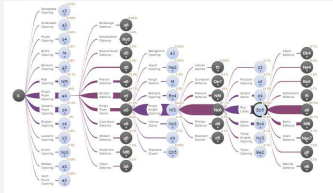


Chess with Deep Reinforcement Learning

1) The Problem

The effect of the size of the search space on the learning process of deep reinforcement learning methods



Hypothesis:

As the size of the search space increases, the number of training steps required to perform at a certain elo also increases

2) The Algorithm

Self-Play:

The best current player plays thousands of games with itself

Deep Neural Network:

The network learns from a blank state. Optimize the network weights every training loop

Monte Carlo Tree Search:

After many moves, select a move deterministically or stochastically

3) Tools

Dataset: 4462- Chess-problems

```
{  
  "problemId": 1,  
  "first": "White to Move",  
  "type": "Mate in One",  
  "fen": "Rn1k1/5pp/5q1/R/R/2B/5PPP/k1 w - - 0 1",  
  "moves": "f6-g7"  
}
```



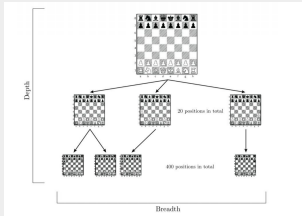
Deep Reinforcement Library: Arjan Groen's "RLC" (Reinforcement Learning Chess), Keras

Game Functionality: Python Chess library

Environment: Google Colab

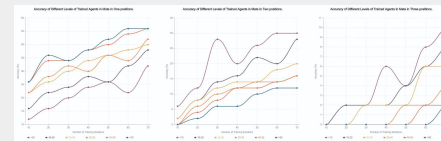
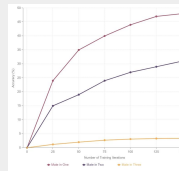
4) The Method

- 1- Have the same engine trained for different amount of steps
- 2- Have different chess positions where the size of the search space grows at different speeds



5) The Results

- 1- Depth is the independent variable. Breadth is constant
- 2- Depth is constant. Breadth is the independent variable



Contact

Aksel Hakim Zakuto
a.hakimzakuto@student.tudelft.nl

Supervisors:
Greg Neustroev, Mathijs de Weerd

