Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm

Tony Allen and Gavin Glenn
Chess, Shogi, and Go

Chess
- Pieces have varying movement abilities, goal is to capture opponent’s king

Shogi
- Similar to chess, but larger board and action space

Go
- Much larger board, but all pieces have same placement rules

Board positions: $10^{50}$, $10^{120}$, $10^{170}$
### Properties of Chess, Shogi, Go

<table>
<thead>
<tr>
<th>Rules</th>
<th>Go</th>
<th>Chess and Shogi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translational Invariance</td>
<td>Yes</td>
<td>Partial (e.g. pawn moves, promotion)</td>
</tr>
<tr>
<td>Local</td>
<td>Yes (based on adjacency)</td>
<td>No (e.g. queen moves)</td>
</tr>
<tr>
<td>Symmetry</td>
<td>Yes (dihedral)</td>
<td>No (e.g. pawn moves, castling)</td>
</tr>
<tr>
<td>Action Space</td>
<td>Simple</td>
<td>Compound (move from/to)</td>
</tr>
<tr>
<td>Game Outcomes</td>
<td>Binary (look at prob. of winning)</td>
<td>Win / lose / draw (look at expected value)</td>
</tr>
</tbody>
</table>

Go seems more amenable to CNNs
Computer Chess and Shogi: Prior Approaches

Computer chess is the most studied domain in AI
- Highly specialized approaches have been successful
- Deep Blue defeated World Champion G. Kasparov in 1997
- Before AlphaGo, state-of-the-art was Stockfish

Shogi only recently achieved human world champion level
- Previous state-of-the-art was Elmo

Engines like Stockfish and Elmo are based on alpha beta search
- Domain specific evaluation functions tuned by grandmasters
AlphaZero, starting from first principles

AlphaZero assumes no domain specific knowledge other than rules of the game
- Compare with Stockfish and Elmo’s evaluation functions
- Previous version AlphaGo started by training against human games
  - It also exploited natural symmetries in Go both to augment data and regularize MCTS

Instead, AlphaZero relies solely on reinforcement learning
- Discovers tactics on its own
Agent receives information about its environment and learns to choose actions that will maximize some reward\(^1\)

Compare with supervised learning, but instead of learning from a label, we learn from a time delayed label called reward

- Learn good actions by trial and error

1. Deep Learning with Python, F. Chollet
Reinforcement Learning

This framework is better suited to games like GO

In theory a network could learn to play using supervised learning, taking recorded human Go games as input

- Large suitable datasets may not exist
- More importantly, the network will only learn to perform like a human expert, instead of learning true optimal strategy

RL can be done through self play

- Using current weights, play out an entire match against self
- Update weights according to results
**AlphaZero**

Deep Neural Network $f_\theta$ with parameters $\theta$

**Input:**
- Raw board representation, $s$

**Output:**
- Policy vector, $p$
- Expected value of board position, $v$

**Loss:**

$$Loss = (z - v)^2 - \pi^T \log(p) + c||\theta||^2$$

Where $\pi, z$ come from tree search and self-play
AlphaZero

Architecture?

- Not stated explicitly (implied to be same as AlphaGo Zero)
- 20 (or 40) residual blocks of convolutions
**Monte Carlo Tree Search**

**Goal:** Reduce depth and breadth of search
- Move distribution $p$ helps reduce breadth
- Value $v$ helps reduce depth

Not *exactly* what happens

It’s best to see a picture!

2017 NIPS Keynote by DeepMind’s David Silver
Monte Carlo Tree Search In a Picture

Training

Algorithm Pseudocode

Initialize $f_\theta$

Repeat:
  Play Game:
    While game not over:
      MCTS from state $s$
      Compute $\pi$
      Pick new $s$ from $\pi$
  Update $\theta$:
    For each $(s, \pi, z)$
    Back propagate

Many game series versus previous state-of-the-art

- Convincingly beat Stockfish, Elmo
- Slight improvement over AG0

Concluding Remarks

Public Criticism
● Unfair advantages over competition
● Hard to replicate

Future Work
● Hybrid approach
● AlphaStar (Starcraft 2 AI)
Thank You