Chess dataset: <http://www.computerchess.org.uk/ccrl/4040/>

1,577,087 games of top engines.

Filename: CCRL-4040.[1577087].pgn

Using the file "create\_games\_fen.py", extract 1,000,000 positions from 1,000,000 different games and save them in the file "fens.txt". Time: 6,322 seconds (~1 hour 45 minutes).

Copy the "fens.txt" file to: C:\Users\User\Documents\tidhar\barilan\DeepLearning\thesis\chess\tucano\Tucano\data\train model

Run 'Tucano\_bpruning\_vs.exe', type command 'dataset'. It reads the fens, then for each of them a game is played, and search begin. At max depth search of 8, and depth left to be search of 2, the position is stored with beta and a label – pruned or not. Time for 1,000,000 records: 3.5 hours. (same for clean+zw data).

Run create\_features, read fen\_labels and create feature vector of length 389 for every fen (15 minutes for 1M records ). Attach the 'beta' value and result is 390 length vector where:

|  |  |  |
| --- | --- | --- |
| Feature name | Length | Group |
| Side to move | 1 | Position centric (global) |
| Castling rights | 4 | Position centric (global) |
| Material configuration | 12 | Position centric (global) |
| Piece lists | 180 | Piece centric |
| Sliding piece mobilities | 64 | Piece centric |
| Attack and defend maps | 128 | Square centric |
| Beta | 1 | Search centric |

Use Neural Network model and train it using "nn.py" for 100 epochs. Each epoch run for about 30 seconds.

(for 500,000 examples):

Split data: 180 sec

Preprocessing: 40 sec

Tucano search method:

1. Try to find a move from book
2. Using multiple threads (in our experiment set to 1)
3. iterative\_deepening
4. Aspiration window (25, 100, 400)
5. Principal variation search
6. If not in pv, run zero-window search
7. Quiesce
8. Transposition table
9. Multiple extension and reduction for the depth (example: move puts opponent in check)
10. If fail-high return score
11. During zero-window search:
    1. Razoring
    2. Null move pruning
    3. Prob-cut

In clean version of Tucano search, most (96%) of nodes are pruned due to fail-high (at depth remaining 2). In clean version using zero-window, 73% of nodes pruned due to fail-high.

After 100 epochs:

Epoch [100/100], Stage [validation], Loss: 0.4162, Accuracy: 81.0188%, Time epoch: 6 seconds

Confusion matrix (rows - true value. columns - predicted value):

tensor([[43893., 9617.],

[10724., 42930.]])

Precision: tensor([0.8037, 0.8170])

Recall: tensor([0.8203, 0.8001])

