**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 games.

**Features**

State: Chess board \(119 \times 8 \times 8\) state matrix
Move: UCI move \(73 \times 8 \times 8\) move matrix
Result: -2 (loss), 0 (draw), +1 (win)

**Model**

AlphaZero neural net:

State

119x8x8

Policy Logits
73x8x8

Value (scalar)

Loss function:

\[ L = (z - v)^2 - \pi^T \log p + c||\theta||^2 \]

**Training Details**

- Memory Leak: \(pgn\) -> h5
- Difficulty learning over 4672 legal actions
- Solution: mask out illegal moves before calculating loss function
- Batch size: 4096 -> 128
- Weight Decay: 0.001 -> 100
- 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: \(\sim 1.5 \text{ mil} > 700\).
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**