Recreating AlphaZero Chess Engine

Ting Liang, Mark Chang
{tngliang,mrkchang}@Stanford.edu
CS 230 – Deep Learning, Stanford University

Motivation

Recently, *Deepmind's AlphaZero* surpassed grandmasters in games of Go, Chess, and Shogi using only *self-play games* (hence the suffix Zero).
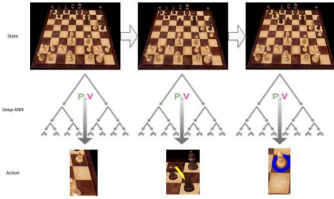
Goal

The goal of our project is to reproduce the AlphaZero chess engine and perform an engine match with the open source *Stockfish* engine, which is the current TCEC champion chess engine.
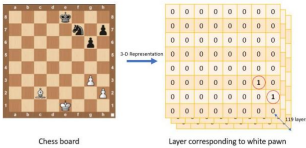
Data

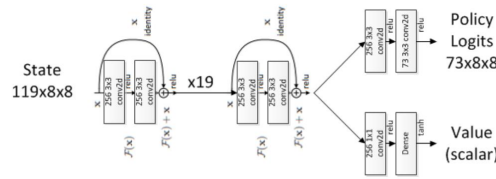Our dataset consists of 1,000,000 *Stockfish* vs. *Stockfish* self-play

Features

State: Chess board -> 119 x 8 x 8 state matrix
Move: UCI move -> 73 x 8 x 8 move matrix
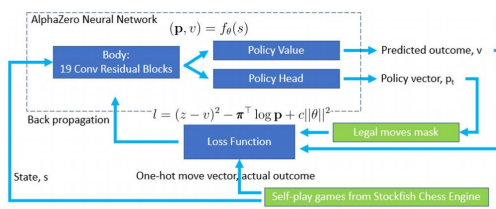Result: -1 (loss), 0 (draw), +1 (win)

Chess board   Layer corresponding to white pawn

Model

AlphaZero neural net:

State
119x8x8

Policy
Logits
73x8x8

Value
(scalar)

Loss function:

$$l = (z - v)^2 - \boldsymbol{\pi}^\top \log \mathbf{p} + c||\theta||^2$$

Training Details

AlphaZero Neural Network

$(\mathbf{p}, v) = f_\theta(s)$

Body:
19 Conv Residual Blocks

Policy Value
Policy Head

Predicted outcome, v
Policy vector, p_t

Back propagation $l = (z - v)^2 - \boldsymbol{\pi}^\top \log \mathbf{p} + c||\theta||^2$

Loss Function   Legal moves mask

State, s   One-hot move vector, actual outcome
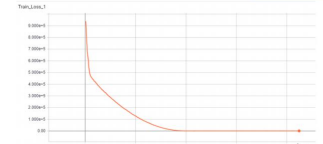
Self-play games from Stockfish Chess Engine

1. Memory Leak: .pgn -> .h5
2. Difficulty learning over 4672 legal actions
Solution: mask out illegal moves before calculating loss function
3. Batch size: 4096 -> 128
4. Weight Decay: 0.001 -> 100
5. Resnet: 19 -> 1 -> 5 -> 19 block

Tournament Evaluation

Reduced breadth (policy)   Reduced depth (value)
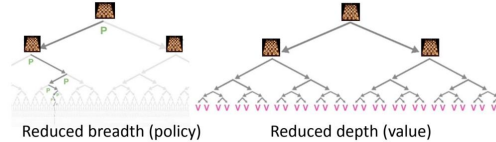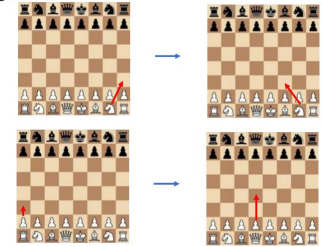
Results

We were able to train our model to mimic *stockfish* at a primitive level.
Loss: ~1 mil -> 700.
Checkpoints 5699 vs 0 (random): 0 loss, 8 draws, 6 wins for 5699
Best elo: 1052 (+50 from starting)

Discussion

Opening boards

Future

1. Training with supervised move probabilities, not 1-hot vectors
2. Larger dataset for larger variety games (resource challenges)
3. Filter moves like *DeepChess* (move count < 5, moves that capture, games ending in draws)
4. Apply *stockfish* rollout during evaluation for increased speed

References

[1] David, Omid E, Nathan S Netanyahu, and Lior Wolf (2016). "DeepChess: End-to-End Deep Neural Networkfor Automatic Learning in Chess".International Conference on Artificial Neural Networks. Springer, pp. 88–96.
[2] Linscott, Gary. "LCZero."lczero.org/.
[3] Fiekas, N. (2014).Chess:A pure Python chess library¶.Retrieved from https://python-chess.readthedocs.io/en/latest/5
[4] Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran,D., Graepel, T., et al. (2017a). Mastering Chess and Shogi by Self-Play with a General Reinforcement LearningAlgorithm. arXiv preprintarXiv:1712.01815.
[5] Yang, Daylen. "Stockfish Chess." Stockfish,stockfishchess.org/.