

Six dubious ways to estimate the difficulty of a chess puzzle

Jan Zyško

Motivation: chess training process

1. Play a tournament
2. Find all the mistakes in the games
3. Prepare training materials and exercises
4. Train on the materials and exercises
5. Repeat

How to perform point 3 without a coach?

- 1) Training materials - game diagnostics
- 2) Exercises: find puzzles with positions similar to the one analyzed in **character** and **difficulty**

our task here!



Mark Dvoretsky, the most famous chess coach in the world

Anatomy of a lichess puzzle

Exercise rating (based on how many people previously solved it correctly)

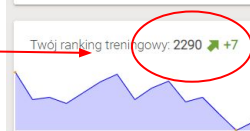


Zadanie 108024
Ranking: 2006
Rozwiązano 3262 razy

Z partii #JlsR56vf
10+0 • Rapid • Rankingowa

- radens514 (2137)
- ChessSpectral (2176)

Player rating (based on her previous performance)



Twój ranking treniingowy: 2290 ↗ +7

A line graph showing a fluctuating performance trend over time, with a red circle highlighting the current rating of 2290 and a green arrow indicating a +7 change.

The better the player is doing, the more difficult tasks they are assigned.



The screenshot shows a chessboard with a puzzle position. The board is oriented with a8 at the top and a1 at the bottom. The pieces are: a8: ♖, b8: ♜, c8: ♜, d8: ♜, e8: ♜, f8: ♜, g8: ♜, h8: ♚; a7: ♜, b7: ♜, c7: ♜, d7: ♜, e7: ♜, f7: ♜, g7: ♜, h7: ♜; a6: ♜, b6: ♜, c6: ♜, d6: ♜, e6: ♜, f6: ♜, g6: ♜, h6: ♜; a5: ♜, b5: ♜, c5: ♜, d5: ♜, e5: ♜, f5: ♜, g5: ♜, h5: ♜; a4: ♜, b4: ♜, c4: ♜, d4: ♜, e4: ♜, f4: ♜, g4: ♜, h4: ♜; a3: ♜, b3: ♜, c3: ♜, d3: ♜, e3: ♜, f3: ♜, g3: ♜, h3: ♜; a2: ♜, b2: ♜, c2: ♜, d2: ♜, e2: ♜, f2: ♜, g2: ♜, h2: ♜; a1: ♜, b1: ♜, c1: ♜, d1: ♜, e1: ♜, f1: ♜, g1: ♜, h1: ♜.

