LeelaChessZero
Open Source Community
(F. Huizinga)
Overview

- What is Lc0?
- The GameTree and A0 in a nutshell
- Contribute
- Useful links
- Technical details
What is Lc0?

- 2016 Deepmind’s AlphaGo
- 2017 AlphaZero
- 2017 LeelaZero
- 2018 LeelaChessZero
The Game Tree
Why care?

- General approach, no domain knowledge required (Go, Chess, Shogi, ...)
- Visual interpretation of the game allows for a deep positional - and materialistic understanding obtained from selfplay
- Fascinating gameplay, see youtube videos on alphazero/leelachesszero
LeelaChessZero

- Initially missing details on the neural network architecture
- Variable compute budget
- Obtain dedicated hardware for training
- Always looking for contributors
  - Developers
  - Computational help
  - Testers/Elo estimators
  - Enthusiasts
Links

- lczero.org
- testtraining.lczero.org
- github.com/LeelaChessZero
- discord.gg/pKujYxD
Thanks to

- DeepMind
- Gian-Carlo Pascutto
- Leela Developers
- Lc0 Developers
- Testers
- Chess enthusiasts
Minimax Algorithm

\[ \text{max}(-1, +1, -1) \]

\[ \text{min}(0, 0, -1) \]

\[ \text{min}(0, +1) \]
Evaluation Function

- Minimax unable to reach *terminal* nodes given time constraints
- Approximate minimax value of subtree
- Must evaluate *non-terminal* nodes
- Centuries of human chess understanding to properly define this function
Minimax + Eval

max(-3, 2, 0)

min(8, -3, 1)

min(2, 4)
AlphaZero

Main objective: Prune the gametree

Learn the evaluation function (value) and most promising moves (policy) of the gametree iteratively from selfplay data.
Neural Network

Expected outcome: 1

Move distribution

<table>
<thead>
<tr>
<th>X</th>
<th>0</th>
<th>θ</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>θ</td>
</tr>
</tbody>
</table>
Training Data

Game state

<table>
<thead>
<tr>
<th>X</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Expected outcome: 1

Result
Win  +1
Loss -1
Draw  0

Obtain data through selfplay

Policy

Move distribution

<table>
<thead>
<tr>
<th>X</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
(MCT) Search
(MCT) Search
(MCT) Search
(MCT) Search
(MCT) Search
Records of data

(State₁, Policy₁, Result₁)
(State₂, Policy₂, Result₂)
...
(Stateₙ, Policyₙ, Resultₙ)

Where n is the total moves in the game played.