

0. まえがき

本稿は、第 28 回世界コンピュータ将棋選手権 [1] に参加を予定しているソフト『the end of genesis T.N.K.evolution turbo type D』のアピール文書の一部である。同ソフトの評価関数に関する技術を論文風の体裁で説明するが、実験は掲載しない。同ソフトの具体的な棋力にも本稿では言及しないので、選手権の結果を参照されたい。

1. はじめに

コンピュータ将棋ソフトの指し手の決定は、主として形勢判断に相当する「評価関数」と、読みに相当する「探索」によって行われる。高い棋力を実現するためには、評価値の精度が高く、高速に計算できる評価関数を用いることが望ましい。評価値の精度が高ければ正確な形勢判断ができ、計算が高速であればより深く読むことができるためである。

2018 年 3 月現在、コンピュータ将棋ソフトで使われる最も代表的な評価関数の方式は、「三駒関係」と呼

ばれる線形モデルである。機械学習によって駒 3 つの位置関係に重みを付けておき、局面上に出現する位置関係に付けられた重みの総和を評価値とする。2003 年に初めてアイデアが示され [2]、2009 年に玉を含む組み合わせに限定して実装した『Bonanza』 [3] のソースコードが公開されて以降、広く用いられるようになった。2014 年に『NineDayFever』 [4] で導入された手番評価など、いくつかの拡張を経て、トップクラスの棋力を持つ将棋ソフトのほとんどで現在も採用されている。

三駒関係の優位性は、膨大な数のパラメータによって表現される評価関数でありながら、高速に評価値を算出できる点にある。ある局面から 1 手指した先の局面の評価値を求めるとき、移動した駒に関する重みを元の局面の評価値に足し引きする差分計算によって、全体を計算した場合と同じ結果を、遥かに効率的に求めることができる。差分計算の容易さは、評価関数に線形モデルを採用することによってもたらされる大きな利点である。

しかし、線形モデルであることは、評価関数の表現能力の足枷にもなっている。特定の駒の位置関係が、ある状況においては有利に働くが、別の状況では無駄になる、といった非線形な評価を、三駒関係では直接表現することができない。線形モデルのまま表現能力を高めるためには、評価項目を変える必要があるが、2018 年 3 月現在、三駒関係より優れた線形モデルは知られていない。

より高い表現能力を持つ評価関数の実現を狙ったアプローチとして、非線形モデル、特にニューラルネットワークの利用も試みられている。先駆けとなったのは『習甦』 [5] で、白玉および相手玉の安全度によって駒の価値を重み付けした、非線形な評価関数を特徴とする将棋ソフトである。近年では、コンピュータ囲碁において、大規模な CNN (Convolutional Neural Networks) による評価関数を用いたソフトが成功を収めた [6] ことから、コンピュータ将棋でも CNN を利用する試みが行われている。しかし、CNN による評価関数を活用するためには、線形モデルよりも大きな計算リソースが必要となることが課題である。CNN を利用する将棋ソフトのほとんどは、並列演算を得意とする GPU を使用し、複数の局面をまとめて評価するバッチ処理を行って高速化している。それにもかかわらず、2018 年 3 月現在、ニューラルネットワークに最適化された特殊なハードウェアを使用する『AlphaZero』 [7] を除いて、トップクラスの将棋ソフトに比肩する棋力を実現した例は報告されていない。

本稿では、CPU で高速に動作するニューラルネッ

トワーク評価関数を提案する。CPU での演算に最適化した設計と、差分計算を始めとする高速化技術により、複数の隠れ層を持つニューラルネットワーク評価関数を、三駒関係と同等の実行速度で動作させることができる。以降、この方式の評価関数を「NNUE 評価関数」 (Efficiently Updatable Neural-Network-based evaluation functions) と呼ぶことにする。

2. NNUE 評価関数

NNUE 評価関数は、GPU もバッチ処理も用いず、CPU で 1 局面ずつ評価するニューラルネットワーク評価関数である。三駒関係と同様、1 局面を評価するまでのレイテンシが小さく、枝刈りを伴うミニマックス系の探索手法と組み合わせやすい。

『the end of genesis T.N.K.evolution turbo type D』に搭載する NNUE 評価関数の模式図を図 1 に示す。CPU で高速に計算するために設計した構造であり、『AlphaZero』などで採用されているニューラルネットワークの構造とは大きく異なる。

以下では、NNUE 評価関数の設計と高速化技術について述べる。

2.1 全結合ニューラルネットワーク

将棋盤の二次元構造を利用する CNN ではなく、全結合ニューラルネットワークを使用する。主な利点は、メモリアクセスパターンが単純で高速化しやすいことと、後述する差分計算が容易に実現できることの 2 点である。

メモリアクセスパターンは、三駒関係より処理効率が低い評価関数を設計するために重要な要素である。三駒関係では、評価値を算出するためにかかる時間の大半を、メモリへのランダムアクセスが占めている。効率的なメモリアクセスパターンを用いれば、同じ時間でより複雑な計算が可能になる。

全結合ニューラルネットワークによる評価値の計算を次式に示す。入力となる特徴ベクトルを $\mathbf{x}$ 、隠れ層の数を L 、層 l の重み行列を $\mathbf{W}_l$ 、層 l のバイアスベクトルを $\mathbf{b}_l$ 、活性化関数を σ とすると、評価値 y は

$$[y]_{1 \times 1} = \mathbf{z}_{L+1} \quad (1)$$

$$\mathbf{z}_l = \mathbf{b}_l + \mathbf{W}_l \mathbf{a}_{l-1} \quad (2)$$

$$\mathbf{a}_l = \begin{cases} \sigma(\mathbf{z}_l) & (\text{if } l > 0) \\ \mathbf{x} & (\text{if } l = 0) \end{cases} \quad (3)$$

と求められる。

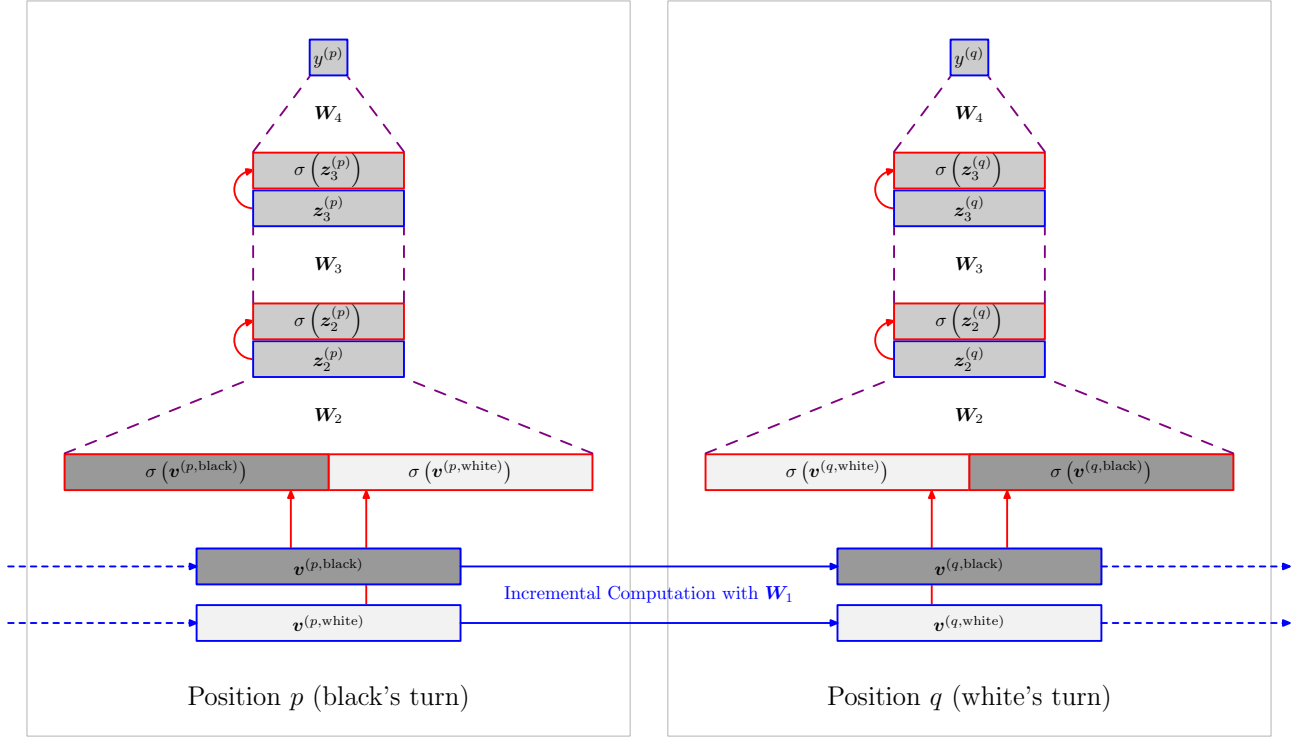


図1 『the end of genesis T.N.K.evolution turbo type D』の評価関数. 局面 p から 1 手指した先の局面 q について評価値を求めるとき, 処理の一部で差分計算を利用する. 評価には手番が考慮されている.

2.2 活性化関数

活性化関数には, 層 l のユニット数を N_l として, 次式で表される clipped ReLU を用いる.

$$\begin{aligned} \sigma(z_l) &= \sigma \left(\begin{bmatrix} z_{l,1} \\ z_{l,2} \\ \vdots \\ z_{l,N_l} \end{bmatrix}_{N_l \times 1} \right) \\ &= \begin{bmatrix} \sigma(z_{l,1}) \\ \sigma(z_{l,2}) \\ \vdots \\ \sigma(z_{l,N_l}) \end{bmatrix}_{N_l \times 1} \\ \sigma(z_{l,i}) &= \begin{cases} 0 & (\text{if } z_{l,i} \leq 0) \\ z_{l,i} & (\text{if } 0 < z_{l,i} < 1) \\ 1 & (\text{if } z_{l,i} \geq 1) \end{cases} \end{aligned} \quad (4)$$

値域が有限であるため整数化に適しており, 後述する SIMD 演算によって効率的に計算できることが利点である.

2.3 特徴量

特徴ベクトルには, 原則として, 各要素が 0 または 1 の値を取る二値ベクトルを使用する. この制約も差分計算を容易にするためである. 高速に差分計算を行うため

には, 1 手指す前後の局面で値が変化する要素の数が少ないことが望ましい.

このような性質を満たす特徴量の例としては, 自玉または敵玉と, 玉以外の駒 1 つとの位置関係を表す KP (King-Piece) が挙げられる. 『the end of genesis T.N.K.evolution turbo type D』で実際に採用している特徴量は, KP をベースとして改変したものである. 詳細は後述する.

2.4 アフィン変換とメモリレイアウト

式 (2) のようなアフィン変換を行うためには, 通常, 行列の行とベクトルの内積を計算するのが効率的である. W_l の i 行目を $W_l(i, :)$ として,

$$z_l = \begin{bmatrix} z_{l,1} \\ z_{l,2} \\ \vdots \\ z_{l,N_l} \end{bmatrix}_{N_l \times 1} \quad (6)$$

$$z_{l,i} = b_{l,i} + W_l(i, :)\mathbf{a}_{l-1} \quad (7)$$

と計算する. 行列の行の要素を連続して参照するメモリアクセスパターンであるから, メモリレイアウトは row-major とする.

しかし, ベクトル $\mathbf{a}_{l-1}$ がスパースである (要素の大

部分が 0 である) 場合は, 行列の一部の列だけを用いて, 次式のように計算する方が高速になる. $\mathbf{W}_l$ の j 列目を $\mathbf{W}_l(:, j)$ として,

$$\mathbf{z}_l = \mathbf{b}_l + \sum_{j \in \{j | a_{l-1,j} \neq 0\}} a_{l-1,j} \mathbf{W}_l(:, j) \quad (8)$$

と計算する. この場合は, 行列の列の要素を連続して参照するメモリアクセスパターンになり, メモリレイアウトを column-major とする方が効率が良い.

NNUE 評価関数では, 特徴ベクトル $\mathbf{x}$ のアフィン変換

$$\mathbf{z}_1 = \mathbf{v} \quad (9)$$

$$\mathbf{v} = \mathbf{b}_1 + \mathbf{W}_1 \mathbf{x} \quad (10)$$

を, 式 (8) の方法によって計算する. 前述の通り, $\mathbf{x}$ は二値ベクトルである. $\mathbf{x}$ がスパースであれば, 式 (8) によってアフィン変換を高速に計算できる^{*1}. さらに, 二値ベクトルであることを利用すると, 式 (8) において $a_{l-1,j} = x_j \in \{0, 1\}$ であるので,

$$\mathbf{v} = \mathbf{b}_1 + \sum_{j \in \{j | x_j = 1\}} \mathbf{W}_1(:, j) \quad (11)$$

と単純化できる.

2.5 差分計算

差分計算は, ある局面から 1 手指した先の局面の評価値を計算する際に, 元の局面との差異が少ないことを利用して高速化する方法である. NNUE 評価関数では, 局面 q の特徴ベクトル $\mathbf{x}^{(q)}$ をアフィン変換した結果のベクトル

$$\mathbf{v}^{(q)} = \mathbf{b}_1 + \mathbf{W}_1 \mathbf{x}^{(q)} \quad (12)$$

を, 1 手前の局面 p について計算済みの $\mathbf{v}^{(p)}$ を利用して差分計算する. 具体的には, 次式によって $\mathbf{v}^{(q)}$ を求める.

$$\begin{aligned} \mathbf{v}^{(q)} = \mathbf{v}^{(p)} - & \sum_{j \in \{k | x_k^{(p)} = 1 \wedge x_k^{(q)} = 0\}} \mathbf{W}_1(:, j) \\ & + \sum_{j \in \{k | x_k^{(p)} = 0 \wedge x_k^{(q)} = 1\}} \mathbf{W}_1(:, j) \end{aligned} \quad (13)$$

これは, 線形モデルの差分計算をベクトルに拡張した式であると解釈することができる.

^{*1} 実際には差分計算を行うので, 1 手指す前後の特徴ベクトルの変化 (ハミング距離) が小さいことが重要であり, スパースである必要はない.

差分計算を利用するのは特徴ベクトルのアフィン変換だけで, それ以外の部分では利用しない.

2.6 手番評価

将棋においては, 駒の配置が同じであっても, 手番がどちらにあるかによって形勢は異なるので, 評価関数も手番を考慮する方が高精度になる. 三駒関係では, 線形性を利用して評価値を 2 つの項に分解し, 手番に依存しない項と手番に依存する項の和で表現している [4].