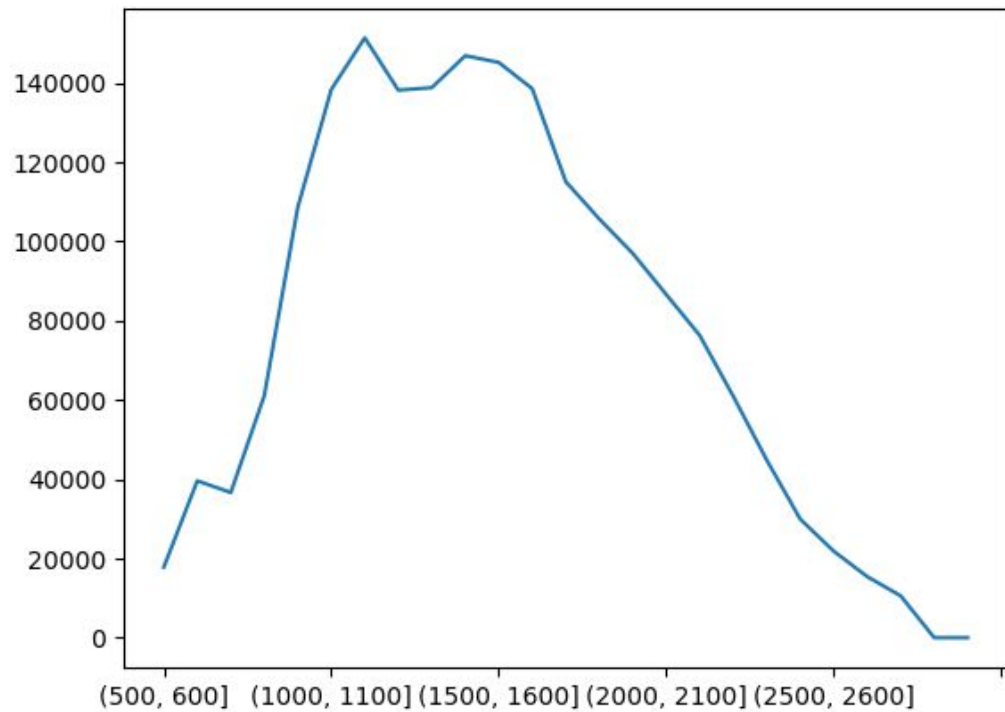
The move list on the right shows the following moves:

```
2 d4 @g7
3 Qc3 d6
4 @e3 c6
5 f3 @f6
6 Wd2 b5
7 O-O-O @bd7
8 @h6 @xh6
9 Wxh6 b4
10 @b1 Wa5
11 @c4 @a6
12 @xa6 Wxa6
13 We3 Wxa2
14 @e2 O-O
15 h4 e5
16 g4 We6
17 h5 a5
18 @g3 a4
19 Wh6 a3
20 @f5 gxf5
21 gxf5 @g4
22 Wg5+ @h8
23 fxe6 @h8
```

The success message at the bottom right says "Sukces" with a green checkmark and a score of 81. Below it is a blue button labeled "KONTYNUUJ TRENING".

At the bottom of the board, there is a row of numbers: +12, -23, +2, -5, +9, +9, +7, -22, +10, +6, -18, +4, -14, -14, +7.

