

Crazy Shogi is an implementation of the AlphaZero algorithm, as published by DeepMind. It is entirely original code developed from scratch, in C++. The neural network code uses the cuDNN library to run on Nvidia GPUs.

Crazy Shogi is part of the generic Crazy Zero system. Crazy Zero can also play chess, go, gomoku, renju, Othello, ataxx, breakthrough, kalah, twixt, and ultimate tic-tac-toe.

In 2020, the neural network of Crazy Shogi will be trained thanks to computing power provided by the NVIDIA Inception Program.

ひまわり アピール文書 (第31回世界コンピュータ将棋選手権)

前回までのプログラムは一切使用せず、Rustで一から書き直しています。  
今年は動くことを目標に細々と開発します。

データ構造は以前のC++のものと変わらないですが、評価関数はニューラルネットワークに変更したいと考えています。

以上

## 第29回世界コンピュータ将棋選手権 アピール文章

オッズの魔法使い 2021 - Random Forest Continues

written by: David Wada



1. parallel best first search continues
2. evaluation function - either bonanza or apery
3. deeper search per node. it was too shallow last time
4. opening book - Qhapaq定跡 (二代目)

<http://qhapaq.hatenablog.com/entry/2016/12/14/222645>

much appreciation to the creator!

4. using AWS ec2. probably about 4 machines



2021-03-31 - created

### うさびょん3 アピール文書

元々「うさびょん2」系列は、「なのはmini」（StockFishを将棋化したもの）をベースに、StockFishの最新のコミットを取り込み、KPPT型の評価関数を用いていました。

今回の「うさびょん3」は、「なのはmini」ベースではなく、StockFishを独自に将棋化しようとしています。

（「れさびょん」のコードを使って将棋化する感じです。

結果、ものとしては、「なのはmini」によく似たものになりそうです。）

また、評価関数については、独自のものを作ろうとしています。

（こちらは、うさびょん2に独自の評価関数を入れて実験しています。）

4駒関係は、以前Aperyの平岡さんが挑戦してみた結果、遅くて

採用を取りやめた経緯がありますが、私も追試してみました。

やはり、遅すぎてそのままでは使い物にならないようです…。

（何か速度的に工夫できる点が見つかれば良いのですが）

探索速度だけを考えると、2駒関係+ $\alpha$ 等が高速で良い感じなのですが、

2駒関係だけですと、今実験している範囲では伸び悩んでいます。

+ $\alpha$ の部分に何を入れれば良さそうなのか、悩んでいます。

（技巧の真似をするのがお手軽かも知れませんが）

非線形の評価関数には以前から興味はあり、高速に差分計算出来るようなものを考えようとしています。

(NNUEに似たような実装になってしまいそうですが)

というわけで、まだまだ実装も含めてこれから、という感じです。

何故か2月から仕事が滅茶苦茶に忙しい状態に陥っており、土日にも将棋のプログラムの方は進められていない（仕事してたりする）状況で、選手権直前まで頑張ってみますが、突然のキャンセルがあり得る状態です。

…アピール文書ではなくて、近況報告のようになってしまいましたw

## Deepさわにゃんアピール文書

2021/03/31 宇賀神

諸所の事情により、参加を取りやめていましたがまた参加することにしました。

前回参加した25回のときのプログラムは使用せず1から作り直しました。

プログラムのアピールポイントとしては下記のとおりです。

- オープンソースで開発
- ほぼフルスクラッチ
- Bitboardを使用
- 深層学習を使用（Pytorchを使用しています。）
- モンテカルロ木探索を使用

以上

# ツバメ将棋 アピール文書

田中 智, 椎野 良介, 小国 英明  
大野 弘喜, 有賀 才貴

2021. 3. 31

# 使用ライブラリ

- やねうら王 コンピューター将棋フレームワーク
  - 定跡生成、評価関数、探索部
- 使用するかもしれないライブラリ
  - [Kristallweizen](#)評価関数
  - [Apery](#)



# マシン情報

- CPU: Ryzen 9 5950X
- GPU: Geforce RTX 3080
- 総コア数: 16
- メモリ容量: 64GB
- OS: Windows 10 Pro 64ビット
- ソケット数: 1

# 学習

- 使用するライブラリ等
  - やねうら王
  - 学習時に使用する可能性がある
    - [Kristallweizen評価関数](#)
    - [Apery](#)
    - やねうら王によって [公開されている教師データ](#)

# 評価関数

- NNUE

# アピールポイント: 探索部

- 先読み: 数手先から先読みを行う
  - 数手: 5手程度を検討中
  - 数手先: 一番評価値の高い手を互いに選択すると仮定して進める
  - メリット: 数手先の評価値をより正確に決定可能
  - デメリット: 先読みした局面に辿り着かないことがある

## アピールポイント: 学習部

- 学習時のパラメータを変更する予定