NNUE 評価関数では, 常に手番側の視点から見た局面を評価することによって手番評価を実現する. まず, 単純な方法について説明する. 局面 p の手番側プレイヤーを $\text{active}(p)$, プレイヤー c から見た局面 p の特徴ベクトルを $\mathbf{x}^{(p,c)}$ として,

$$\mathbf{z}_1^{(p)} = \mathbf{b}_1 + \mathbf{W}_1 \mathbf{x}^{(p, \text{active}(p))} \quad (14)$$

とすると, 手番側の視点から見た局面の評価が実現できる.

手番がどちらにあるかによって特徴ベクトルが変わるため, 差分計算は, 先手 (black) および後手 (white) の両方の視点について行う必要がある. 1 手指すごとに, $c \in \{\text{black}, \text{white}\}$ に対して, それぞれ

$$\mathbf{v}^{(p,c)} = \mathbf{b}_1 + \mathbf{W}_1 \mathbf{x}^{(p,c)} \quad (15)$$

を差分計算しておく. 評価値の計算には手番側を使用し,

$$\mathbf{z}_1^{(p)} = \mathbf{v}^{(p, \text{active}(p))} \quad (16)$$

とする.

次に, この方法を拡張して, 非手番側のベクトルも評価値の計算に活用する方法を説明する. 局面 p の非手番側のプレイヤーを $\text{opponent}(p)$ として, 式 (16) の代わりに次式を用いる.

$$\begin{aligned} \mathbf{z}_1^{(p)} &= \begin{bmatrix} \mathbf{v}^{(p, \text{active}(p))} \\ \mathbf{v}^{(p, \text{opponent}(p))} \end{bmatrix}_{2N_1 \times 1} \\ &= \begin{cases} \begin{bmatrix} \mathbf{v}^{(p, \text{black})} \\ \mathbf{v}^{(p, \text{white})} \end{bmatrix}_{2N_1 \times 1} & (\text{if } \text{active}(p) = \text{black}) \\ \begin{bmatrix} \mathbf{v}^{(p, \text{white})} \\ \mathbf{v}^{(p, \text{black})} \end{bmatrix}_{2N_1 \times 1} & (\text{if } \text{active}(p) = \text{white}) \end{cases} \end{aligned} \quad (17)$$

すなわち, 手番側と非手番側の特徴ベクトルをそれぞれアフィン変換し, 結合したベクトルを $\mathbf{z}_1^{(p)}$ とする. アフィン変換のパラメータ $\mathbf{b}_1, \mathbf{W}_1$ は共有されるので, こ

の計算は手番方向の畳み込みであると解釈できる。

図 1 に示した『the end of genesis T.N.K.evolution turbo type D』の評価関数は、式 (17) の方式を採用している。

2.7 HalfKP 特徴量

前述の通り、『the end of genesis T.N.K.evolution turbo type D』では、二駒関係の一種である KP を改変した特徴量を採用している。KP が「自玉または敵玉と、玉以外の駒 1 つとの位置関係」を表す特徴量であるのに対して、敵玉を含む位置関係を使用せず、自玉を含む位置関係だけを使用する。以下では、この特徴量を「HalfKP」と呼ぶことにする。

HalfKP 特徴量は、前項で説明した、手番側と非手番側の特徴ベクトルを両方とも使用する手法と組み合わせるための特徴量である。HalfKP 特徴量を使用すると、式 (17) において、 $\mathbf{v}^{(p, \text{active}(p))}$ は「手番側から見た、手番側の玉に関する KP」に相当する情報を持ち、 $\mathbf{v}^{(p, \text{opponent}(p))}$ は「非手番側から見た、非手番側の玉に関する KP」に相当する情報を持つことになる。従って、特徴量そのものには敵玉の位置に関する情報を含まないが、手番側から見た特徴量と、非手番側から見た特徴量の両方を使用することによって、局面全体の情報を表現することができる。通常の KP を特徴量として使用する場合と比較して、同等の情報を表現するために必要な計算コストが小さくなることが利点である。

2.8 整数 SIMD 演算

SIMD (Single Instruction Multiple Data) 演算は、同一の処理を複数のデータに同時に適用する演算である。NNUE 評価関数は、各部分において整数の SIMD 演算を活用して高速に計算できるように設計されている。

特徴ベクトルのアフィン変換は、差分計算を行う場合も行わない場合も、ベクトルの加減算で求められる。重み行列 $\mathbf{W}_1$ を 16-bit の整数に変換して保持し、16-bit の符号付き整数ベクトルの加減算を行う命令 (AVX2 では VPADDW, VPSUBW) で計算する。

特徴ベクトル以外のアフィン変換では、直前の層のアクティベーション $\mathbf{a}_{l-1}$ と重み行列 $\mathbf{W}_l$ を 8-bit で表現し、8-bit 整数ベクトル 2 つの積和を求める命令 (VPMADDUBSW) などを用いて内積を計算する。

活性化関数は、整数のベクトルのビット幅を変換する命令 (VPACKSSDW, VPACKSSWB) と、8-bit 整数ベクトル 2 つの要素ごとの最大値を取る命令 (VPMAXSB) などによって計算する。

表 1 『the end of genesis T.N.K.evolution turbo type D』の評価関数で使用する重み行列のサイズ。

	列数	行数
$\mathbf{W}_1$	125388	256
$\mathbf{W}_2$	512	32
$\mathbf{W}_3$	32	32
$\mathbf{W}_4$	32	1

2.9 モデルサイズ

表 1 に、『the end of genesis T.N.K.evolution turbo type D』の評価関数で使用する重み行列のサイズを示す。列数と行数が、それぞれアフィン変換の入力と出力の次元数に対応する。このモデルサイズは、単位時間あたりに読める局面数が、評価関数に三駒関係を使用した場合と同等になるように設定したものである。

全パラメータ数の大半を、特徴ベクトルのアフィン変換に使う重み行列 $\mathbf{W}_1$ が占めている。この部分は差分計算を行うため、実際の計算コストは大きくない。 $\mathbf{W}_2, \mathbf{W}_3, \mathbf{W}_4$ を使うアフィン変換は差分計算ができないが、パラメータ数を比較的少なく設定し、8-bit 整数の SIMD 演算によって高速な計算を可能にしている。

3. おわりに

本稿では、ニューラルネットワークを用いた将棋の評価関数を提案し、CPU で高速に動作させるための設計と高速化技術について述べた。本技術による評価関数を搭載する初めての将棋ソフト『the end of genesis T.N.K.evolution turbo type D』は、第 28 回世界コンピュータ将棋選手権 [1] に参加を予定している。

また、本技術を実装したソースコードを後日公開する予定である。オープンソースの将棋ソフト『やねうら王』[8] をベースとして C++ で実装したもので、本稿で述べた内容に加えて、以下のような特徴がある。

- クラステンプレートを利用した、特徴量やネットワーク構造のカスタマイズ性
- コンパイル時計算やコンパイラによる最適化を活用した高速な実装
- 三駒関係の学習で用いられる手法と同様の、
 - － 静止探索と組み合わせた学習
 - － 特徴量の分解による学習時のパラメータ共有

特徴量やネットワーク構造は、『the end of genesis T.N.K.evolution turbo type D』で採用したものが最適

であるとは限らない。より優れた構造の検討は今後の課題であり、ソースコードを拡張して利用される他の開発者にも期待したい。

本技術がコンピュータ将棋の更なる発展に繋がれば幸いである。

参考文献

- [1] 第 28 回世界コンピュータ将棋選手権. <http://www2.computer-shogi.org/wcsc28/>.
- [2] 金子知適, 田中哲朗, 山口和紀, 川合慧. 駒の関係を利用した将棋の評価関数. ゲームプログラミングワークショップ 2003 論文集, pp. 14–21, 2003.
- [3] Kunihito Hoki and Tomoyuki Kaneko. Large-scale optimization for evaluation functions with minimax search. *Journal of Artificial Intelligence Research*, Vol. 49, No. 1, pp. 527–568, 2014.
- [4] 金澤 裕 治. NineDayFever アピール文書. <http://www.computer-shogi.org/wcsc24/appeal/NineDayFever/NDF.txt>, 2014.
- [5] 竹内章. コンピュータ将棋における大局観の実現を目指して. 人工知能学会誌, Vol. 27, No. 4, pp. 443–448, 2012.
- [6] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, Vol. 529, No. 7587, pp. 484–489, 2016.
- [7] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. [arXiv:1712.01815v1](https://arxiv.org/abs/1712.01815) [cs.AI], 2017.
- [8] 磯崎元洋. やねうら王オープンソースプロジェクト. http://yaneuraou.yaneu.com/yaneuraou_mini/.

読み太 アピール文書

2018/3/30 塚本隆三

自己紹介

- ・世界コンピュータ将棋選手権は第26回から参加させて頂いております。
- ・開発自体は4年前から行っています。プログラミングを覚えたのもそれくらいからです。
- ・毎回ノートパソコン1台で出場していました。
- ・ノートパソコンでの出場を期待されている方は、ごめんなさい。今回は違います。



評価関数

- ・3駒関係+手番評価(KPPT型)です。
- ・第5回電王トーナメントで使った評価関数に追加学習を行いました。
- ・depth8で200億局面の教師局面を作りました。
- ・現時点でWCSC27のelmoに勝率75%程度の強さです。



疎結合並列探索

- ・疎結合並列探索(クラスタ並列探索)を行います。
- ・Amazon EC2 を利用します。c4.8xlargeを16台借りる予定です。
- ・並列化アルゴリズムに関しては「技巧」を非常に参考にさせて頂きましたので、ライブラリとして申請させて頂きました。
- ・「技巧」は並列化部分のソースを公開している唯一のプログラムであり、並列化に限らず、将棋プログラムとしての設計が非常にきれいで私の好みであることが選定理由です。

Byteboard

- ・Byteboardというデータ構造を使っていますが、現時点ではBitboardよりも高速化できていません。どうやらByteboard本家である「たこっと」とは少し違った実装をしていたようです。
- ・指し手生成にByteboardを使ってはいけないそうです。びっくりしました。
- ・大会までに余裕があれば使えるようにします。(Byteboardで高速化できる部分はボトルネックになっている部分ではないので、どうしても後回しになってしまっています)



その他

- ・Linuxでの開発に戸惑っています。
- ・AWSの使い方が難しいです。
- ・学校を卒業したので、もうパソコンが借りることができなくなりました。今まで快く貸してくださって、近畿職業能力開発大学校の先生方には本当に感謝しています。おかげでとても強くなりました。



最後に

- ・読み太はStockfish、やねうら王、Apery、技巧を参考にさせて頂いております。感謝申し上げます。
- ・ハードウェアが強くなった分、もう言い訳はできません。全力の読み太にご期待ください。



HoneyWaffle

第28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

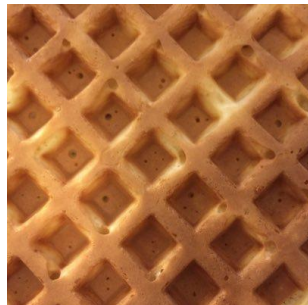
職業: プログラマー

棋力: 将棋ウォーズで3級、ぴよ将棋でR700-800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いな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

2016年、Go言語でオリジナル開発版

- ・2016年5月 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
- ・2016年10月 第4回将棋電王トーナメント 出場 予選リーグ3勝5敗

2017年、やねうら王ライブラリ使用

- ・2017年5月 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
- ・2017年11月 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

本題に入る前にポエム①

今回のコンセプトの前に、将棋とは何かをもういちど考える。将棋とは？

> 二人が「盤」の上で交互に「駒」を動かします。そして相手の「玉将(ぎょくしょう)」という駒を先に捕獲したほうが勝ちとなります。

(日本将棋連盟サイトから引用 <https://www.shogi.or.jp/knowledge/about/>)

→先に捕獲するとは、先に攻めるということ？

→先に攻めるのが有利ということ？

→そう考えるのが自然か。

本題に入る前にポエム②

ここ最近の将棋(コンピュータ将棋や、それに影響を受けたプロの将棋)は、確かに先に攻めようとする傾向がさらに強くなったかも。

- ・玉は1手2手動かすだけで、金銀もあまり玉に近づけずバランス重視(角換わりとか横歩取りとか)
- ・じっくりした矢倉が減った
- ・早く攻めるために桂馬も早く跳ねる

今現在の3駒関係の学習も、要は「早く勝ちやすい形を高く評価する」もの (現局面の評価値を、何手か先の局面の評価値に近づくよう学習する。勝った局面はより高く評価するよう学習する。)

本題に入る前にポエム③

「早く攻める」のを重視する前提なら、なるほど振り飛車は不利だと言わざるを得ない。少なくとも飛車を振る手で1手かかっているのは事実。工夫しないで学習回すと、飛車振らなくなっていくのも納得。

ただ、逆に、将棋の勝敗そのもので不利なわけではないよねと。純粹に手数がのびたとしても、最後に相手よりも1手早いだけでいいから。

先に攻めるのは先に詰ます可能性もあるけど、先にカウンターを食らう、先に攻めが切れるという恐れもあるよねと。

相掛かりで5手目に▲2四歩から無理やり攻めようとする、先に歩を手にした後手がある利になる変化。他の戦型でも極限まで攻めを早くしていったら、それと似たような局面ばかりになるかもしれない。

本題に入る前にポエム④

現在の潮流(居飛車が多い):最速の攻めを目指す。攻撃が最大の防御。

→相居飛車でお互いに最速の攻めを目指すのは、戦型にこだわらない陣営が自然と当たり前になっている。

→ならば、それに対抗できる振り飛車のソフトはかくあるべし!

