

水匠アピール文書（2）

令和元年5月13日
参加者 杉村達也

第1 開発動機

今大会での目的は、3月21日付「水匠アピール文書」第3に記載した工夫（教師局面の事後的変更）によって、NNUE 評価関数を有意に強くすることを目標としており、開発動機もそれに一致します。

第2 独自工夫の詳細

1 前後2手の評価値を見た評価値の修正

n 手目の評価値を $Score(n)$ としたとき、①n 手目の手番側が負けており、かつ② $Score(n) < Score(n-2) < Score(n+2)$ and $Score(n-1) < Score(n+1)$ となっている場合、 $Score(n)$ の値は間違っている（事後に反省をしている）可能性が高いと考えられるため、 $Score(n)$ を $Score(n-2)$ の値まで引き上げるという修正をしました。

2 勝敗と評価値のずれの排除

勝利側の手番の評価値であるにも関わらず、負の評価値となっている場合、又は敗北側の手番の評価値であるにも関わらず正の評価値となっている場合、その局面の判断が間違っている可能性が高いため、教師局面から排除してしまうこととしました。その代わり、elmo 式学習の際の LAMBDA は概ね 1 としました（勝敗項を見ない学習、詳細は、やねうら王の learner.cpp を参照¹⁾）。

第3 実験結果（選手権後の事後的実験）

1 方法

- (1) 任意の NNUE 評価関数（以下の実験は、Qhapaq Research Lab (QRL) におけるトップレーティング付近の評価関数を使用しました。）を準備する²⁾。
- (2) 当該評価関数から、ランダムムーブを入れず、depth18 で作成した検証用局面 1（30 万局面、手数は 24 手目から記録）、及び初手から連続 10 手のランダムムーブを入れた、depth18 で作成した検証用局面 2（5 万局面）を準備する。
- (3) depth14 で作成した、教師局面約 4000 万局面を準備する。
- (4) 上記（1）乃至（3）の評価関数、検証用局面及び教師局面から、以下の 3 通りの学習方法を試行する（その他の条件は同じ（eta = 0.01 , batchsize 2000000 , nn_batch_size 2000 など。))。
 - ① 通常の elmo 式学習 (Lambda = 0.5)
 - ② 通常の elmo 式学習 (Lambda = 0.8)
 - ③ 本文書第 2 で記述した学習 (Lambda = 1)

2 結果

- (1) 学習前評価関数

¹ <https://github.com/yaneurao/YaneuraOu/tree/master/source/>

² 今回は、<https://www.qhapaq.org/shogi/> における illqha4 を使用しました。

検証用局面1 : hirate eval = 13 , test_cross_entropy_eval = 0.590526 ,
test_cross_entropy_win = 0.519564 , test_cross_entropy = 0.555045 , norm
= 1.53195e+08 , move accuracy = 37.7632%

検証用局面2 : hirate eval = 13 , test_cross_entropy_eval = 0.230496 ,
test_cross_entropy_win = 0.156125 , test_cross_entropy = 0.193311 , norm
= 1.4517e+08 , move accuracy = 35.9364%

(2) 通常の elmo 式学習 (Lambda = 0.5)

検証用局面1 : hirate eval = 14 , test_cross_entropy_eval = 0.592798 ,
test_cross_entropy_win = 0.506909 , test_cross_entropy = 0.549854 , norm
= 1.86824e+08 , move accuracy = 37.8687%

検証用局面2 : hirate eval = 14 , test_cross_entropy_eval = 0.229684 ,
test_cross_entropy_win = 0.142708 , test_cross_entropy = 0.186196 , norm
= 1.56518e+08 , move accuracy = 36.0255%

(3) 通常の elmo 式学習 (Lambda = 0.8)

検証用局面1 : hirate eval = 21 , test_cross_entropy_eval = 0.589921 ,
test_cross_entropy_win = 0.510048 , test_cross_entropy = 0.549985 , norm
= 1.76822e+08 , move accuracy = 38.0958%

検証用局面2 : hirate eval = 21 , test_cross_entropy_eval = 0.22914 ,
test_cross_entropy_win = 0.147632 , test_cross_entropy = 0.188386 , norm
= 1.52022e+08 , move accuracy = 36.1327%

(4) 本文書第2で記述した学習 (Lambda = 1)

検証用局面1 : hirate eval = 41 , test_cross_entropy_eval = 0.590012 ,
test_cross_entropy_win = 0.50995 , test_cross_entropy = 0.549981 , norm
= 1.78723e+08 , move accuracy = 38.1903%

検証用局面2 : hirate eval = 41 , test_cross_entropy_eval = 0.229326 ,
test_cross_entropy_win = 0.148904 , test_cross_entropy = 0.189115 , norm
= 1.48885e+08 , move accuracy = 36.4891%

3 まとめ

以上の実験結果から、本文書で提示した学習方法は、lambdaを1と指定している(elmo式を利用していない)にもかかわらず、elmo式と同様に交差エントロピーを下げることができ、かつmove accuracy(検証用局面との最善手の一致率)をelmo式よりも上昇させることができると考えられます(勝率の測定については、3月21日付アピール文書のとおり)。

4 追試について

本参加者の公開したデータに、教師局面変更のプログラム(Python)、水匠の評価関数、上記Depth14の4000万教師局面(シャッフル前)等を置いておきます³ので、ご興味がある方は、追試等いただくと幸いです。

以上

³ <https://1drv.ms/f/s!AjMACEoJwb5ngpY6NcU9qhdV5XS4UQ>