Lichess puzzle dataset



Total: over 1.9 million puzzles

Benchmarks

Median prediction

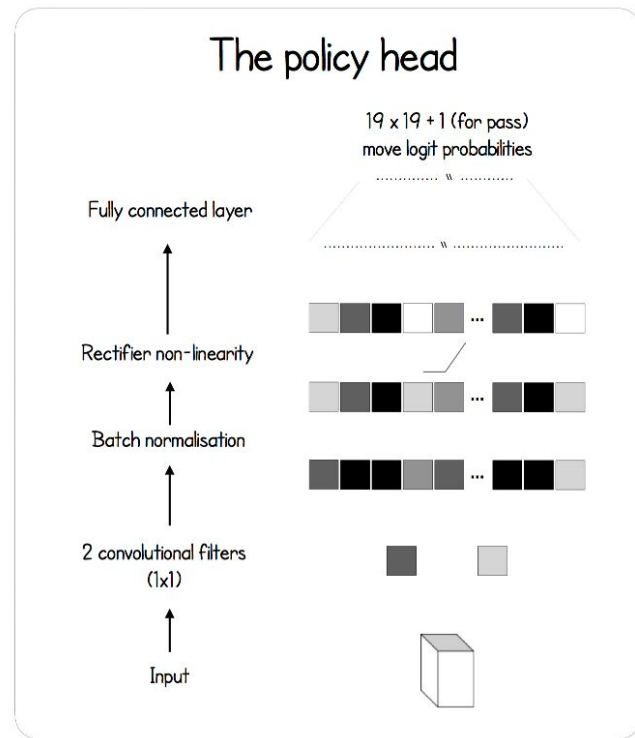
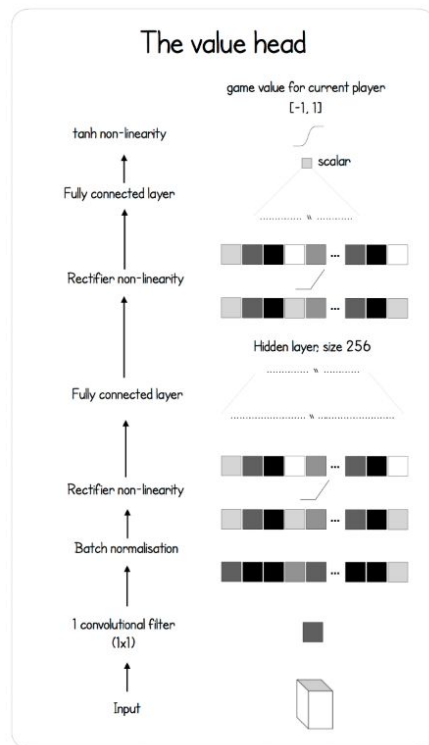
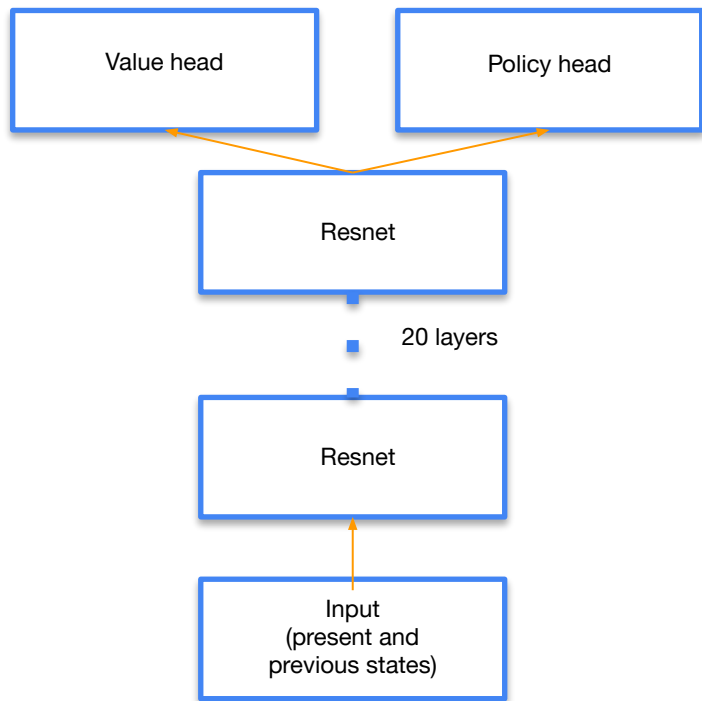
Results:
mean: 361
std: 256

Expert knowledge



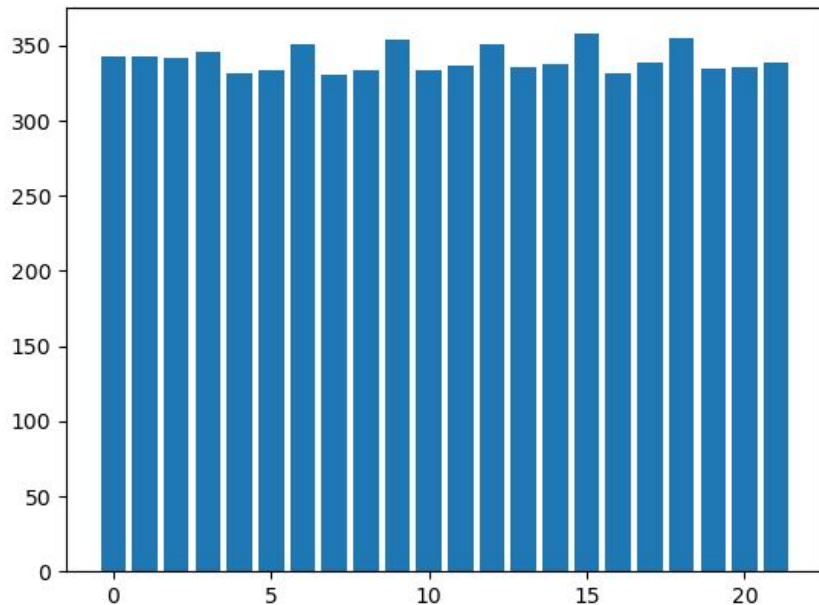
Results:
mean: 278
std: 227

Method 1: Use activations from an existing model (transfer learning)