- ・最速の攻めが来てもギリギリ受けられる受けの力(構想、囲い)

- ・隙を見て豪快に攻め、鮮やかに1手違いで勝つ

定跡や変な評価関数でただ飛車を振るだけでは真の「振り飛車」ではない。1手遅れたあげくに、中途半端に囲って、それから最速の攻めを目指してもね…。この辺は割と当たり前というか基本のはずなのにどうして…。

コンセプト

・「公開されている技術を最大限に活用し、最強の振り飛車党ソフトを目指す」

実は第5回将棋電王トーナメントの時と同じですが、意味合いはポエムで述べた通りまったく違います。去年一年、飛車を振らせるのに苦労したのは確かなのですが、それが精一杯で、考えが全然足りませんでした。

一般的に形勢判断の基準は①駒の損得②駒の働き③玉の堅さ④手番と言われていますが、現状の3駒関係では、①②④の評価がとても洗練された半面、前述のとおり攻めを重視した結果③の影が薄いように思います。意識して「堅く囲ってから鋭く攻める」ことができるよう、以前までの取り組みに加えて③を重視します。

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。

構成

■開発部

ライブラリ申請したやねうら王を強引に改造(既存の3駒での評価に加え、別に堅さ評価する等)★採用理由:最高の探索、フレームワーク。去年から使用している。

AWS接続兼時間攻めモジュール(ローカルにて使用)は自作

■評価関数

ライブラリ申請したAperyのものがベース ★採用理由:最高の評価関数

教師局面生成にて評価値を意図的に改変し追加学習する等で調整 <https://github.com/32hiko/HoneyWaffleSDT5>

■定跡

第5回将棋電王トーナメントで使用したものをベースに、より持久戦を目指す方針で調整

最後まで読んでいただき、ありがとうございます。

局面評価の極北を目指しています。

2018年の相違点ですが、強化学習に使用する自己対戦棋譜の初期状態生成のため、以前からやっていた自己対戦の勝率に基づいた定跡生成手法を流用しています。定跡生成のため抽出したbookの拡張候補各局面から最低数十局程度の棋譜を生成し(並列処理による結果変動によって結果は異なる)、生成された5千万局程度の棋譜に基づいてelmo方式(勝率からのロジスティック回帰+クロスエントロピーによる数手先の探索結果の学習)による学習を行うと同時に、勝率に基づいて book 内の各手の採用確率を決定します。

定跡では2017年にえらい目にあつたので、Qhapacさんが公開していた定跡を取り込みました。定跡の末端での勝率ベースで更新していく手法なので、他の定跡でもその末端の勝率を求めることで取り込むことができるようになってます。

・ライブラリ選択理由

2012年ごろにKPP/KKP テーブルで機械学習手法を試そうとした時点では bonanza 以外に選択肢がなかったので、そのまま使い続けています。惰性です。

以下は2017年の内容です。

- ・対戦中に現れた局面を調べ、機械学習結果の欠陥を探して修正しています。
- ・三駒関係の各変数を分解して共通する要素を抽出したうえで機械学習することで未知の局面への対応能力を高めています。
- ・手番を考慮した評価値を使用しています。
- ・定跡では自己対戦結果から各局面の勝率の分布を推定して各手の採用確率を決めています。
- ・プログラムはbonanza 6.0にstockfish の手法を取り入れています。

第 28 回 世界コンピュータ将棋選手権
大將軍（たいしょうぐん） アピール文書

開発者 横内 健一 横内 靖尚

○大將軍の概要

評価関数に主眼をおいた将棋ソフトです。

過去には 4 駒の位置関係（KPPP:N4(2013)/KKPP:N4S(2013-2015)）を評価関数に使用していました。現在は、学習作業の効率を考慮し現在は 3 駒関係の評価関数を用いています。

○大將軍の特徴

評価関数の作成に関しては、以下の点を工夫しています。

（評価関数は、以下の 2 ステップの手順にて学習しています）

1. プロの棋譜からの学習

いわゆる **Bonanza** メソッドをベースに、手番の評価やミニバッチの手法を取り入れています。

ミニバッチを用いることで、学習が安定し、短時間で学習の成果を確認することができます。

手番に関する評価は、3 駒に手番加えた **KKPT** 型を採用しています。**KPPT** 型よりも計算コストが小さいため、10%程度探索速度が向上します。

2. 自己の探索結果からの強化学習

プロの棋譜からの学習において勝率が飽和したところで、浅い探索結果と深い探索結果の評価値を用いて、評価値の不整合を修正していきます。

学習させる局面により勝率に影響するようですが、どのように設定するとよいかは今後の課題です。

○ライブラリの使用と実装方針

やねうら王ライブラリを使用します。評価関数の開発に注力するために、探索部分はライブラリを使用して他のソフトと同等レベルをキープしたいと考えました。他のライブラリと比較して、ソースコードが理解しやすいというのも選定理由となりました。

今日では、プロの棋譜を使わなくても評価関数を作成することが可能ですが、従来手法の学習結果（ウナギ屋のタレ）とミックスして活用していきます。

elmo ライブラリは現在使用していませんが、評価関数が優れており、選手権までに定跡部や追加学習等で使用するかもしれませんので、使用申請をしておきます。

妖怪惑星Qhapaqのアピール文

Ryoto Sawada, Yuki Ito and Toshihiro Shirakawa

ITに強い将棋部 (updated 2018/04/08)

対局結果からの学習
ビッグデータ作成
ディープラーニング
敵対学習...



定跡の整備
時間制御
ソフトを使った研究
詰まらずに勝つ...

Who is Qhapaq ?



カパック(ケチュア語で「偉大なもの」を指す形容詞)と読みます。本作が数々の巨人の肩に立った作品であることを示しています。

数理解析を駆使したご家庭レベルのPCで行える強化学習や、高速化を売りにしています

主な実績:

- ・第27回世界コンピュータ将棋選手権10位
- ・第五回将棋電王トーナメント:5位入賞
- ・現在最強の公開関数Apery-Qhapaq関数
- ・電王トーナメント累計23戦中18回後手

チーム名:ITに強い将棋部

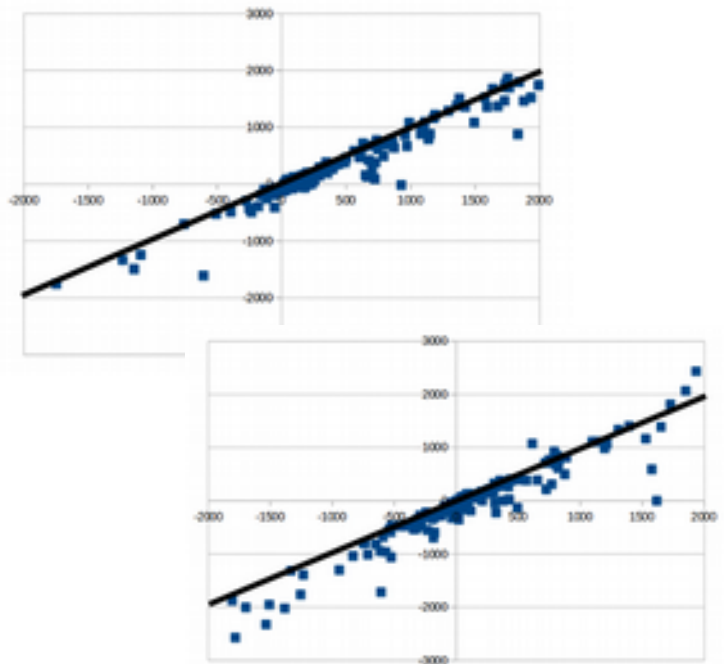
Ryoto Sawada : 某企業 and 大学の研究者。専門は量子物理

Yuki Ito : Ubuntu帽を被った野生の物理学徒。高速化のプロ

Toshihiro Shirakawa : パズル世界の怪人。趣味は電子ゴルフ

将棋フィーバー(ただしAI)

AIを使って将棋の可能性を広げたい!



将棋界:

- ✓ 藤井フィーバー
- ✓ 羽生永世七冠誕生
- ✓ ひふみん超おもしろい

AI将棋界:

- ✓ Ponanza引退
- ✓ オープンソース勢の大躍進
- ✓ AIブームに便乗した手法の進歩

評価値推移から見る棋風解析
藤井 vs 羽生は藤井勝率65%
ぐらいらしい(Qhapaq曰く)

1年前のチャンピオンとのレート差が
300-400。角落ちのレート差が600ぐらい
と言われているので、2年で角落ち
を埋める程度の成長率

Qhapaqの挑戦: まず強くする

【図は102手目△8二番まで】

	9	8	7	6	5	4	3	2	1	
△	桂							桂	成	一
	皇						王			二
	飛		馬			王	争			三
争		争			争	争				四
										五
歩	争	歩	歩	歩						六
		銀	金		歩	質				七
	玉	金								八
香	桂					皇				九

七歩入山九△

金銀桂歩二

今の流行は強化学習

- ✓ 人間やソフトの棋譜を使った学習
- ✓ 勝った局面は良い、負けた局面は悪い
- ✓ ソフトの評価値と勝敗を合議(elmo絞リ)
- ✓ やねうら王などのライブラリで個人でも再現可能

今後の課題:

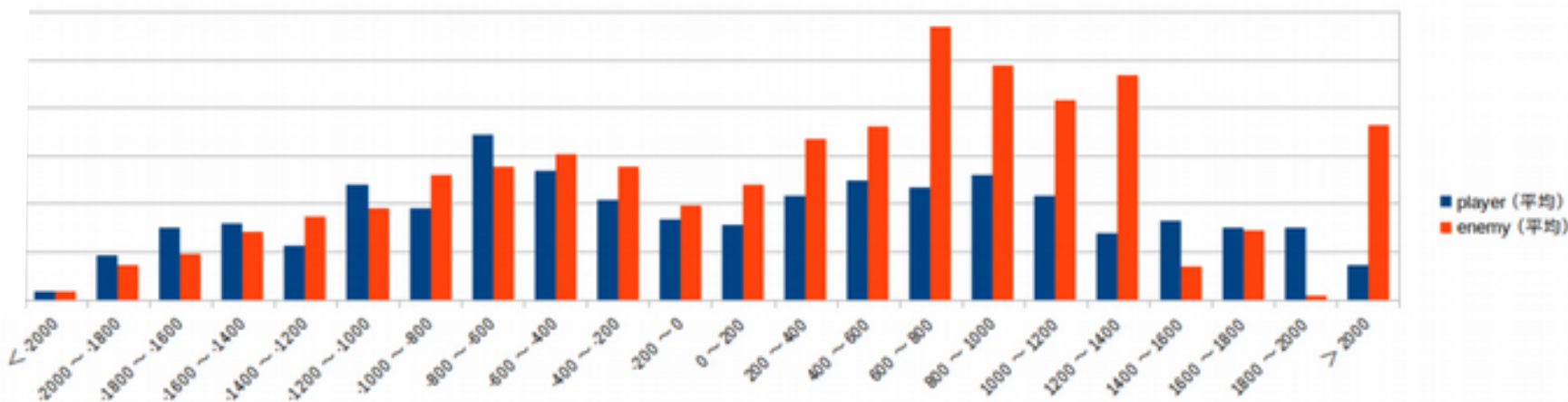
- ✓ 普通に棋譜を作ると滅茶苦茶計算資源がかかる
- ✓ 教師の数、読みの深さ双方が必要
- ✓ 人間ならすぐ駄目と解ることもコストが高い
- ✓ 評価関数は線型のままで良いのか

Qhapaqの工夫:

- ✓ floodgateなどで得られる深い読みの局面を利用(教師精度はNo1)
- ✓ 教師データが少ないのでその局面から浅い読みで作られた局面も教師にする
- ✓ その教師の勝敗や評価値は他データから類推する
- ✓ 今で言うVirtual Adversarial Trainingに近い
- ✓ KPPTを強くする以外にも新規評価関数開拓をやってる
- ✓ 高速化もやってる(チームメンバーのItoさんが)

Qhapaqの挑戦2: 将棋を面白くする

棋力解析ソフトELQ



上の図は藤井六段と対局相手のソフト視点での悪手率。
横軸は評価値（左が不利な局面、右が有利な局面）

- ✓ 藤井六段、対局相手の双方とも、-600～600ぐらいのイーブンな局面が一番ミスが出やすい
- ✓ 藤井六段は有利な局面でのミスが少ない。詰将棋万歳

このデータを用いて対局者の強さを予想することも可能。藤井六段は今後羽生永世七冠を越えて行くとQhapaqは予想していますが果たして...

利用ライブラリとその選定理由

- ・ Apery

評価関数の初期値に使っています。また、学習部にはAperyの学習部を改造したものを用いています

- ・ やねうら王

探索部、学習部として使っています

- ・ 技巧

進行度などの取り扱いや、ssh通信部は技巧のコードを参考にしています。ゼロから書き直しているとはいえ、コピー部分も多いのでライブラリ申請しておきます

最後にひとこと

Qhapaq強いですよ！

将棋ブームをエンジョイしてる
その人、コンピュータ将棋も
見てくれると嬉しいです！

再放送

妖怪惑星Qhapaqのアピール文

Ryoto Sawada, Yuki Ito and Toshihiro Shirakawa

AIが導く
明るい未来

ITに強い将棋部

ディープ
ラーニング

ハイテク株で
X万円を溶かした
Qhapaq

KPPT強くするぞ

GPU

K

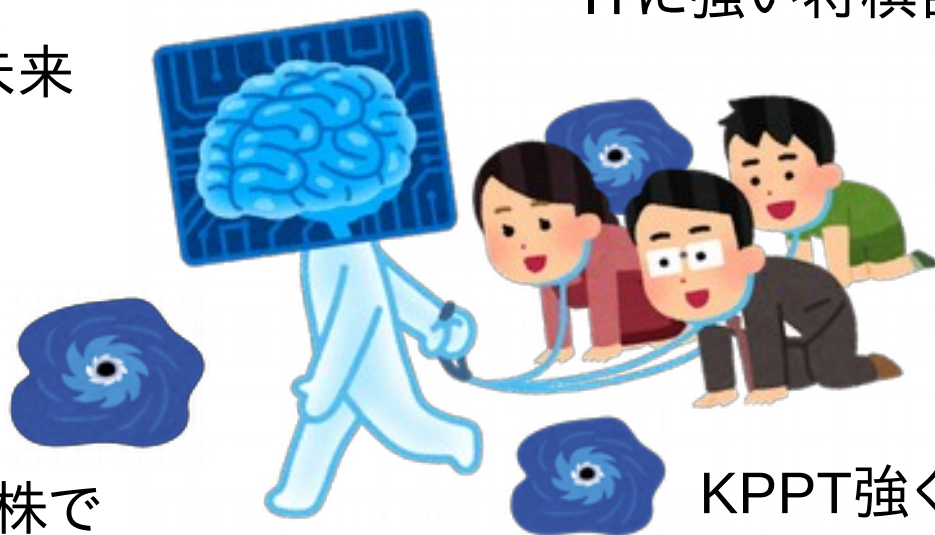
P

棋譜

AWS

Ryzen

小市民



Who is Qhapaq ?



カパック(ケチュア語で「偉大なもの」を指す形容詞)と読みます。本作が数々の巨人の肩に立った作品であることを示していますが後述するやらかしは巨人たちの責任ではありません。

主なやらかし:

- ・被り物を被ってニコニコ動画出演
- ・振り駒の結果に中指を立てる
- ・対戦カードを正しく組んでももらえない
- ・通信器具を会場に持ち込み忘れる
- ・電王トーナメントでのサイリウム配布テロ
- ・なんとかchに経歴を晒される

チーム名: ITに強い将棋部

Ryoto Sawada : wikipediaに名前がある。1日の食費は300円ちょい

Yuki Ito : 異議あり民の一人。ラーメン屋で外人に絡まれて困惑してた

Toshihiro Shirakawa : 物書き時々YouTuber。服装がファンタスティック

将棋フィーバー(ただし身内)

AIを使って開発者のキャリアを広げたい!

