

# Ryfamate アピール文書

Ryfamate チーム<sup>1</sup> 水無瀬 香澄

協力 杉村 達也, 磯崎 元洋

## ■ 開発動機

数年前に病気で倒れ入退院を繰り返していたが、自力で外の世界を歩けるまでに回復した時には世の中は変わり、機械学習の世界も大きく進歩していた。

そこで、機械学習の最近の動向を勉強するため、本大会に参加した。

## ■ 開発過程

近年、コンピュータ将棋の世界では、標準的な HalfKP 型 NNUE の評価関数(以下、NNUE)の成長が限定的となっている一方、Deep Learning 系の評価関数(以下、DL)<sup>2</sup>が目覚ましい成長を遂げている。

本来であれば、DL に関する文献やソースコードを読み込み理解したいところだが、初心者であり寝込んでいることが多い筆者には準備期間が短すぎた。また、寝込んでいる間に、手元の PC で教師局面の生成と学習を行わせていたが、特に教師局面の数を用意することが難しく過学習 (overfit) を起こすことなどから、十分な成果が得られなかった。

一方で、手元の PC で様々な計測を行う中で、DL は NNUE に比べて勝率は低いが序盤で有利になる頻度が高いことや、棋力が伯仲している NNUE の中でも、序盤が得意な評価関数と、終盤、特に入玉系の得意な評価関数があることが分かった。加えて、自分の作成した評価関数も、特定の条件下では既存のものより僅かに強い可能性があることが分かった。

これらの経緯により、それぞれの評価関数の長所を活かすべく、合議制を採用した。

## ■ 実験結果

まず、手元の PC で DL と NNUE を対局させると NNUE が大幅に強く、2021 年 1 月時点で公開されている評価関数を用いた場合、DL 側に思考時間を 2.5 倍程度与えることで勝率が釣り合うことが確認できた (Table (1)-(2))。これは、DL と NNUE のエンジンを単純に手数で切り替えることの難しさを示している。過去に、序盤に強い NNUE と終盤に強い NNUE を手数で切り替える実験結果はあるが<sup>3</sup>、今回の実験では同じ持ち時間では DL が弱いため、DL と NNUE を組み合わせて有意に強くなる条件 (手数) は見つけられなかった (Table (3)-(5))。

---

<sup>1</sup> <https://twitter.com/komafont>

<sup>2</sup> 話の単純化のため、本文では NNUE はやねうら王 (YaneuraOu) による alpha-beta pruning search で用い、DL は DeepLearningShogi (dlshogi) による Monte Carlo tree search で用いることを前提とする。

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

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

<sup>3</sup> <http://qhapaq.hatenablog.com/entry/2019/07/23/221409>

次に、単純な多数決について検討した。2つの異なる NNUE に4スレッドずつと1つの DL に2スレッドを割り当てた多数決エンジンを、4スレッドの NNUE 単独エンジンと対局させた場合、多数決エンジンが僅かながら有意に強いことを確認した(Table (6)-(8))。

もちろんこれは使用したスレッド数に  $10(=4+4+2)$  と 4 の違いがありフェアな比較ではないが、多数決エンジンの2つの NNUE は単独エンジンと比べて NPS が2割ほど低下しており<sup>4</sup>、多数決の有効性が確認できた。強いソフトや評価関数同士の多数決は前例があるが、同じ条件(時間)のもとでは弱い評価関数を多数決に加えても効果が見られたことに、本実験の意義があると考えた。

その上で、最初に思考するプロセスを DL と NNUE それぞれ1つずつとし、最善手が異なる場合のみ第3のプロセスに意見を聞くことを原則とするエンジンを開発した。定跡やその定跡に特化するように学習した評価関数を用いて計測したところ、序盤と最終盤を除き6-7割の局面で、DL と NNUE の最善手が一致することを確認した。これらの結果をもとに、プロセスの優先順位や時間管理など各種調整を行い、今回の出場に至った。

なお、本文では公開された探索部と評価関数を用いた実験結果を掲載しているため、これらについては第三者による追試が可能である。手数による切り替えや単純な多数決については、tttakさん公開の合議将棋(GougiShogi)<sup>5</sup>を用いて検証できる。NNUE 同士の自己対局に比べて環境依存性が高く、CPU と GPU のバランスはもちろん、バックグラウンドで動く他のプログラムによる影響も受けやすいため注意が必要である。<sup>6</sup>

(Table)	Player 1				Player 2				Result			
	Initial	Eval	Threads	Time	Eval	Threads	Time	Games	Win	Lose	Draw	WinRatio
(1) Hirate	GCT		4	0.6s	S3K	32	0.2s	256	143	109	4	56.7%
(2) fg20_r4000_agr16	GCT		4	1.0s	S3K	32	0.4s	288	139	141	8	49.6%
(3) fg20_r4000_agr16	GCT/S3K (ply: 32)		4/32/32	1.0s	S3K	32	1.0s	160	68	79	13	46.3%
(4) Hirate	GCT/S3K/BB (ply: 24,60)		4/32/32	1.0s	S3K	32	1.0s	138	49	63	26	43.8%
(5) fg20_r4000_agr16	GCT/S3K/BB (ply: 32,72)		4/32/32	1.0s	S3K	32	1.0s	148	58	83	7	41.1%
(6) fg20_r4000_agr16	GCT/S3K/BB (voting)		2+4+4	1.0s	S3K	4	1.0s	320	161	138	21	53.8%
(7) Hirate	GCT/S3K/BB (voting)		2+4+4	1m+1sf	S3K	4	1m+1sf	256	126	107	23	54.1%
(8) Hirate	GCT/S3K/BB (voting)		2+4+4	1m+2sf	S3K	4	1m+2sf	103	54	38	11	58.7%

fg20\_r4000\_agr16: floodgate 2020/Jan/01-2021/Jan/31, Rate>=4000, ply=16, Aigakari Opening  
GCT: GCT電竜(model-0000167), S3K: 水匠3改 (Suisho3Kai), BB: BURNING BRIDGES (BB\_denryu-tsec1)  
CPU: Ryzen Threadripper 2950X (15.7Mnps at 32 threads)  
GPU: GeForce RTX 3090 (47Knps at 4 threads)

<sup>4</sup> 稼働コア数の増加に伴うクロック低下と、キャッシュメモリのヒット率低下によるものと思われる。

<sup>5</sup> <https://github.com/tttak/GougiShogi>

<sup>6</sup> 余談だが、計測中に動画を再生しただけでも計測結果に影響する。

間違っても対局中の PC で、盛り上がっている Zoom 対局室に参加してはならない。