Method 1: Use activations from an existing model

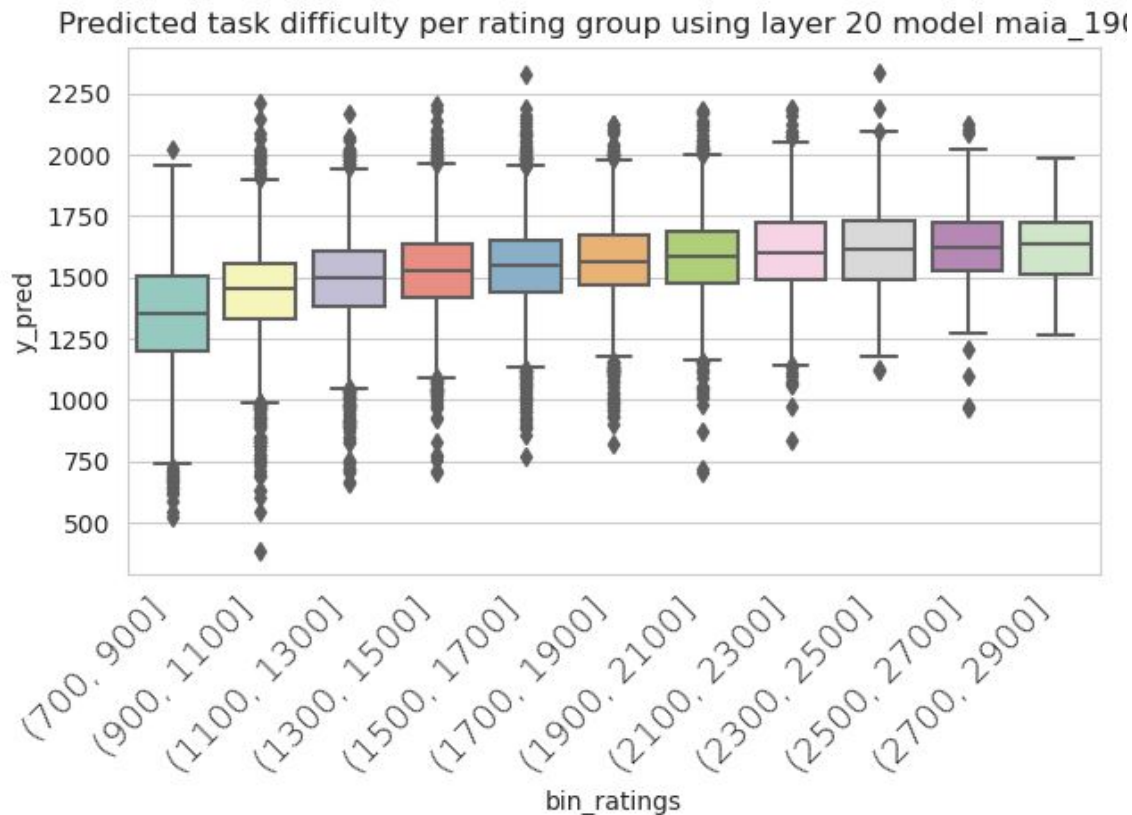
mean absolute
error



Best layer errors:
mean: 331
std: 243

activation layer
number used for
prediction

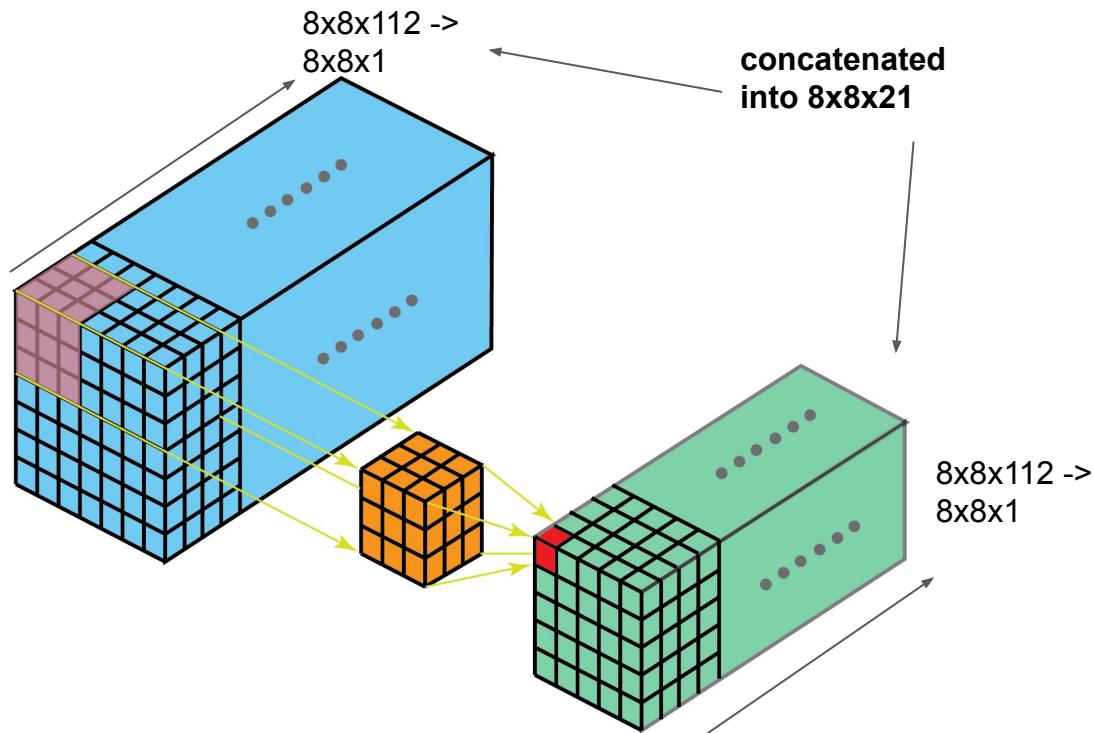
