



十六式いろは焔（きらめき）

第 33 回世界コンピュータ将棋選手権
アピール文



メンバー

末吉竜介、宇井絢音、市川翠、加藤凜、Phan Xuan Hoa、
有賀宏樹、Panwar Abhishek、大熊琉斗、茂木海輝



「十六式いろは 煌」の由来

昨年 2022 年に 2 期生の先輩達が決めた「十六式いろは煌（きらめき）」を今年も引き継ぎます。

以下、2022 年当時の由来の記載です。

様々な名前の候補が上がり最終的に決まったのが考え始めてからなんと1か月かかりました！。
皇（すめらぎ）、煌（きらめき）、日本工学院、かまトウ（学校のマスコットキャラ）…などなど。
「日本工学院の名前があった方がよいのではないか」や
「ローマ字で書いた方がかっこいい！」などかなりの意見などがありました。
最終的には末吉先生の「十六式いろは」と生徒達で考えた「煌（きらめき）」を
組み合わせて決定しました！



ソフトの概要

採用予定

- dlshogi
- やねうら王
- KomorningHeights
- Electron 将棋

ソフトの説明（予定）

- dlshogi のモデルを軽量なものに変更
- やねうら王での評価関数の学習
- 定跡ファイルの作成
- floodgate、AobaZero の棋譜の利用
- 詰将棋エンジンを含めて合議制の見直し



WCSC33 での実装について（2023-05-11 追記）

（次のページから記します）



基本的な動作

昨年 2022 年の wcsc32 と同様。

dlshogi とやねうら王 (★) の両エンジンによる
楽観合議がこのソフトの最大の特徴。

Ayane を使用して両エンジンを呼び出し、評価値を比較して指し手を決める。

★ 昨年の評価関数は「水匠 5」を使用していたが、
今年は独自のものに変更。詳細は後述。



wcsc32 時との違い

- 1) やねうら王の評価関数の変更（きらめくゼロ NNUE 評価関数）
- 2) 定跡ファイルを新規で作成（きらめく Aoba 広葉樹定跡）
- 3) dlshogi の学習モデルの変更
- 4) その他の変更（持ち時間や先読み機能等の改修）



変更 1) やねうら王の評価関数の変更、その 1

水匠 5 から、以下の内容で独自に作成したものに変更
人間の棋譜を用いず、AobaZero の棋譜を元に作成した定跡（後述）を使い、
ゼロからの NNUE 評価関数を、自己対戦による棋譜を用い強化学習させた。
「きらめくゼロ NNUE 評価関数」（略称：キラゼロ評価関数）を命名。

深さ 1 から 6 まで深さを深くし、自己対戦で生成した棋譜による強化学習を行った。
基盤となる第 1 段階、1 回 2000 万局面の棋譜での学習を 10 回行う。（計 2 億局面）
← 最初に深さ 1（棋力が弱い＝精度が低い）と浅い理由は
深いものを使っても、すぐに精度が向上しないため。



変更 1) やねうら王の評価関数の変更、その 2

第 2 段階、定跡に沿うように序盤用に生成した
深さ 4、評価値の上限 200、16 手目までの
2000 万局面の棋譜を用い機械学習を行った。

← 経験上、評価値 200 までが形勢は互角とみなし
定跡で置き換えられると考えたため。

序盤は定跡を使うためとは言え、定跡が外れた場合に
見当外れな手を指す可能性を減らすために学習させた。



変更 1) やねうら王の評価関数の変更、その 3

第 3 段階、中盤用に生成した

深さ 6、評価値の上限 1200、17 手目以降の 1 億局面の棋譜を用い、
lambda は 0.1 (←深さ 6 は棋力が低いので勝率項を低く) で機械学習を行った。
← これも経験上、評価値 1200 くらいまでが勝負の要であるとみて、
この機械学習部分は最重要視している。

lambda について

elmo 式 $dsig = \text{勝敗項} + \text{lambda} * \text{勝率項}$

lambda は 0.5 がデフォルト。大きい (0.8 や 1 など) ほど勝率項に重みが出る。



変更 1) やねうら王の評価関数の変更、その 4

第 4 段階、序盤から中盤の深さ 6、評価値の上限 1200 の 1 億局面の棋譜を用い機械学習させるセットを 1 ループに、9 ループ行った。(計 9 億局面)