カバック
カバック・ニャン
カバック・ニャン アンデスの道
カバックニャン
カバック インカ
カバックス
カバック・ナン アンデスの道
カバックニャン 世界遺産
カバック・ニャン アンデスの道 wikipedia

- ✓ bing+カタカナだと出てこない
- ✓ googleだと将棋をサジェストされる
- ✓ Qhapaqで検索すると出てくる
- ✓ カタカナ検索の結果は諸事情で出せない

将棋界:

- ✓ 三浦九段の冤罪事件
- ✓ 加藤 One hundred twenty three
- ✓ 某電王による大量献金

妖怪惑星:

- ✓ アカデミアから(強制)引退
- ✓ AIブームに便乗(ただしAI株で失敗)
- ✓ 履歴書に書けるネタを稼いでキャリアアップ°

最近AIができる物理屋ではなく、物理ができるAI屋とされているフシが有る。

Qhapaqの挑戦：人の失敗を吊るし上げる

elmo絞りの此処が駄目：



- ✓ 短い持ち時間だと強くならないことが多い
- ✓ 一度しか出てないKPPTの値がelmo絞りだと大きくなりすぎる問題
- ✓ 教師局面110億とか狂気の沙汰
- ✓ ニューラルネット系の学習だと超強力

評価関数合成の此処が駄目：

- ✓ 合議制では強くなるような関数でも強くならない(ことがある)
- ✓ 強くなる関数では合議より強くなる
- ✓ どの成分の変化がレート変更に結びつくかを分析する必要あり

Qhapaqの此処が駄目：

- ✓ こうご期待の一言でアピール文を片付けられず、大会前に役立つ情報を漏らしてしまう

Qhapaqの挑戦2

ソフト開発者 = 秀才の構図を破壊する

■参加プログラム情報

振り飛車!飛車!飛車!飛車ううううわあああああああああ
ああああああああああん!!! ああああああ…ああ…あっ
あっ!あああああああ!!!飛車飛車飛車ううううわあああああ
あ!!! ああクンカクンカ!クンカクンカ!スーハー!スーハー!
スーハー!いい匂いだなあ…くんくん んはあ!!48に居る飛
車たんの裏面の桃色プロンドの文字をクンカクンカしたいお!ク
ンカクンカ!ああ!! 間違えた!モフモフしたいお!モフモフ!モフ
モフ!龍の文字モフモフ!カリカリモフモフ…きゅんきゅんきゅい!!
りゅうおうのおしごと7巻絶賛発売中だよ!!あああああ…ああ
あ…あっあああああ!!ふあああああんっ!! sdt未だ終わって
なくて良かったね飛車たん!あああああああ!かわいい!飛車た
ん!かわいい!あっあああああ! アニメも評価されて嬉し…い
やああああああ!!!にやああああああああん!!ぎやあああああ
ああ!! ぐあああああああああ!!!将棋なんて現実じゃな
い!!!!あ…小説もアニメもよく考えたら… 振り飛車が勝ち越す
未来は現実じゃない?にやああああああああああん!!うあ
あああああああああ!! そんなあああああああ!!いやああああ
あああああああ!!はあああああああ!!48飛車あああああ!! こ
の!ちきしょー!やめてやる!!振り飛車なんかやめ…て…え!?
見…てる?角交換四間飛車を指しこなす本の飛車ちゃんが僕を
見てる? 四筋の飛車ちゃんが僕を見てるぞ!飛車ちゃんが僕を
見てるぞ!駒台の角ちゃんも僕を見てるぞ!! アニメのあいちゃん
が僕に話しかけてるぞ!!!よかった…世の中まだまだ捨てたモン
じゃないんだね!! いやっほおおおおおお!!!僕には将棋ちゃ
んがいる!!やったよケティ!!ひとりでするもん!!! あ、コミックの
ひなたちゃああああああああああああん!!いやあああああ
あああああああ!!!! あっあんあっあんあアン様ああ!!
せ、セイバー!!シャナあああああああ!!!ヴィルヘルミナああ
あ!! ううううう!!俺の想いよ将棋盤へ届け!!readyok! と
言った感じでITに強い将棋部は常に最強を求め将棋ソフト開発
に余念がない。しかし、開発の疲れからか不幸にも疲れからか、
不幸にも黒塗りの高級車に追突してしまう。後輩をかばいすべ
ての責任を負った三浦に対し、車の主、暴力団員谷岡に言い渡
された示談の条件とは…。

チ
ム
名

WCSCのサイトが前時代的なcgiであり
チーム名の文字数制限が抜けていたので
チーム名をルイズコピペにしてみた
(360RTぐらいされた)



たぬき開発者と共謀して電王トーナメントの
ライブに自腹でサイリウムを持ち込んだ

肩を借りた巨人とその選定理由

- ・ 既存定跡を組み込んだ量子アニーリング
定跡作りの際に使っています。複数の既存定跡にノイズ
(具体的には一定確率で定跡を無視する)を加えたものと
自己対局を行い、勝った棋譜を収集することを繰り返す
ことで、未知の定跡に対してロバストな定跡を作っています
- ・ 妖怪惑星クラリス
今回のソフト名の元ネタです。このゲームが末永く
発展することを願って付けました
- ・ Lostorage conflated WIXOSS
sdt5でのソフト名の元ネタです。四期の放送(当時未確定)を
願って付けました。願いが叶ったわけですが、これはもしや...

最後にひとこと

妖怪惑星クラリスが唐突に
サービス終了しました

ウキッアアアアアア

nozomi アピール文書 2018

2018/3/31 Yuhei Ohmori

はじめに

- コンピューター将棋業界にもDeepLearningとMCTSのビッグウェーブがやってきました
- 「乗るしかないこのビッグウェーブに！」

評価関数

- 今まで通り、KPP + KKPTの3駒関係で、自己対局の勝敗と評価値を教師として学習しています
- なるべく深いdepthで自己対局を行うようにしています
- 時間とお金を節約するため、少ない棋譜からでも学習できるようにしました
 - 一回のパラメーターの更新に2000万局面使用しています
 - 損失関数もelmo形式ではなくちょっと変更しています
- ちょっとだけ評価値の計算を改善しました

探索

- 今まで通り、Stockfishベースの探索です
- 指し手の選択確率を計算し、オーダリングに使用しています
 - 単純に指し手と駒の位置関係でロジスティック回帰しました
 - 評価関数の学習とほぼ一緒なのでお手軽
 - nozomiの指し手を一手も読まずに35%の確率で予言可能
 - 残念すぎますが、まあこんなもんかなあ
- 指し手の選択確率をLMRに対しても使用する予定です

その他

- チェスソフトのStockfishをベースに作成しています
 - <https://github.com/official-stockfish/Stockfish>
- 一部Aperyを参考にさせていただいています
 - <https://github.com/HiraokaTakuya/apery>
- いつものことながらこれらのソフトがなければnozomiはここにいなかったなので、大変感謝しております
- C#はAWSとのSSH通信で使用しています

おわりに

- 誠に遺憾ではありますが、ビッグウェーブに乗れなかったものと認識しております
- それでは今年もよろしくお願いいたします

第28回世界コンピュータ将棋選手権
Aperyアピール文書
2018年3月31日 平岡 拓也、杉田 歩

- Aperyとは

- 読み方は「えいぷりー」です。
- GPL v3ライセンスでオープンソースで開発しています。
- ソースコードや実行ファイルは以下で公開しています。
- <http://hiraokatakuya.github.io/apery/>
- CSAライブラリとして登録しているので、誰でもAperyを改造してご自身のソフトとして出場出来ます。

- 作者紹介

- 平岡 拓也 (ひらおか たくや)
趣味で作っています。
Mail : hiraoka64@gmail.com
Twitter: @HiraokaTakuya
- 杉田 歩 (すぎた あゆむ)

- 実績

- 第22回世界コンピュータ将棋選手権 22位 (2次予選敗退)
- 第23回世界コンピュータ将棋選手権 9位 (2次予選敗退)
決勝進出に一步足りず。
新人賞に一步足りず。
- 第1回将棋電王トーナメント 6位 (5位決定戦敗退)
第3回電王戦出場に一步足りず。
- 第24回世界コンピュータ将棋選手権 優勝
- 第2回将棋電王トーナメント 5位
- 将棋電王戦FINAL 斎藤慎太郎五段(段位は当時)に完敗
- 第25回世界コンピュータ将棋選手権 4位
- 第3回将棋電王トーナメント 3位
出場ソフト名は「大樹の枝」。
名前はヤフオク!の「みんなのチャリティー」で命名権をオークション(全額チャリティー)で販売して落札者の方に決めて頂きました。
- 第26回世界コンピュータ将棋選手権 4位
- 第4回将棋電王トーナメント 2位
出場ソフト名は「浮かむ瀬」。
名前はヤフオク!の「みんなのチャリティー」で命名権をオークション(全額チャリティー)で販売して落札者の方に決めて頂きました。
- 第26回世界コンピュータ将棋選手権 12位
- 第5回将棋電王トーナメント ベスト12

- 技術的特徴

- 全体的にチェスソフトのStockfishの設計を取り入れており、評価関数は3駒関係です。利きのデータなどは持っておらず、シンプルな構成になっています。
- 評価関数で手番も評価しています。
- 評価関数の教師データ生成を皆さんにお願いするシステムを今年も使っています。
今回は Linux 限定になっています。
ご協力ありがとうございます！
<https://github.com/HiraokaTakuya/apery-machine-learning>

- 昨年との違い

- 昨年の選手権で elmo が採用していた、評価関数の学習データ生成時の対局で、勝敗を記録し、学習にその局面と後の勝敗を考慮する方式を取り入れました。
- 学習部を書き直した過程でバグが取れたのか、WCSC27 の elmo の評価関数より精度が上がり、半年前の第5回将棋電王トーナメント時には、オープンソースの中では最も評価関数の性能が高かったようです。
- 半年前からの変更点は、学習データ生成時の探索深さを8から10に変更した点です。
とても計算量が多いので、半年前は出来ませんでした。やることにしました。
これは DeepMind の AlphaZero が AlphaGo と同様の方法で将棋でも既存のソフトより強くなったとい

う発表を受けて、

単に学習に使った計算量が桁違いに大きいからではないかと考えたからです。

既存手法でももう少し計算量を増やして学習すれば、更に強くなるのではないかと考えました。

具体的には、探索深さを 10 で学習し、その後 12 でもう一度学習する計画でした。

学習データ生成を誰でも出来るシステムなども作りましたが、時間的に探索深さ 12 は実験出来そうにないです。

まだ実験中で、探索深さ 10 でのデータ量が 8 の時よりも少し少なかったりで、結論を出すには早いですが、

探索深さ 10 の時点で伸びがほとんど無いかも知れません。

本番までにもう少し実験しようかと思います。

第28回
世界コンピュータ将棋選手権
なのはアピール文書

2018年4月1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
- 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！、とある科学の超電磁砲、ラブライブ！
- 将棋ウォーズ 1級
- 囲碁は日本棋院 初段(アマチュア)
- スマホゲーム非課金勢...スクフェス、デレステ、Fate/GO、ぷにぷに等
- ！職業プログラマ∩！研究者 ∩！東大
∩！学生



■ 開発環境

➡ こんなPCで開発していますw

➡ 出場PC(予定)

CPU : AMD R7 1700

メモリ : 32GB

OS : Windows10



■ なののはの構成

- Visual Studio Community 2017(C++)にて開発
- 手生成では歩、角、飛の不成も生成
- 探索はStockfish使用
 - 評価ベクトル縮小、詰めルーチン(df-pn)削除したものを、**なののは**miniとして公開
- Bitboard未使用(盤情報は配列)
 - AMD RYZENに適したデータ構造
- 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
- 評価ベクトルはライブラリ使用予定
- 詰めルーチン(df-pn)搭載

...と、どこにでもあるような平凡な構成

■ なのは何の特徴は？

- ➡ 強力な詰めルーチン搭載(なのは何詰めとして公開)
- ➡ なのは何詰めは江戸時代の名作 611手詰めの「寿」を解く
- ➡ 常に詰みを狙い、一発逆転するポテンシャルを秘めています！



■なのはの新たな特徴(予定)は？

- ➡ 詰めルーチンの更なる強化
- ➡ 終盤の強化
 - ➡ コア数が増加したため、詰め探索用のスレッドを設定

ユ「詰め探索変更ってどんな風に変えたの？」

な「うん...あれって探索が遅くて高速戦では使えないから」

ユ「やっぱり高速化？」

な「ううん 逆！チャージタイムを増やして威力を大幅アップ！」

な「最大詰め手数強化を最優先してみたの」

ユ「そ... そう.....」

※以下を参考に改変:「魔法少女リリカルなのはA's THE COMICS」 pp.20-21, 原作 都築真紀, 作画 長谷川光司, 学習研究社, 2006.