Method 1: Use activations from an existing model



Observations:

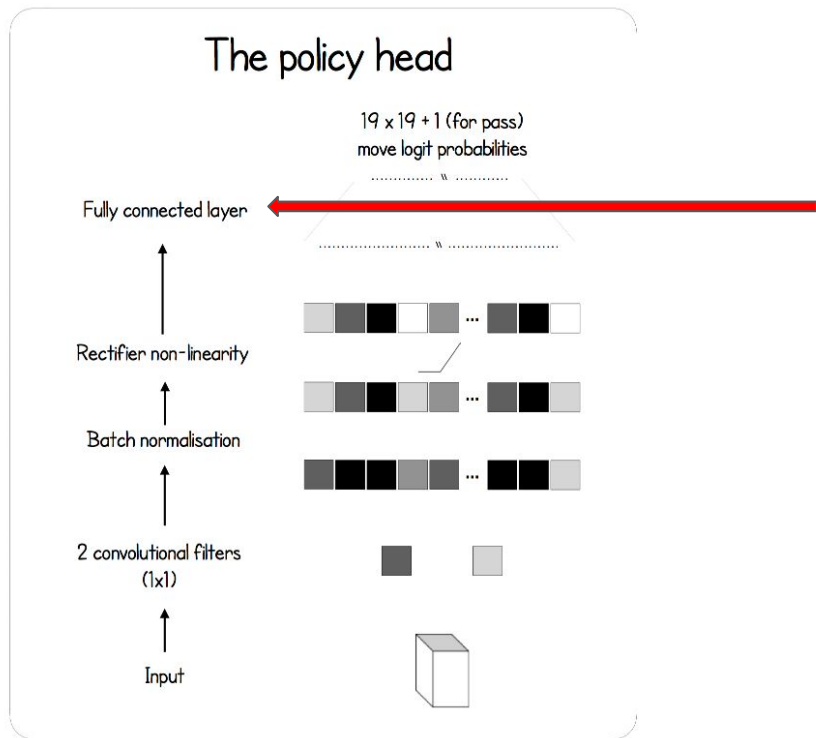
- Predicted ratings rise (on average) as the While predicted ratings rise, the variance is big and the rise