大会までの時間の関係で、最深の深さ 6 の 12 億 2000 万局面の棋譜の学習まで。なお、最新のやねうら王 v7.6.1 は学習部にデグレがあるため v5.3.3 で学習。

ちなみに、この評価関数を用いて CPU:Core i7-9700K のマシンで floodgate のレーティングは 3613
(以降、floodgate のレーティングを R とし、R3613 と略す)



変更 2) 定跡ファイルを新規で作成、その 1

AobaZero の棋譜 (50700000 番台) の 10 万局面から
やねうら王を使って定跡ファイル化する。
また、その定跡ファイルを ShogiGUI で読み込める形式に
変換したのもも利用する。(時間の関係上、試験的なものになった)

上記の定跡ファイルから 1 手目、2 手目は出現回数の多いものに絞る。
← 最深部の局面数を減らすため。

出現回数 1 回以下は切り捨て。最上位の 10 % 未満も切り捨て。



変更 2) 定跡ファイルを新規で作成、その 2

各局面に変更 1) で作成した NNUE 評価関数を使い
深さ 24 で探索した評価値を付ける。

やねうら王で定跡をテラショック化 (ツリー化) する。
評価値を付けた元の定跡ファイルと、
テラショック化した定跡ファイルをマージする。
← 幅広い定跡にするため

評価値 -150 未満の行を削除してスリム化する。



変更 2) 定跡ファイルを新規で作成、その 3

64 手目までの定跡ファイルだが、今回は手数を無視した設定で利用した。

登録されている城跡の局面数は約 13 万局面。

「きらめく Aoba 広葉樹定跡」 (略称：キラショック Aoba 定跡)
←AobaZero の棋譜を使っている。テラショック化＝ツリー化した。
通常のテラショックよりも初期の選択枝が広い。以上のことから命名。



変更 3) dlshogi の学習モデルの変更、その 1

まず「標準 dlshogi のネットワーク名の書き方」を確認する。

例)

senet3x382_fcl128_reduction8_relu

←3 ブロック, 382 チャンネル, ReLU 関数, 全結合層 128,
ボトルネックの係数 8 の SE Net という意味。

ソースコードでの正規表現

```
^(resnet|senet)(¥d+)(x¥d+){0,1}(_fcl¥d+){0,1}(_reduction¥d+){0,1}(_.+){0,1}$
```

変更 3) dlshogi の学習モデルの変更、その 2

前述の正規表現内の簡単な説明

- 1: (resnet|senet) ネットワークの種類 (2 択)
- 2: (¥d+) ブロック数。小さいほど軽量化する。
- 3: (x¥d+){0,1} フィルター数は 1 つのブロックで扱うチャンネル数。
小さいほど軽量化する。
- 4: (_fcl¥d+){0,1} 全結合層 (Fully Connected layer) の出力サンプルのサイズ。
小さいほど軽量化する。
- 5: (_reduction¥d+){0,1} senet 用のボトルネックの係数
(デフォルトの 8 か、16 が良いらしい) 大きいほど軽量化する。
- 6: (._+){0,1}\$ 活性化関数の種類 (デフォルトの「_relu」か、「_swish」の 2 択)



変更 3) dlshogi の学習モデルの変更、その 3

今回採用した手法

SENet (ResNet に Squeeze-and-Excitation block を組み合わせたもの) の
3 ブロック 384 チャンネルで、fcl が 128 のネットワークで

書籍「強い将棋ソフトの創りかた」付属の学習データを学習させた。

← ブロック数減らしてチャンネル数を増やしたのは、WideResNet っぽい効果が狙い。

また、出力部の fcl を減らしたのは軽量化のため。

dlshogi の TensorRT 版 (GeForce RTX 2070 SUPER を使用) で R3861
nps (秒間ノード数) は 2 万 7000 程度と dlshogi のモデルとしては高速。



変更 4) その他の変更、その 1

- ゲーミングノート PC から、高性能のゲーミングタワー型 PC にハードを変更。
- どのエンジンの手が選ばれたかの数の出力の改修。
- ゲーム終了時にエンジンを停止する処理を改修。
- やねうら王エンジンでの ponder を出力していなかったところを改修。
- 設定画面を実装。
- Electron 将棋の評価値出力に対応。
- 指定局面戦 sfen 対応と手番の改修。
- 先読み時の両エンジンが同じ指し手のときの改修。
- dlshogi の ponder 処理の改修。