■ なののはの強さ

- ➡ 第24回世界コンピュータ将棋選手権で**1位**！(*1) (*2)
- ➡ 第25回世界コンピュータ将棋選手権で**1位**！(*1) (*3)
- ➡ 第26回世界コンピュータ将棋選手権で**2位**！(*1) (*4)
- ➡ 第27回世界コンピュータ将棋選手権で**1位**！(*1) (*5)

(*1) AMD製CPUをメインに使ったシステム構成での参加ソフトの中で(当者調べ)

(*2) トータルでは17位 (*3) トータルでは11位

(*4) トータルでは12位 (*5) トータルでは13位

■ 意気込み

- ➡ できれば**シード権獲得**！
- ➡ あわよくば**入賞**！(あと、AMD勢で1位奪取したい)
- ➡ **全力全開、手加減なしで！！**

(今後の課題)

- ➡ 検討に使われるように振り飛車が不利飛車にならないようにする
- ➡ 詰めろ絡みの局面が続いても正着を続ける終盤力(即詰みがない時)
- ➡ 最長詰め手数の詰将棋の「ミクロコスモス」(1525手詰め)を解く
- ➡ カットインやエフェクトなど**派手な演出**！！



■ 使用ライブラリについて

- なのはmini 自作のため、いろいろなところで使っています。
- やねうら王 学習部や評価ベクトルを使用(予定)。公開からこなれて信頼性が高いため。
- Bonanza ちょっとしたテストをするときなど、慣れもあって使いやすいため。

■ 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

昨年からの主な変更点は以下の通りです。

- (1) 自己対戦棋譜からの局面評価関数と着手確率の学習
- (2) 局面評価関数の拡張
- (3) 着手確率の特徴の拡張、及び多クラス分類へのモデルの変更
- (4) モンテカルロ木探索を用いた定跡生成

他にも何かしらやる予定です。

2018 年第 28 回世界コンピュータ将棋選手権アピール文書

GPS 将棋は、東京大学大学院総合文化研究科の教員・学生が開催しているゲームプログラミングセミナー (Game Programming Seminar = GPS) のメンバーが中心となって開発が行われているソフトウェアです。フリーソフトウェアとしてソースコードやデータを公開しています。

2003 年から GPS 将棋として世界コンピュータ将棋選手権に参加し、2009 年の第 19 回及び、2012 年の第 22 回世界コンピュータ将棋選手権で優勝した他、2010 年の第 20 回及び、2013 年の第 23 回世界コンピュータ将棋選手権では 3 位の成績を得ています。また、清水女流王将（当時）とコンピュータ将棋（あから 2010）との対局では、激指、Bonanza、YSS とともに GPS 将棋も参加しました。2013 年に行われた第 2 回電王戦において三浦弘行八段（当時）と対局しました。

2018 年の第 28 回世界コンピュータ将棋選手権は GPS 将棋として 16 回目の参加となります。

技術的な特徴としては、コンピュータチェスやコンピュータ将棋の最新の研究を取り入れていることが挙げられます。例えば、利きを管理する高速な将棋盤、実現確率を用いた探索、評価関数の自動学習などがあります。また評価関数は現在、序盤、中盤 1、中盤 2、終盤の 4 種類を用いています。他にも、疎結合並列探索や df-pn（並列協調）を用いた詰探索にも対応しています。技術的な詳細は参考文献をご覧ください。

Team GPS

参考文献・ウェブサイト

- WWW サイト: <http://gps.tanaka.ecc.u-tokyo.ac.jp/gpsshogi>
- 多数の計算機を活用したゲーム木探索技術の進歩 - 三浦弘行八段と GPS 将棋との対局を振り返って-, 金子知適, 田中哲朗, 情報処理 54(9), 914—922, 2013. :
<http://id.nii.ac.jp/1001/00094757/>
- コンピュータ将棋の新しい波: 3. 最近のコンピュータ将棋の技術背景と GPS 将棋金子知適, 情報処理 50(9), 878—886, 2009.:
<http://id.nii.ac.jp/1001/00067191/>
- S. Yokoyama, T. Kaneko, and T. Tetsuro: Parameter-Free Tree Style Pipeline in Asynchronous Parallel Game-Tree Search, The 14th International Conference on Advances in Computers and Games (ACG2015) :
<http://www.graco.c.u-tokyo.ac.jp/~kaneko/papers/acg2015-yokoyama.pdf>
- Twitter: <http://twitter.com/gpsshogi>
- Floodgate: <http://wdoor.c.u-tokyo.ac.jp/shogi/>

第28回世界コンピュータ将棋選手権

スーパーうさびょん2 アピール文書

=====

2018.3.31 うさびょんの育ての親

●目標

- ・AMD製CPU1位の座を取り戻す！

●使用ライブラリ

- ・なのはmini
- ・Apery

●プログラム全体

スクラッチから「うさびょん3」として開発していたものが行き詰って（というか仕事と並列だと十分な時間が確保できなかった…）去年の「うさびょん2' TURBO」に追加開発を行うこととしました。その為、名前は見る人が見れば去年からの連番になっています（苦笑）。

その為、コードのベースには以前と変わらず「なのはmini」を利用しています。

探索部はStockFishを参考にしつつ、自己対局で効果がマイナスだったり、効果が認められなかった枝刈をいくつか外したり、自前で考えてみた枝刈を追加してみたりしています。（自前の枝刈は効果がプラスになったものがまだありませんが…。）

●定跡

今回は「なのはmini」の定跡部は使いません。新たに定跡部だけコードを含めて書き起こしています。なお、定跡は0ベースで作成中です。定跡に少し力点を置いて開発しているのですが、公開されている評価関数の中では「Apery」の最新のものが

強い&準備中の定跡との相性が良いようです。

その為、今回の大会では「Apery」の評価関数を用います。

定跡の話は細かく書くと他の開発者に対策を練られそうなので思いっきり割愛させていただきます…が、驚かれるような事はやっていません、とだけ書いておきます。

「習甦」

特徴：多層構造を持つ評価関数

- ・駒の価値：盤面全体の駒の利きと持駒および手番から算出する玉の安全度に対応した非線形関数
- ・駒の働き：入玉していない玉と玉以外の2つの駒の位置関係

→入玉して玉の安全度が高くなると、持駒と敵陣にいる駒が小駒：大駒＝1：5の駒割に近づく。

評価パラメータの機械学習方法

- ・前世代の評価関数を用いてフィッシャーランダムチェスに準じたユニークな初期局面2494800からの自己対戦棋譜を数セット作成する。
- ・自己対戦棋譜における評価値の推移をフィードバックして推定された勝率を割引報酬として強化学習する。

第 28 回世界コンピュータ将棋選手権アピール文書

概要

「たこっと」は、電王戦を見てコンピュータ将棋に興味を持った筆者らが、フルスクラッチで実装した(している)将棋プログラムです。Web 上の解説記事や論文, Stockfish, Apery, やねうら王, Bonanza 6.0 のソースコードを参考にしています。

特徴

- 各種処理が AVX2 命令で実装されていて高速
- AVX2 命令を適用しやすいデータ構造 (非ビットボード)
- Stockfish 風の探索アルゴリズム
- オンライン学習ルーチン

以前のたこっとから引き継いでいる特徴については過去のアピール文書を参照してください。

- [第 26 回世界コンピュータ将棋選手権アピール文書](#)
- [第 27 回世界コンピュータ将棋選手権アピール文書](#)
- [第 4 回将棋電王トーナメント PR 文書](#)
- [第 5 回将棋電王トーナメント PR 文書](#)

ライブラリの使用について

以下のライブラリの最新バージョンを使用申請しています。

- **Apery**
教師データの生成や強化学習時の初期値として評価関数バイナリのみを使用する予定
強いということが選定理由
- やねうら王 コンピューター将棋フレームワーク
現時点では使用するかどうか未定
必要なモジュールがそろっており、ソースコードを読み込んでいて慣れ親しんでいるのが選定理由
- **elmo**
現時点では使用するかどうか未定
Ponanza を倒したという実績が選定理由

ライブラリを提供していただき、Apery の平岡様、やねうら王の磯崎様、elmo の瀧澤様には感謝いたします。

学習ルーチン

以下に挙げるような近代的な機械学習の機能を一通り実装してあります。

- 確率的勾配降下法
- オンライン学習
- ミニバッチ
- 複数の損失関数
- 複数の最適化手法
- 正則化

第 5 回将棋電王トーナメントで elmo 方式の学習アルゴリズムを実装し、コンピュータ将棋界のトレンドにも追従しています。

評価関数に KPPT 型の三駒関係のモデルを採用していますが、近年は強化学習しても棋力の向上が微々

たるものになってきました。大量の学習データの作成と学習を何度も繰り返す必要がありますが、計算資源の確保に苦慮している貧乏プログラマーには厳しい限りです。そこで学習で棋力向上が望めるような新たなモデルの開発に取り組むことにしました。

が．．．

今のところ結果がついてきません。

あまりに強くない場合はライブラリの評価関数バイナリを強化学習し、大会に参加するかもしれません。

第28回世界コンピュータ将棋選手権

PAL

アピール文書

山口 祐

PALの概要

- pal /パ°ル/【名】<話> 仲間、友だち、仲よし
(例) a pen ー
- ショーギはともだち こわくないよ!

開発者

- 山口 祐 [@ymgaq](https://github.com/ymgaq/AQ)
- 囲碁もやってます
 <https://github.com/ymgaq/AQ>

PALの特徴

- Deep Learningを使わない
- 0ベースの評価関数の学習
- 強化学習と **敵対的学習** のハイブリッド
- クラウドサーバにおける高回転並列学習 (N連ガチャ)
- 並列探索の最適化

使用ライブラリ

- やねうら王 v4.80 (改変しやすそうだったので)

チーム Barrel house のアピール文書@第 28 回世界コンピュータ将棋選手権

Barrel house とは岡山の駅前にあるビアバーです。職場や社会人勉強会などで仲間を求めさすらって行きついたところで一人目の仲間を見つけたのと、マスターに許可頂いたので名前をお借りしました。その後、メンバーが増えて当初の方向性とは全く違って来ましたが、まぁ一度出した名前を変更するのもアレなのでそのまま行きます。

プログラム名：Hefeweizen

ドイツ南部の酵母入りビール。濁った白ビールってのが日本で通る表現かと。命名経緯は上記の通りです。

チームの特徴.

メンバーが全員初参加で冷たい白ビールのようなフレッシュなチームです。

メンバー間の擦り合わせも適当ですが昨年秋の電王トーナメント経験者が2名おり、多少は実績がありますので乞うご期待という感じです。電子工作が得意なメンバーや家電販売が得意なメンバーもおります。結成が遅れ現在手分けしている段階ですので確定事項は少ないのですが、やねうら王の探索部とオリジナルの局面評価関数と分岐先読みクラスタを使用する予定です。深層学習で作った評価関数や魔改造した技巧2などがあるのですが、本番で使われるかどうかはある程度ツールが揃って強さを計測してから決定する予定です。疎結合なクラスタ構成なのでパーツ交換で対戦ごとに構成を変える可能性もあります。

CSA 使用可能ライブラリ使用表明.

メンバーの意思統一が図れておりませんので多めに入れておきます。また、試行錯誤ツールで色々使わせて頂いております。

Apery, やねうら王, tanuki-, Qhapaq, elmo, 技巧, python-shogi, 人造棋士 18 号
(作者本人は入れなくていい?)

使用マシン.

普通のノートパソコンに加えて、クラウドの力をお借りする予定です。ベンチマーク等も未だなので具体的なことは全く決まっていません。

4 月 13 日追記

ライブラリの選定理由を加筆修正しろとのことですので、追加します。

ライブラリ選定理由

python-shogi：python を用いた棋譜および局面の管理，sfen 文字列の展開など

Apery：評価関数作成のための教師データ作成に利用。

やねうら王：探索部が高速なため主に探索部の利用。定跡部作成時にも利用。

tanuki-：互換性があり強いため評価関数の作成及び仮想敵として勝率計算に利用。

Qhapaq：互換性があり強いため評価関数の作成及び仮想敵として勝率計算に利用。

Elmo：互換性があり強いため評価関数の作成及び仮想敵として勝率計算に利用。

技巧：評価関数の作成及び仮想敵として勝率計算に利用。探索部のオーダリングにも利用。

人造棋士 18 号：独自のルーチンを用いた評価関数の作成に利用。

クラスタ形式でのテストをまだ行っていないため本戦での運用で若干変更があるかもしれません。

Novice PR文書

Noviceとは

2015年より開発を行う

第三回電王トーナメントから5大会連続最年少出場
(当社調べ)

前回からの変更点

過去のコードを全て捨てました

0からのスタート

Stockfish 9に沿って作成

横型のRotated Bitboardを採用

実装

(結局、高速1手詰入れました)

王手周りは0.5手延長や枝刈りを甘くしている

PolyglotBook というチェスと同様の定跡形式を採用

StockFishの探索から将棋に向いてなさそうなものを除去

指し手のオーダリングも将棋向けに調整

ライブラリ

AperyのKPP.bin/KKP.binをお借りします

自作のConverterを通して横型にしたものを使います

評価バイナリ以外は自作です

選定理由として最新の探索部と自己の探索部との比較が

行いやすいことが挙げられます

ハードウェア

とりあえず初日は友人から借りた i9 7900X を使います。

意気込み

そろそろ最年少じゃなくなりそうなので結果を出したいです

雑感

大学も折り返しが見えてきて

特になりたい職もなく

人生いろいろ悩みます

(ここに書くな)

名人コブラアピール文書

生粋のライブラリ勢

過去の経験から、ポンコツプログラマな自分がプログラムを改造すると大幅な低速化を招くことがわかりました。ですから、今回も探索部や評価部には手を付けないことにしました。低速化をしても直接的には読みのスピードに影響の無い、学習部だけに手を加えたいと思います。

使用ライブラリ

- Apery
 - 評価関数だけならば、トップを走っているといわれているので、学習部を改造して使用させていただきたいと思います。キメラの素として、評価関数も使用させていただきます。
- やねうら王
 - 現在探索部は最強だといわれているので、探索部を使用させていただきたいと思います。また学習部も改造して使用させていただきたいと思います。キメラの素として、評価関数も使用させていただきます。
- elmo、人造棋士 18 号
 - Apery-やねうら王系統の評価関数を使用しているため、キメラの素として、評価関数を利用させていただきます。

評価関数のキメラ化にこだわります

評価関数をブレンドすると強くなるらしいので、少なくとも 20 個の評価関数を作って、それを混ぜたいと思います。お手軽だし、「アンサンブル学習」という言葉やコンセプトがなぜか大好きなのです。

キメラ素材の調達方法

1. ライブラリの評価関数

ある程度実績のある評価関数を使用させていただきたいと思います。

2. ゼロから新たに学習させた評価関数

同じものを混ぜても、キメラ化の効果はありません。バリエーションが必要なので、次の方法でバリエーションを持たせたいと思います。

- 教師局面の復元抽出
- 特徴の部分的使用（ランダム部分空間）

教師局面と評価値

キメラ素材となる評価関数の学習には以下の教師局面と評価値を使用します。

1. Apery を使用して生成したもの
2. やねうら王作者の磯崎氏が配布しているもの

最後に

ライブラリ勢なのに今まであまりライブラリを使いこなせていなかった感じがします。今回はあまり余計な改造はせず、ライブラリを使いこなして、良い評価関数を作ることに重点を置きたいと思います。

第 28 回世界コンピュータ将棋選手権 「S.S.E.」アピール文書

2018 年 3 月 30 日

開発者：和田悠介、斉藤優輝、吉野拓真

S.S.E.の特徴

- フルスクラッチで開発

Stockfish 8 をベースに 0 から開発を行いました。盤面を表現するビットボードは縦型ビットボードを使用しています。合法手生成部はオリジナルのものを実装しました。この間まで、開き王手の生成にバグがあったので、他にもないか少し不安です。