Method 2: Use all the activations, averaged by channel



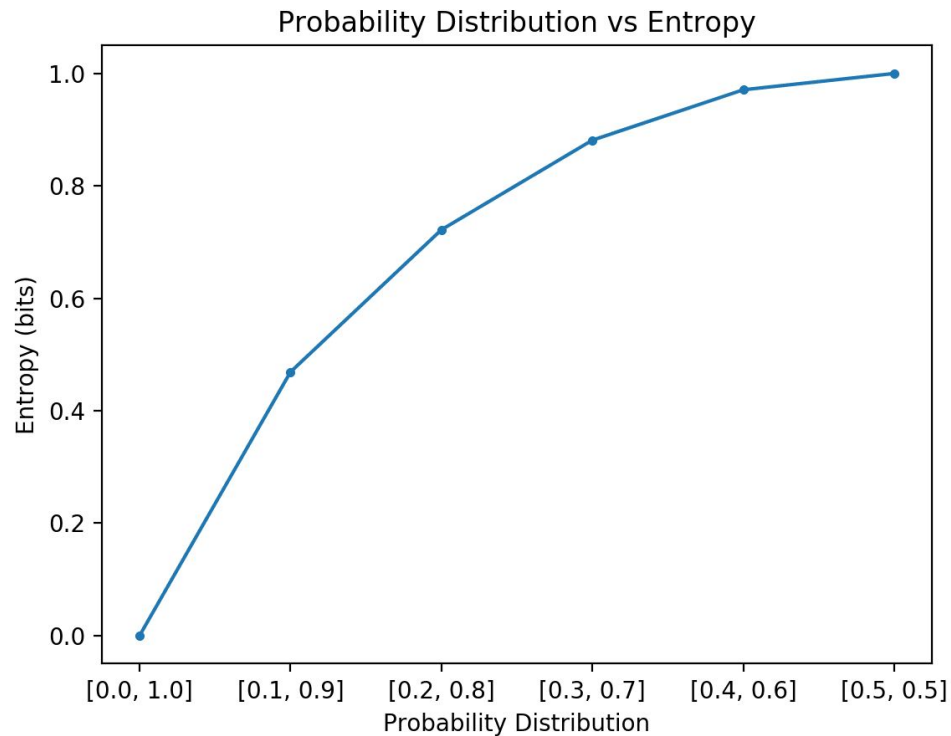
Results:
mean: 338
std: 246

Method 3: Use policy layer



Results:
mean: 350
std: 249

Method 4: Policy layer entropy

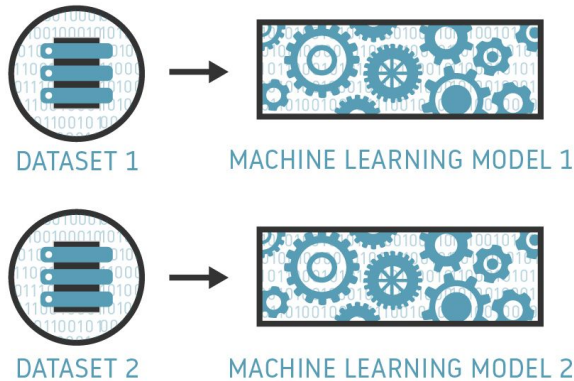


Results:
mean: 352
std: 251

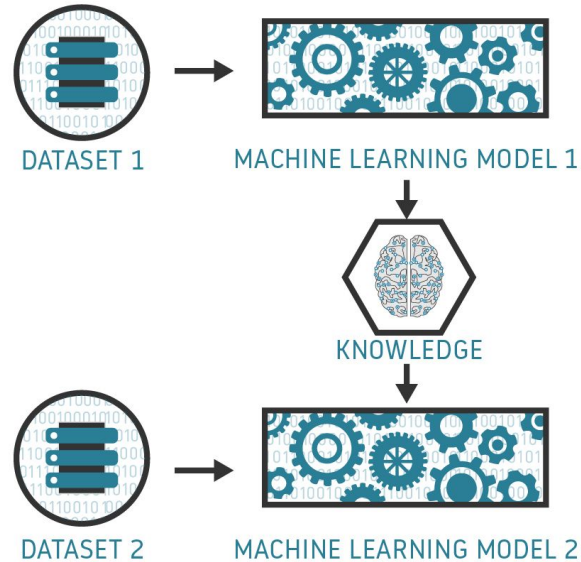
Pearson
correlation
-0.08

Method 5: Relearning weights for the rating prediction task

TRADITIONAL MACHINE LEARNING

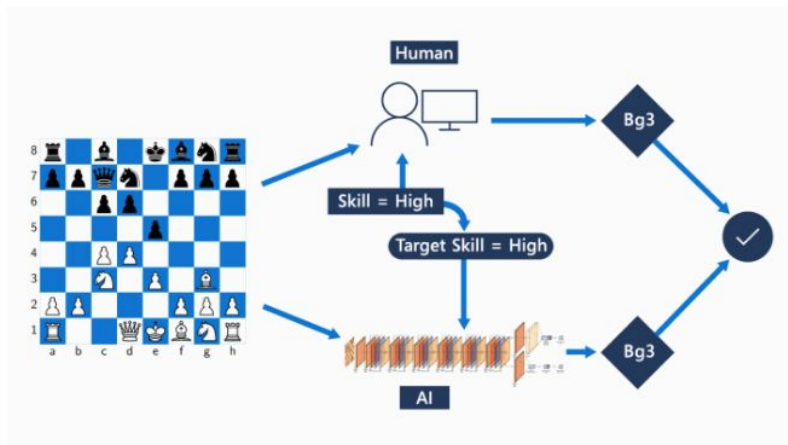


TRANSFER LEARNING

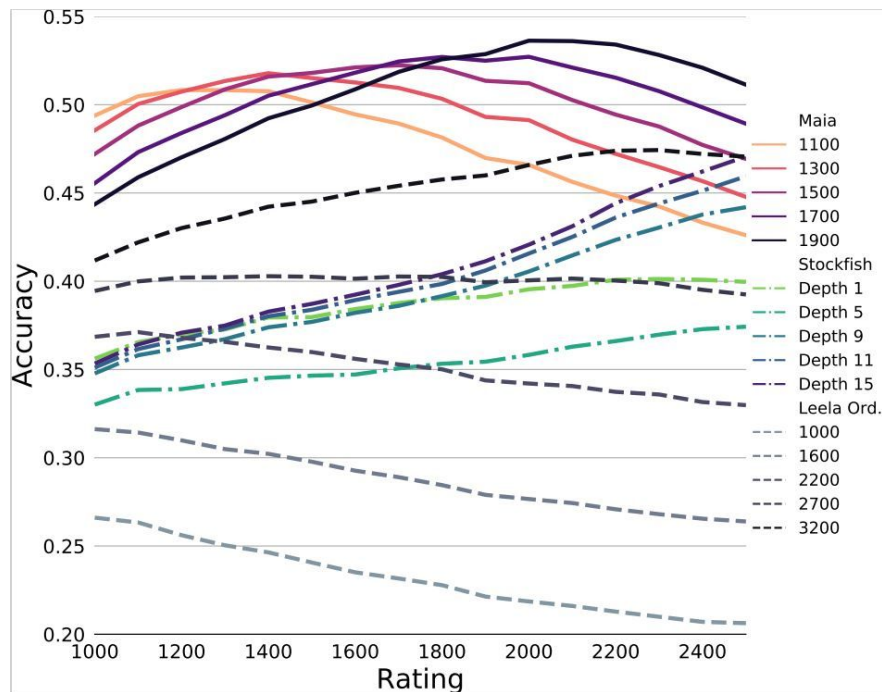