変更 4) その他の変更、その 2

- 20 手目まで序盤用のやねうら王を使用。
キラゼロ評価関数、キラショック定跡を搭載をしたので伸ばした。
- 序盤先読み時間を 3 秒、序盤以降を 20 秒に増やした。
- 序盤の思考時間を 10 秒固定、60 手目までを 40 秒固定に増やした。
60 手目くらいまでが勝負の要になると考え、前回より増やした。



使用ソフト、参考文献等、その1

sueyoshiyosuke/16shiki-Iroha_kirameki:
https://github.com/sueyoshiyosuke/16shiki-Iroha_kirameki
←wcsc32 時のもの。

TadaoYamaoka/DeepLearningShogi
<https://github.com/TadaoYamaoka/DeepLearningShogi>

yaneurao/YaneuraOu
<https://github.com/yaneurao/YaneuraOu>

Electron 将棋
<https://sunfish-shogi.github.io/electron-shogi/>

Home · yaneurao/YaneuraOu Wiki
<https://github.com/yaneurao/YaneuraOu/wiki>

tttak/GougiShogi: 合議将棋
<https://github.com/tttak/GougiShogi>



使用ソフト、参考文献等、その2

yaneurao/Ayane:

<https://github.com/yaneurao/Ayane>

AobaZero

<http://www.yss-aya.com/aobazero/>

floodgate

<http://wdoor.c.u-tokyo.ac.jp/shogi/floodgate.html>

次世代の将棋思考エンジン、NNUE 関数を学ぼう (その1. ネットワーク構造編)

- コンピュータ将棋 Qhapaq

<https://qhapaq.hatenablog.com/entry/2018/06/02/221612>

tanuki- 2022-06-07 やねうら王学習部リグレーション調査 - nodchip のブログ

<https://nodchip.hatenablog.com/entry/2022/06/07/000000>



使用ソフト、参考文献等、その3

人間の棋譜を用いずに評価関数の学習に成功 | やねうら王 公式サイト

<https://yaneuraou.yaneu.com/2017/06/12/%E4%BA%BA%E9%96%93%E3%81%AE%E6%A3%8B%E8%AD%9C%E3%82%92%E7%94%A8%E3%81%84%E3%81%9A%E3%81%AB%E8%A9%95%E4%BE%A1%E9%96%A2%E6%95%B0%E3%81%AE%E5%AD%A6%E7%BF%92%E3%81%AB%E6%88%90%E5%8A%9F/>

lambda 混合絞りについて | やねうら王 公式サイト

<https://yaneuraou.yaneu.com/2017/08/21/lambda%E6%B7%B7%E5%90%88%E7%B5%9E%E3%82%8A%E3%81%AB%E3%81%A4%E3%81%84%E3%81%A6/>

テラショック定跡の生成手法 | やねうら王 公式サイト

<https://yaneuraou.yaneu.com/2019/04/19/tera-shock-book-generation/>

ShogiGUI

<http://shogigui.siganus.com/>

ai5/BookConv

<https://github.com/ai5/BookConv>



使用ソフト、参考文献等、その4

ak110/Blunder.Converter: 棋譜変換ツール。
<https://github.com/ak110/Blunder.Converter>

各種将棋ソフト間での教師データの変換ツールの開発 - コンピュータ将棋 Qhapaq
<https://qhapaq.hatenablog.com/entry/2017/12/25/002820>

たややん /ToSfenpack20210122
https://twitter.com/tayayan_ts/status/1338443561272950787?s=20

将棋 AI の進捗 その 31(cuDNN による SENet の推論処理の実装)
- TadaoYamaoka の開発日記
<https://tadaoyamaoka.hatenablog.com/entry/2019/07/24/011150>

強い将棋ソフトの創りかた | マイナビブックス
<https://book.mynavi.jp/ec/products/detail/id=126887>

ResNet の改良モデル WideResNet を詳細解説！ | DeepSquare
<https://deepsquare.jp/2021/08/wideresnet/>