- 探索

基本的には Stockfish 8 ベースになっています。定跡は使用しないつもりです。今後、Stockfish 9 の変更を一部取り入れ、探索パラメータの調整や枝刈り手法の変更を行う予定です。

- 評価関数

評価関数は、コンピュータ将棋選手権使用可能ライブラリである elmo の評価関数をそのまま使用する予定です。選定理由は、ライブラリの中で最も強い評価関数であるためです。ディープラーニングを利用した評価関数の開発も進めており、うまくいけば今後変更する可能性があります。

謝辞

縦型ビットボードの実装、千日手の判定を行う際に、一部 Apery の実装を参考にしました。Apery 開発者の平岡拓也さんに感謝を申し上げます。また、コンピュータ将棋選手権使用可能ライブラリである elmo の評価関数を使用しました。elmo 開発者の瀧澤誠さんに感謝を申し上げます。

WCSC28 アピール文

2018/3/26

ソフト名「ArgoCorse_IcSyo（あるごころせいっしょ）」

市村豊（いちむらゆたか）

Twitter：@argonworks

ブログ：<http://argon1.blog55.fc2.com/>

使用ライブラリは「やねうら王」です。選定理由はそもそもコンピュータ将棋を始めるにあたってやねうら王nano/miniのコードを読んで改造するところから始めてそのまま使い続けている感じです。また今回は評価関数のキメラ化というのをやってみたくなったので、混ぜる元としてelmo,tanuki-Apery,Qhapaqを選定させていただきました。また、1月のときになんか技巧もやねうら王系だと思いこんでいたため書いてしまいましたが違っていたので取り消します。すみません。

独自性として、SDT5のときの文と重複しますが、もともとコンピュータ囲碁から入ったことからモンテカルロ法の将棋への応用の文献をインターネットで読んでいて、それが「現時点から10手先の局面を評価関数で評価して元局面よりも評価値が高い場合を勝ちとする」という方法でモンテカルロ将棋をやっていたものがあつたので、メインの探索には使えないとしても定跡の選択手法としてはやって見る価値があるのではないかと考えて、定跡に候補が複数ある場合に二手先読みをしてその時の評価値が一番高い手の一手目を採用する、という変更を行いました。それでSDT5のときは「やねうら王」に付属していた定跡を使ったのですが、今回はfloodgateの2009-2017までの棋譜からとりあえず31手目までをそのまま定跡ファイルに変換したものを定跡として作ってみたのでそれでやりたいと思っています（棋譜を全部変換すると定跡のファイルサイズが大きくなって動くかどうか不安なので31手までにした、そのくらいだと650MBくらいですむので問題なく動作するかなと）。それで、「やねうら王」に付属の定跡では元よりも悪くなる手は入っていないとあつたのでそのまま使っていたのですが、floodgateの棋譜から単純に定跡に変換したものだとそのままだと元より悪くなる手が含まれているため、「元局面を評価関数で評価して、それよりも評価値が低くなる手は採用しない」というコードを追加しました。

ちなみに、評価関数のキメラ化を少し試してみて、現状では以下ようになりました。定跡にyaneura_book3を使って16手目まで確率で動いているので全く同じ手順にはなっていないと期待してい

ます。時間は0秒+一手あたりそれぞれ10秒ずつ加算で、名前の変え方が分からないのでsse41とsse42でコンパイルして「41」「42」の部分で判別していたのでその違いが出ているかもしれないです。ちなみに左側が41で右側が42でコンパイルしたものです。「&」が1：1でキメラ化したことを表しています。256手で勝敗がつかない場合を引き分けに設定しています。51回ずつ自己対戦させてみました。メインのPCのi5-2500Kを1コアでことごと動かしていました。左側から見て「勝ち-引き分け-負け」です。

110億生成 VS elmo(wcsc27) 39-1-11

110億生成 VS Apery(sdt5) 19-3-29

110億生成 & Apery(sdt5) VS Apery(sdt5) 22-3-26

tanuki-(sdt5) VS Apery(sdt5) 21-4-26

elmo(sdt5) VS Apery(sdt5) 16-2-33

tanuki-(sdt5) & Apery(sdt5) VS Apery(sdt5) 27-6-18

aperypaq VS Apery(sdt5) 23-8-20

aperypaq VS tanuki-(sdt5) & Apery(sdt5) 18-10-23

tanuki-(sdt5) & Apery(sdt5) VS aperypaq 16-10-25

aperypaq VS tanuki-(sdt5) & aperypaq 26-6-19

aperypaq VS (tanuki-(sdt5) & Apery(sdt5)) & aperypaq 25-9-17

とりあえず、「やねうら王」の最新版が公開されたのでそれにSDT5のときの旧版から移植をしたいと思っているのですが、適当にコードをコピペしてコンパイルしようとしたらエラーがたくさん出て、そのエラーの消し方がいまいちわからないので「C++は本当に難しいなー」と思って面倒くさくなって放置している現状です。なんとかやる気を出して最新版に移植したいとは思っています。評価関数にtanuki-(sdt5) & Apery(sdt5)を双方使ったの自己対戦結果が、

> やねうら王旧版（手作り定跡） VS やねうら王新版（yaneura_book3） 13-4-33

だったのが最新版じゃないと勝負権が無いような気がするのでなんとかやる気を出して大会までには移植したいと思っています（希望的観測）。

あとはAWSの高そうなのを借りないと勝負権が無いような気がするので、そうしたいとは思いますが、現状AWSの使い方がよくわかりません。AI竜星戦の時に会場でDeepEsperの中の人にレクチャーし

て頂いてアカウントは作りましたが、操作は「簡単ですよ」というほど言うほど簡単じゃないと思うのですが。あと、使い方はともかく、プランがたくさんあってどれを借りたらいいのかがよく分からないのですが。なんでみんな「サーバーを借りる」なんてマニアックな事をごく普通にできるのですかね。それは義務教育ではやってないと思うのですが。

それはともかく、まあ、テンションを上げてAWSが使えるようになって大会に臨めたら嬉しいなと希望しています。それなので「よくわからないけどAWSの高そうなのを使う」みたいな申請をしたいと思いますが、テンションが上がらなかったら「やねうら王」旧版で、パソコンはレノボのX220、OSはWindows10になりますので、そうになったらすみません。できるだけテンションを上げてやりたいとは今は思っています。

・SDT5参加とかの感想

2017/11にSDT5に参加出来ました。コンピュータ将棋の大会に参加してみたいと思っていたので参加できて嬉しかったです。その一ヶ月後の2017/12にコンピュータ囲碁のAI竜星戦に参加することが出来ました。参加できて良かったです。ただ、年末のコミケの原稿を書かないといけなかった時期でだいぶ大変でした。

両大会ともに参加していて「前から思っていたが、つくづくすごい人しかいない感じだなー」ということを改めて思いました。なんか、エース・オブ・エースがごろごろいる。私なんか混じっていて本当にすみませんと思いました。ただ、この場所にいると本当に面白いということも改めて思いました。

それにしても、念願かなってコンピュータ囲碁とコンピュータ将棋の大会に参加することが出来て考えさせられたのは、「そういえばこれって何を目的にしているゲームだったんだっけ?」ということ。私は「これ、参加して対局したい」と思っていて「対局すること」を目的にしていたし今もしているのだが、そういえば囲碁とか将棋って「勝つこと」が暫定の目的だったのだなということを周りの参加者を見ていると思い出してきた。

ゲームの目的が何であるかということを思うのだが、私はゲームの目的は「楽しむこと」だと思っている。だから「勝ち負けはあくまでも暫定的な仮の目的であり、本来の目的ではない」ということを思っている。だって、勝つことが目的だとしたら優勝者以外のすべての人が目的不達成になる。それはかわないけど、それで参加者が翌年から優勝者一名になったらもはや大会が成立しないんじゃないかとい

うことを思う。だから「負けたとしても楽しかった」と負けた人に思わせるようなゲームじゃないとゲームとして成立しないんじゃないかと思うことを思う。囲碁・将棋に限らず、スマホのゲームとかプレステのゲームみたいなのを新しく作るとしたら「プレイすることそれ自体に楽しさ・快感を感じさせる」ようなゲームを設計しないとダメなんじゃないかと思ってしまっている。2017/12月の囲碁のAI竜星戦で私はDeepEsperの中の人とチームを組んで出場したけど、DeepEsperの中の人が2017/3の囲碁のUEC杯で最下位になって心が折れたのでやめようと思う、という意味のことを言っていて私は本当にびっくりした。「どうしてこんなにすごいソフトを開発できるあなたがたかだか大会で最下位になった程度のことでやめようと思うのか」と。

SDT5のアピール文に書きましたが個人的に自動車のレースが好きなので、思い出するのが2006年にF1のレースにスーパーアグリF1が参戦した時のことです。「準備期間半年でのF1参戦は無謀」という状態で開幕戦バーレーンGPの予選でタイムを出した時に、テレビの解説者が「決勝レースのグリッドを獲得しました！」と興奮気味に言っていた気がする。後にも先にも、決勝レースの最後尾のグリッドを獲得することを「快挙」として言っていたのはこの時しか聞いたことがない（翌日の決勝レースで彼らはトップから4周遅れの最下位でチェッカーを受けて、「F1の決勝で完走したのが快挙」と日本のメディアは概ね肯定的だったのもつくづく印象的だった）。多分、私の勝負っていうのはこういうような勝負をしているんだろうな—ということを今でも時々思う。SDT5の会場で「ここにいる人達は基本的に頭のいい人しかいない」というコメントが流れてきて、「そうなんだろうな—」と思った。だから、大会に参戦する権利を獲得することが一番の難題なんだと私は思っていた。インターネットのスラングなのかもしれないですが「0回戦負け」というやつです。基本的に「0回戦突破」が一番の難題であって、参戦権さえ得られたのならばあとはもう最下位でもそれは「快挙」だと私は思っている。

それにしてもSDT5のときに「オリジナリティ」がどうすればあるといえるのかという話になって、そういうことを考えると私もだんだんと自信がなくなってきたものである。「やねうら王」を改造したと言っても、元との違いがどの程度あるかということを考え出すと、戦闘機で言ったら「ミラージュ3とネシェル」、爆撃機で言ったら「B29とツポレフ4」、宇宙機で言ったら「スペースシャトルとブラン」くらいの違いしか無いんじゃないかというようなことを思って、せめてVTOLで言ったら「ハリアーとYak38」くらいには違っていないとまずいかな—というようなことを思わないでもない。というか、改造元が強かったせいでそれから大きく変更ができなかったせいで結果的に強い、と言うのは当人である私も余り気分が良くない。対局中に対局相手に話を聞いていると全部自分で作っているという人がたくさ

んいて、そのほうが明らかにすごいのに改造元のソフトが強かったせいで勝ってしまうというのは何か違う気がする。いっそデチューンした方がいいだろうかということも思うがそれもなんか違う気がする（前回の選手権で最下位だった「Mirage」の荒木さんは、全部自分で作って挑戦したから最下位だったわけで、それはむしろ評価すべき事のはずだ）。

SDT5に参加してその時のネタは「定跡のより良い選択手法の考案」ということだったのですが、それをするのが個人的にはだいぶ一杯一杯だったのですがやってみて「そういえば、ネタとして瑣末過ぎないか？」ということに参加してからなんか思ってきたわけです。

なのですが、「定跡を選択するときのよりよい方法を考える」というのは研究テーマとして十分に成立するのではないかということを経験して思いました。それというのも、タヌキさんやカパックさんと話をしていてSDTみたいな時間が限られている勝負のときは序盤は定跡でノータイムで指して時間をセーブする、ということが有効であるということを経験していただいた。お二方に限らず結構色々な人から「時間攻め」という事を聞いて、そんな用語があるのかと勉強になりました。それなので定跡の使用がわりと重要である、ならば定跡のより良い選択方法を考える、ということも十分重要的な研究テーマではないかということを経験して思いました。SDT5の文に書きましたが、2006年頃のモンテカルロ将棋の試行錯誤で、「現時点から10手先の局面を評価関数で評価して」という方法は、メインの探索のアルファベータ法よりも優れてはいないようですが、それでも定跡の選択手法としては一考の価値があるのではないかと、ということを経験して改めて思いました。

それにしても、タヌキさんがSDT5の賞金300万円を全部寄付したとツイッターで見たときはつくづく衝撃的だった。そしてカパックさんがやねさんのほしいもののリストのものを全部買ってプレゼントしたというのを読んだときも衝撃的でした。SDT5で彼らはタヌキのキグルミを着たり狼の被り物をかぶったりしていた。そんなような人がどうして誰よりも立派なことができるのだろうかということが改めて衝撃的でしょうがありません。私はおそらくはこういうふうな振る舞いは出来ないだろうと思わずにはられません。そしてそんなんだから私は彼らに勝てないのだろうなということも改めて思いました。

あとは改めて思うのが「きふわらべ」の高橋さんの素晴らしさです。彼の著作物を読んでいて「数学というのはこんなにも面白いものだったのか」というのをほとんど初めて知った（学校で数学をちょっとやったというかやらされたけどあんまり面白いものだと思えなかった）。高橋さんこそまさに現代のラマヌジャンであるというようなことを個人的に思ってしまう「なんでこんなにすごい人があ

んまり評価されていない感じなのだろう」というか、「きふわらべ」の高橋さんはものすごい能力者だと思うのだけど、正当に評価されていない感じがして何なのだろうかということと思う。

これからのコンピュータ囲碁・将棋の開発について、示唆的だと思ったのがSDT5でHoneyWaffleさんと対局して話を聞いていた時のことです。HoneyWaffleさんが「現状では居飛車が優勢で振り飛車は不利とされているが、それは単に多数派が居飛車だからそうなっているだけでよく研究したら振り飛車はそんなに不利じゃない可能性がある」という事をおっしゃっていて、たしかに単純に強いソフトを作るとかよりも「振り飛車は居飛車に比べて本当に不利なのか？」を研究するというこのためにコンピュータ将棋のソフトを作る、というようなことがこれからの囲碁・将棋のソフト開発では重要なテーマとしてなってくるんじゃないかという事を思います。有名なテーマかも知れませんが、そもそも「先手・後手必勝なのか？」とか、そういうことがコンピュータ将棋のソフトを作るということからは導けるのではないかと思って、単純に強さを競うよりもそういうことをすることのほうがよほど大切なことなんじゃないかということをHoneyWaffleさんの開発コンセプトから考え込んでしまっただろうがないです。