Results:
mean: 527
std: 396

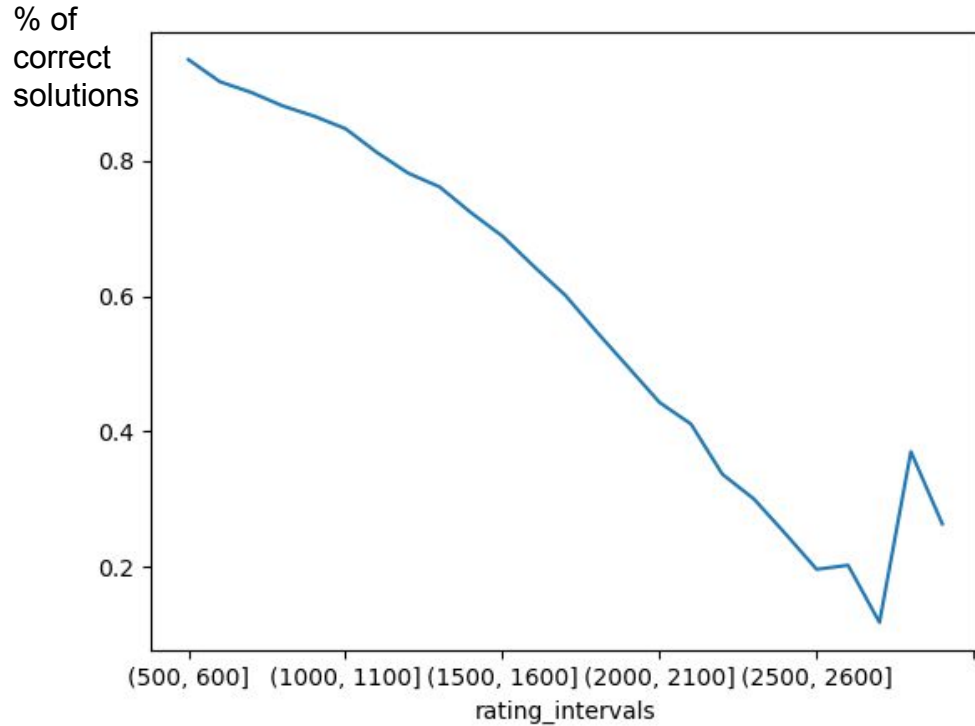
Method 6: Successful engine runs



Results:
mean: 275
std: 200



Method 6: Successful engine runs



Summary of the results

| | mean error | error std | train size | test size | model type |
|--|------------|-----------|------------|-----------|-------------------|
| <u>models</u> | | | | | |
| engine ensembling | 275 | 200 | 35000 | 5000 | XGBoost |
| transfer learning from activations | 331 | 243 | 60000 | 20000 | XGBoost |
| ensembling activations from different layers | 338 | 246 | 60000 | 20000 | XGBoost |
| relearning weights for prediction task | 527 | 396 | 75000 | 5000 | DNN |
| transfer learning from policy layer | 350 | 249 | 60000 | 20000 | XGBoost |
| policy entropy | 352 | 251 | 90000 | 10000 | Linear Regression |
| <u>benchmarks</u> | | | | | |
| median benchmark | 361 | 256 | 0 | 100000 | mean |
| aimchess | 278 | 227 | 0 | 1000 | ensemble model |

Future work

- Using more data and bigger models
- Ensembling more engines
- Analyzing MCTS tree:
 - evolution of the predicted best move
 - distribution of Q in the tree