・ソフト名「ArgoCorse_IcSyo（あるごころせいっしょ）」について

SDT5の時にこのソフト名を誰も読めなかったのでなんか悪いような気分になる。ただ、個人的にレースが好きなのでこんな名前にしたけど、レーシングチームはスポンサー名がチーム名に入るので「オートボックス・レーシング・チーム・アグリ」とか「伊藤忠エネクス・チーム・インパル」とか、正式名称が妙に長かったりする。それなので読むほうが適当に略して読むのが慣習的なのでそういうふうなことでいいかと思うことにしたい。ちなみに、富士スピードウェイとかは、コーナーごとにスポンサーが付いていて「コカコーラ・コーナー」「アドヴァン・コーナー」「ダンロップ・コーナー」「プリウス・コーナー」「パナソニック・コーナー」とか、割と読みやすい名前が良いけど、鈴鹿サーキットのシケインは「日立オートモティブシステムズ」がスポンサーになったので、解説者はシケインに言及するたびにこのスポンサー名を言わないといけならしく「ただいま先頭車両が日立オートモティブシステムズシケインを通過しました」というようなことを毎回言うのがいかにも言いにくそうだ。だから「鈴鹿サーキットのシケイン名よりは読みやすからいいか」ということにする。

・羽生竜王と井山七冠の国民栄誉賞受賞について

優れた棋士が評価されるのは大変に良いことだと思います。お二方とも偉大な棋士であることは明らかであると思うので順当ではないかと思います。

井山七冠は世界戦を重視されているという話を読んだので、先日のLG杯決勝も注目していたのですが、第二局を劇的に逆転勝ちして三局目に期待していたのですが、惜しくも敗れてしまって、今回は残念でしたがまた近いうちに世界を制してくれるものと個人的に期待していたら、その後のワールド碁チャンピオンシップでも決勝に進んで、さすがだと思っていたら今回も残念だったのですが、また近いうちに世界を制してくれるものと個人的に期待しています。というか単に時間の問題だと思いますが。

また、それにしても藤井六段が強すぎますね。Twitterでよく書かれているとおりですが「アニメや漫画の主人公の設定だったらリアリティがないと言って編集に蹴られる」ようなことを現実に行っている様子はまさに驚愕です。

・アニメ「りゅうおうのおしごと！」の感想

1月から将棋をテーマにしたライトノベル「りゅうおうのおしごと！」のアニメが放送されました。面白い作品だと思ったので以下に感想を書きます。

> りゅうおうのおしごと！ 第一局「押しかけ弟子」

原作既読組。アニメ版はコメディ色が原作よりも強いと思った。原作はもうちょっと重っ苦しい感じ。ただ個人的には原作の重っ苦しい部分が好きなのでアニメ版がコメディによりすぎて重い部分の良さが薄くならないかなとちょっと心配している。

最初のハ一が竜王戦をフルセットで竜王位を奪取するシーン。ハ一が勝ちまで読み切るが疲労と緊張で体が動かずに駒を持つことが出来ない、それどころか「発声しても着手は認められるが発声すら出来ない」というほどに追い詰められているところに、あいから水をもらうシーン。原作を読んでいるときはここまでへ口へ口になって死闘を戦っているということに衝撃を受けたものであるが、アニメ版では妙にあっさりとスルーしたなって感じなところがちょっと惜しい。そもそも「発声しても着手が認められる」というのが豆知識として妙に面白かった。「月下の棋士」を読んでいるときに、最終回で主人公が

名人戦の対局で「早く指せ、俺が投了できないだろう」という台詞があり、私「そうか、投了って自分の番じゃないとできないのか」というのが豆知識として妙に面白かったのだが（作中では名人が自分の手番で対局を放棄して時間切れで主人公の勝ちになる）、「りゅうおうのおしごと」の原作の最初のシーンでも発声で着手が認められるということとその発声すら出来ない、という状態がなんだか妙に印象的だった。

あとは、あいを読むシーンも一話から出てきてすごく良かったと思った。原作を読んでいてあいを読むシーンがものすごく印象的だったのでこれをアニメ版ではどういうふうに描いてくれるのか、ちゃんと描けるのかというのが気になっていたのだが、素晴らしいと思う演出で、「日高里菜グッジョブ！」と見ながら心で思った。「ハチワンダイバー」においては「読む」という行為を「深い水の中に潜る」というメタファーで表現していたが、本作は漫画ではなくて小説である。その文章での表現をこういうふうにアニメでもちゃんと表現できることが本当に素晴らしいと思った。

個人的には原作では銀子が一番好きなキャラなのだが、アニメ化発表のときに銀子の声を金元寿子がやるとあったので、そのときにやっていた「妹さえいればいい。」の可児那由多の声の印象が強すぎて、「これが銀子になるのか、え、どうやって？」と思っていたのだが、アニメ版の一話を見ると「なるほどなるほど」となかなか納得する声で良かった。

原作のプロローグは大阪城公園の満開の桜の下で八一とあいが将棋を指すシーンで、それがすごく印象的で良かったのでアニメ版ではどうするのだろうかと思っていたらOPで使っていて「ああ、なるほど」と思った。

「将棋+萌え」の作品では「駒ひびき」が個人的に好きだったのだが、あんま受けなかったらしく3巻で終わったのが残念だなと改めて思う。20巻くらい読みたかった。

星4つ。

> りゅうおうのおしごと！ 第二局「弟子のいる日常」

JS研の設立とか、あいの両親が来ることとか。1話を受けて話を展開してきた。歩夢との対局はほぼ原作通り。本作に限らず主人公が女の子とばっかり仲が良い、というなかで同性の友人がいるということが貴重であると思った。原作でももっと男性棋士を出してほしいということは思う。あとは、あいのアホ毛がなにげにいい味を出している。あのアホ毛の動きはアニメ版で初めて実現できたことで実に良い。研修会試験で全勝しろというところでの引きがうまい。「勝敗は関係ない」といっているのに勝敗にこ

だわりまくるところがなんか将棋指しっぽくて妙なリアリティが有る。

星3.5。

> りゅうおうのおしごと！ 第三局「研修会試験」

研修会の入会試験。対局シーンが凄く良かった。2話の八一と歩夢の対局は演出で描いていたが、今回のあいの対局は、「将棋を指す」という行為をガチで描いていてそれでいて見応え充分だった。原作でも将棋を指すシーンを文章で表現するということが出来ていることがすごいと思ったが、アニメでも将棋を指すシーンをここまで見応えがある映像として描けていることがすごいと思った。

それにしても、原作を読んでいると銀子がすごく萌えるのに、アニメ版では銀子の萌えエピソードを結構まるまるカットしているっぽいのがファンとしては実に残念である。

星4つ。

> りゅうおうのおしごと！ 第四局「もう一人のあい」

天衣登場。原作ではなんのためにいるのだからよく分からないキャラだと思っていたが、アニメでは非常に映えるなと思った。声が佐倉綾音というのも非常に良い。予想以上にロリ声だったのもうちよつとボーイッシュな感じで演じてくれても良かったんじゃないかと思わないでもないが「ニセコイ」の春ちゃんみたいな感じでこれはこれで良いと思う。

新世界の将棋屋に行くエピソードは、原作では真剣をしていたのでアニメ版でどういうふうに描いてくれるのかと楽しみにしていたのだが、アニメ版では真剣をしないことになっていたのもちょっと残念な気はする。「タバコに千円札を巻きつけるんだ」とか、妙にディテールが細かくて新世界ならばほんとうに真剣をやっているところがあるが面白かったのだが。個人的には天衣がタバコの箱を見せて真剣を誘うシーンが文章だとよくわからなかったのがビジュアルがついたらどうなるのだろうというのが見たかったのがそれが見れなかったのが残念なのだが。

ラストの引きが非常に良い。

星4.5。

> りゅうおうのおしごと！ 第五局「天衣無縫」

あいVS天衣の対局シーンが見応えがあった。文字通りに目の色が変わるというアニメ版での描写が非常に面白い。そしてあいの瞳の色がもとに戻ることで負けを表現するところとかすごく良かった。今回の対局は演出と解説で見せていたが、これまでの将棋の対局シーンの見せ方が違っているところがすごいなと思う。天衣に負けて泣いているあいのシーンが印象的すぎて、日高里菜はうまいなーとまた思った。というかよく分からないが将棋指しと言うのはこういうふうに考えるものなのだろうかというのもよくわからなかったことなので負けることがここまで悔しいというそれが印象深かった。

星4つ。

> りゅうおうのおしごと！ 第六局「オールラウンダー」

終盤の八一が将棋を指すシーンが凄く良かった。これまではあいが将棋を指すシーンに注力していた感じがしたが、今回の八一の対局シーンが実に見ごたえがあったと思う。原作だと将棋についての豆知識が結構はいつているのだが、アニメ版ではあまり深く将棋の話をする事が出来ないのがちょっと残念な気がする。今回のように振り飛車と居飛車の違いが、ってやるのが限界なのかなと。個人的には一手損角換わりについての解説がこれを読んで初めて意味を知ったのですごく面白かったのだが。それでもエッセンスは出ていると思う。

星3.5。

> りゅうおうのおしごと！ 第七局「十才のわたしへ」

原作既読組は「はしよりすぎ」として評価が低いというのを見る前にTwitterで見ていたのだが、たしかに原作の重さからしたら「はしよりすぎ」には違いがないということは思うが、それでもワンクールのアニメでやるにはこれ以上は出来ないだろうということはわかるし、エッセンスは伝わる内容だったと思うからこれはこれでいいんじゃないかなと思った。

序盤の八一が読むシーンが格好良すぎた。あいが読むシーンとの対比で八一が読むシーンも原作では非常に面白いのだが、今回の八一の読むシーンを見ていて「こんなにこいつは格好良く読んだっけ？」とは思った。

星4つ。

> りゅうおうのおしごと！ 第八局「はじめての大会」

マイナビオープン編。原作を読んでいると「これまでの総力戦」という感じがしていたが、アニメで見ているとどうしてもダイジェストな感じがして重さと言うか凄さがいまいち伝わりにくい感じがする。やっぱり天衣とか桂香さんのことをもっと掘り下げて描いてほしかったという気がする。そこの掘り下げが足りていないので見ていて軽く感じられてしまうのだ。

星3.5。

> りゅうおうのおしごと！ 第九局「八月一日」

祭神雷のキャラが強くてすごく良かった。戸松遥が本気を出して演じたら視聴者がドン引きするような狂気を演じられると思ったが、視聴者が引かないようにマイルドに演じてくれたのかなと思った。

あいVS祭神の対局シーンがまた見応え充分ですごく良かった。王道ではあるが、あいの「あなたに私の竜王が取れますか！」のダブルミーニングのセリフが面白かった。

星4.5。

2018/3/31追記

日付がドラフト時点の3/20だったので提出時点の3/26に変更。

ライブラリの選定理由を追記します。

「やねうら王」コードがわかりやすかったから。

「elmo,Qhapaq,tanuki-,Apery」評価関数が強いから。

以上です。

第 28 回世界コンピュータ将棋選手権

dlshogi アピール文章

山岡忠夫
2018 年 5 月 1 日 更新

※下線部分は、第 5 回将棋電王トーナメントからの差分を示す。

1 特徴

- ディープラーニングを使用
- 指し手を予測する Policy Network
- 局面の勝率を予測する Value Network
- 入力特徴にドメイン知識を活用
- モンテカルロ木探索
- 並列化
- 自己対局による強化学習
- 既存将棋プログラムの自己対局データを使った事前学習
- REINFORCE アルゴリズムによる Policy Network の学習
- ブートストラップ法による Value Network の学習
- マルチタスク学習
- 詰み探索
- 序盤局面の事前探索（定跡化）
- マルチ GPU 対応
- CUDA、cuDNN を直接使用
- GPU ごとに異なるモデルの読み込み

2 使用ライブラリ

- elmo¹ (Commits on May 29, 2017)
- Apery(elmo の派生元) (14 Apr, 2017 commit:e3eb33ffa6aa840765d2e2efdacf1c618528a3be)

※学習データ生成、局面管理、合法手生成のために部分的に使用

2.1 ライブラリの選定理由

本プログラムは、将棋におけるディープラーニングの適用を検証することを目的としており、学習局面生成、局面管理、合法手生成については、使用可能なオープンソースがあれば使用する方針である。そのため、学習局面を圧縮形式(hcpe)で生成する機能と読み込む機能

¹ https://github.com/mk-takizawa/elmo_for_learn

を備えており、合法手生成を高速に行える elmo(派生元 Apery)を選定した。

3 各特長の具体的な詳細（独自性のアピール）

3.1 ディープラーニングを使用

DNN(Deep Neural Network)を使用して指し手を生成する。

従来の探索アルゴリズム(α β 法)、評価関数(3 駒関係)は使用していない。

3.2 Policy Network

局面の遷移確率を Policy Network を使用して計算する。

Policy Network の構成には、Wide Residual Network²を使用した。

入力の畳み込み 1 層と、ResNet 10 ブロック(畳み込み 2 層で構成)と出力層の合計 22 の畳み込み層で構成した。フィルターサイズは 3 (入力層の持ち駒の面のみ 1)、フィルター枚数は 192 とした。

3.3 Value Network

局面の勝率を Value Network を使用して計算する。

Value Network は、Policy Network と出力層以外同じ構成で、出力層に全結合層をつなげ、シグモイド関数で勝率を出力する。

3.4 入力特徴にドメイン知識を活用

Alpha Zero では、入力特徴に呼吸点のような囲碁の知識を用いずに盤面の石の配置と履歴局面のみを入力特徴とすることで、ドメイン知識なしでも人間を上回ることが示された。しかし、その代償として、入力特徴にドメイン知識を活用した AlphaGo Lee/Master に比べて倍のネットワークの層数が必要になっている。AlphaGo Zero の論文の Figure 3 によると、ネットワーク層数が同一のバージョンでは Master を上回る前にレーティングが飽和している。

強い将棋ソフトを作るという目的であれば、積極的にドメイン知識を活用した方が計算リソースを省力化できると考えられる。

そのため、本ソフトでは、入力特徴に盤面の駒の配置の他に、利き数と王手がかかっているかという情報を加えている。それらの特徴量が学習時間を短縮する上で、有効であることは実験によって確かめている。

3.5 モンテカルロ木探索

対局時の指し手生成には、Policy Network と Value Network を活用したモンテカルロ木探索を使用する。

² <https://arxiv.org/abs/1605.07146>

ノードを選択する方策に、Policy Network による遷移確率をボーナス項に使用した PUCT アルゴリズムを使用する。PUCT アルゴリズムは、AlphaGo の論文³に掲載された式を使用した。

また、末端ノードでの価値の評価に、Value Network で計算した勝率を使用する。

通常のモンテカルロ木探索では、末端ノードからプレイアウトを行った結果（勝敗）を報酬とするが、プレイアウトを行わず Value Network の値を使用する。

3.6 並列化

複数スレッドで並列化を行う。並列化の方式には、スレッド間でゲーム木のノード情報を共有するツリー並列化を採用する。

モンテカルロ木探索は、並列化が容易だが、Policy Network と Value Network を計算するための GPU の数以上の並列化を行うと GPU の使用で競合が発生する。

GPU は複数の計算要求をバッチで処理することが可能であるため、各スレッドからの要求をキューイングして、専用スレッドでバッチ処理することで競合を回避する。

3.7 自己対局による強化学習

Alpha Zero⁴と同様の方式で強化学習を行う。自己対局により教師局面を生成し、その教師局面を学習したモデルで、再び教師局面を生成するというサイクルを繰り返すことでモデルを成長させる。

※教師ありより強くなっていないため、大会では使用しない

3.8 既存将棋プログラムの自己対局データを使った事前学習

本プログラムを使用して、Alpha Zero と同様に、ランダムに初期化されたモデルから強化学習を行うことも可能だが、使用可能なマシンリソースが足りないため、スクラッチからの学習は行わず、既存将棋プログラムの自己対局データを教師データとして、教師あり学習でモデルの事前学習を行う。

教師データには、elmo で生成した自己対局データを使用する。

3.9 REINFORCE アルゴリズムによる Policy Network の学習

単純に自己対局の指し手を学習するのではなく、学習局面の価値と勝敗データと関連付けて学習を行う。良い局面から負けになった手は、悪手として負の報酬を与え、悪い局面から勝ちになった手は善手として正の報酬を与える。学習アルゴリズムには、AlphaGo の論文に掲載されている REINFORCE アルゴリズムを使用した。

³ <https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf>

⁴ <https://arxiv.org/abs/1712.01815>

3.10 ブートストラップ法による Value Network の学習

Value Network の学習の損失関数は、勝敗を教師データとした交差エントロピーと、探索結果の評価値を教師データとした交差エントロピーの和とした。

このように、本来の報酬（勝敗）とは別の推定量（探索結果の評価値）を用いてパラメータを更新する手法をブートストラップという。

経験的にブートストラップ手法は、非ブートストラップ手法より性能が良いことが知られている。

3.11 マルチタスク学習

Policy Network と Value Network のネットワーク構成が同じ層を共通化し、出力層を分けることで、同時に学習を行う。

関連する複数のタスクを同時に学習することをマルチタスク学習という。タスク間に関連がある場合、単独で学習するよりも精度が向上する。

また、対局時に Policy Network と Value Network を同時に計算できるため、高速化の効果もある。

3.12 詰み探索

モンテカルロ木探索は最善手よりも安全な手を選ぶ傾向があるため詰みのある局面で手を抜くことがある。

対策として、詰み専用の探索を行い、詰みの場合はその手を指す。

また、モンテカルロ木探索の末端ノードでも、数手の詰み探索を行い、詰みの局面を評価できるようする。Policy Network と Value Network の計算中に、CPU が待ち状態の間に詰み探索を行うため、探索速度が落ちることはない。

3.13 序盤局面の事前探索（定跡化）

出現頻度の高い序盤局面は、対局時に探索しなくても、事前に探索を行い定跡化しておくことができる。また、事前に探索することで、対局時よりも探索に時間をかけることができる。

定跡データには、局面に対して複数の手を記録し、それぞれの手に探索時の訪問回数を記録しておく。対局時に定跡を使用する際は、訪問回数に応じた確率で手を選択することで、ゲームの進行が固定されないようにする。

3.14 マルチ GPU 対応

複数枚の GPU を使いニューラルネットワークの推論を分散処理する。

GPU ごとにキューを処理する専用スレッドを割り当てる。キューを処理するスレッドに対

して、複数のモンテカルロ木探索を行うスレッドを割り当てる。ゲーム木のノード情報は、異なる GPU に割り当てた探索スレッド間で共有する。

3.15 CUDA、cuDNN を直接使用

モデルの学習にはディープラーニングフレームワークとして Chainer を使用しているが、対局プログラムには、ディープラーニングフレームワークを用いず CUDA、cuDNN を直接使用する。Chainer で学習したモデルを読み込み、推論を行う処理をスクラッチで開発した。

高速化と対局の実行環境にディープラーニングフレームワークの環境構築を不要とすることを目的とする。

3.16 GPU ごとに異なるモデルの読み込み

モデルごとに誤る確率が独立である場合、複数モデルが同時に誤る確率は、単一のモデルを使用する場合より低くなる。GPU ごとに異なるモデルを読み込むことで、探索の精度を向上が期待できる。

4 学習について

4.1 自己対局による強化学習

4.1.1 学習データ、パラメータ

- 事前学習データ：elmo(wsc27)で深さ 8 で生成した 4.9 億局面
- ミニバッチサイズ：64
- 学習アルゴリズム：Momentum SGD (学習率 0.01、慣性係数 0.9)
- 強化学習 1 サイクルで生成する局面：500 万局面
- 強化学習のイテレーション数：18 (4 月末時点)

4.1.2 学習結果

【事前学習を行ったモデル】

- Policy Network の一致率：45.6%
- Value Network の一致率：78.1%

※ 既存将棋プログラムの自己対局データを増やすことでさらに一致率を上げられるが、Alpha Zero 方式の強化学習によって成果を上げたいため、収束する前のモデルから開始した。

4 月末時点で、18 サイクルの強化学習を実施した結果、学習開始モデルに対して、1 手 3 秒 50 回対局で勝率 81%となり、有意に強くすることができた。しかし、教師ありで収束するまで学習したモデルよりまだ弱いため、大会では強化学習したモデルは使用しないことにする。

4.2 教師ありによる学習

4.2.1 学習データ、パラメータ

- elmo(wcsc27)で深さ 8 で生成した 11 億局面
- ミニバッチサイズ : 64
- 学習アルゴリズム : Momentum SGD (学習率 0.01~0.0001、慣性係数 0.9)

4.2.2 学習結果

- Policy Network の一致率 : 46.1%
- Value Network の一致率 : 78.1%

以上

W@ndre アピール文

2018/03/27

【由来】

W@ndreはwanderでwonderな将棋を指してほしいという意図で命名したプログラムです。

【目標】

長期的：角頭歩戦法を指しこなしてもらう。

今回：ディープラーニングのプログラムと既存プログラムを組み合わせで戦う。

【使用ライブラリ】

dlshogi

やねうら王

【定跡部】

角頭歩を指します。角頭歩ができない（▲26歩△34歩▲25歩のオープニング等）ときは阪田流向飛車を指します。

昨年の第5回将棋電王トーナメント（以下SDT5）ではプロの棋譜やFloodgateの棋譜を利用して定跡を作成していましたが、

特定の変化に対して角頭歩側にマイナスとなる変化も多々含まれていました。

今回は自己対戦や手調整で特定局面での探索結果を反映させることで、簡単に不利な局面に持っていられないよう工夫を試みています。

【評価関数】

やねうら王で利用するKPPT型評価関数について3種類の評価関数を用意したので、それぞれを単体かブレンド(*a)して使い分ける予定です。

1.

昨年の選手権以降、教師局面の作成時に定跡を用いると、定跡を指しこなせるように見える評価関数が出来ることが確認されています。[1], [2], (*b)

本選手権では、上記【定跡部】で作成した定跡を用い、ゼロベクトルから強化学習を行い評価関数を作成しました。

作成された評価関数を元に定跡を改良し、その定跡で評価関数の追加学習を行うというサイクルを回しました。

2.

SDT5にてHoneyWaffleが行っていた学習法[3]を参考に、特定の形に強制的に減点・加点することで角頭歩局面を評価する評価関数を作成しました。

3.

やねうら王ライブラリより配布されていた教師局面データ(*c)で、評価関数バイナリ[181_0020G]を再学習させました。

1, 2は角頭歩戦法を理解できるかの試みとして作成した評価関数、

3はやねうら王ライブラリで配布されている教師局面と評価バイナリを学習の練習も兼ねて作成した評価関数です。

また、dlshogiに関して、3の教師局面を学習させたモデルを利用する予定です（執筆時点）

【探索部】

序盤は2駒や3駒より表現力の高いディープラーニング系の評価関数と探索手法が優れている気がしたので(*d, *e, *f, *g, *h)、序盤に関してはdlshogiを利用し、

中終盤以降はやねうら王探索部に頑張ってもらいます。

現時点ではライブラリの探索部に変更を加える予定はありませんが、

やねうら王ライブラリ利用者の参加状況次第では、探索延長部のパラメタに手を加えるかもしれません。[4][5]

【謝辞】

本プログラムを作成するにあたり、やねうら王ライブラリとdlshogiライブラリの作者様に深く感謝いたします。

また、金沢将棋、白砂将棋、Labyrinthus、HoneyWaffleの存在なくしてはW@ndrelはありませんでした、アイデアをくださった皆様に御礼申し上げます。

【参考文献】

[1] 棋風を覚える将棋ソフトが完成してた件

<http://yaneuraou.yaneu.com/2017/06/26/%E6%A3%8B%E9%A2%A%E3%82%92%E8%A6%9A%E3%81%88%E3%82%8B%E5%B0%86%E6%A3%8B%E3%82%BD%E3%83%95%E3%83%88%E3%81%8C%E5%AE%8C%E6%88%90%E3%81%97%E3%81%A6%E3%81%9F%E4%BB%B6/>

[2] 素人による評価関数自作記---ゼロから自分好みの評価関数を育成

<http://www.uuunuuun.com/single-post/2017/06/22/%E7%B4%A0%E4%BA%BA%E3%81%AB%E3%82%88%E3%82%8B%E8%A9%95%E4%BE%A1%E9%96%A2%E6%95%B0%E8%87%AA%E4%BD%9C%E8%A8%98---%E3%82%BC%E3%83%AD%E3%81%8B%E3%82%89%E8%87%AA%E5%88%86%E5%A5%BD%E3%81%BF%E3%81%AE%E8%A9%95%E4%BE%A1%E9%96%A2%E6%95%B0%E3%82%92%E8%82%B2%E6%88%90>

[3] HoneyWaffle 第5回将棋電王トーナメント版

<https://github.com/32hiko/HoneyWaffleSDT5>

[4] 魔女より強くすると本当は強くない？！

<http://yaneuraou.yaneu.com/2016/06/14/%E9%AD%94%E5%A5%B3%E3%82%88%E3%82%8A%E5%BC%B7%E3%81%8F%E3%81%99%E3%82%8B%E3%81%A8%E6%9C%AC%E5%BD%93%E3%81%AF%E5%BC%B7%E3%81%8F%E3%81%AA%E3%82%89%E3%81%AA%E3%81%84%EF%BC%9F%EF%BC%81/>

[5] Stockfish DD - search 探索部

<http://yaneuraou.yaneu.com/2015/02/24/stockfish-dd-search->

[%E6%8E%A2%E7%B4%A2%E9%83%A8/](http://yaneuraou.yaneu.com/2015/02/24/stockfish-dd-search-%E6%8E%A2%E7%B4%A2%E9%83%A8/)

(*a) 評価関数のキメラ化コマンド公開しました

<http://yaneuraou.yaneu.com/2017/06/28/%E8%A9%95%E4%BE%A1%E9%96%A2%E6%95%B0%E3%81%AE%E3%82%AD%E3%83%A1%E3%83%A9%E5%8C%96%E3%82%B3%E3%83%9E%E3%83%B3%E3%83%89%E5%85%AC%E9%96%8B%E3%81%97%E3%81%BE%E3%81%97%E3%81%9F/>

(*b) 定跡の生成に使った評価関数を用いるべき？

<http://yaneuraou.yaneu.com/2014/12/27/%E5%AE%9A%E8%B7%A1%E3%81%AE%E7%94%9F%E6%88%90%E3%81%AB%E4%BD%BF%E3%81%A3%E3%81%9F%E8%A9%95%E4%BE%A1%E9%96%A2%E6%95%B0%E3%82%92%E7%94%A8%E3%81%84%E3%82%8B%E3%81%B9%E3%81%8D%EF%BC%9F/>

(*c)-1 将棋ソフトの機械学習の成否を判定するための資料

<http://yaneuraou.yaneu.com/2017/05/26/%E5%B0%86%E6%A3%8B%E3%82%BD%E3%83%95%E3%83%88%E3%81%AE%E6%A9%9F%E6%A2%B0%E5%AD%A6%E7%BF%92%E3%81%AE%E6%88%90%E5%90%A6%E3%82%92%E5%88%A4%E5%AE%9A%E3%81%99%E3%82%8B%E3%81%9F%E3%82%81%E3%81%AE%E8%B3%87/>

(*c)-2 depth10で作った110億局面の教師データ、期間限定で公開します

<http://yaneuraou.yaneu.com/2017/11/21/depth10%E3%81%A7%E4%BD%9C%E3%81%A3%E3%81%9F110%E5%84%84%E5%B1%80%E9%9D%A2%E3%81%AE%E6%95%99%E5%B8%AB%E3%83%87%E3%83%BC%E3%82%BF%E3%80%81%E6%9C%9F%E9%96%93%E9%99%90%E5%AE%9A%E3%81%A7%E5%85%AC%E9%96%8B/>

(*c)-3 クリスマスプレゼントの配布はじめました

<http://yaneuraou.yaneu.com/2017/12/25/%E3%82%AF%E3%83%AA%E3%82%B9%E3%83%9E%E3%82%B9%E3%83%97%E3%83%AC%E3%82%BC%E3%83%B3%E3%83%88%E3%81%AE%E9%85%8D%E5%B8%83%E3%81%AF%E3%81%98%E3%82%81%E3%81%BE%E3%81%97%E3%81%9F/>

(*c)-4 月刊教師局面 2018年1月号

<http://yaneuraou.yaneu.com/2018/01/23/%E6%9C%88%E5%88%8A%E6%95%99%E5%B8%AB%E5%B1%80%E9%9D%A2-2018%E5%B9%B4%E6%9C%88%E5%8F%B7/>

(*d) https://twitter.com/messiah_ai/status/920236374527053824

(*e) https://twitter.com/ymg_aq/status/940899696389767168

(*f) https://twitter.com/cute_na_piglets/status/944583388874153984

(*g) <https://twitter.com/hillbig/status/938910529396936704>

(*h) アピール文から読み解く WCSC27の見どころ（技巧編）

<http://qhapaq.hatenablog.com/entry/2017/04/18/210624>

(*i) 開発メモ：角交換に特化した高速化

<http://d.hatena.ne.jp/LS3600/20100117>

*前回(SDT5)のアピール文書

<http://denou.jp/tournament2017/img/pr/wandre.pdf>