Deep Reinforcement Learning applied on the Chess Variant Crazyhouse

Deep Learning: Architectures and Methods - Project Proposal - 15th May 2018

TECHNISCHE UNIVERSITÄT DARMSTADT

Alena Beyer 2735580; Moritz Willig 2527455; Stefan Werner 2041986; Johannes Czech 2422372

1 Introduction

Google Deep Mind demonstrated the effectiveness of deep neural networks in the field of reinforcement learning by beating the world champion Lee Sedol in the game of Go [Silver et al. (2016)]. In 2017 the technique was extended to the games chess and shogi, and was then generalized by learning without any human domain knowledge. [Silver et al. (2017)]. These successes led to an increasing interest in Google's learning algorithm and is now used for problems in different domains [Wang and Rompf (2018)]. In this project we aim to apply deep reinforcement learning on the chess variant Crazyhouse.

2 Rules of Crazyhouse

Crazyhouse is a chess variant which is similar to normal chess. It incorporates all of the classical chess rules such as castling, En Passant and draw by 3 fold repetition. The main addition is the ability to reintroduce pieces that can be captured from the opponent. The captured piece switches its team to the opponent player's *pocket*. Instead of playing a classical move, the player has then the possibility to drop the piece back on an empty square of the board. Only pawns cannot be dropped first or eighth rank. The element of dropping captured pieces is similar to the game of shogi, with the difference that pieces can also be dropped to achieve immediate checkmate. Crazyhouse is similar to Bughouse chess, but played by two players instead of four [Droste and Fürnkranz (2008)].

Figure 1: Example of a playing position in Crazy House [Duplessis (2018)]

3 Motivation

There exists fewer chess engines for Crazyhouse then for classical chess and it's opening theory hasn't been explored as extensively as for classical chess. Furthermore, there is a broader search

space for each move due to the ability to drop pieces. We expect to have a stronger learning signal compared to original chess because game tend to be shorter and sub-optimal move can instantly be disputed by a strong player. The drawing percentage is lower than in chess because there's no draw due to insufficient material or stalemate. On the website `lichess.org` there exists an database of over 4 million downloadable human Crazyhouse games. These games can be used for supervised learning and the opening choices can be compared with the opening preferences of the neural network during it's training process [Duplessis (2018)].

4 Related Work

One open source project which is called *Leela Chess* aims to replicate the work of Deep-Mind on classical chess by using distributed computing of the community [Linscott (2018)]. The project uses Tensorflow as it's deep learning framework and will help accelerating the coding progress. The open source chess engine *Stockfish* has also been extended to the rule set of Crazyhouse and could be used as a reference evaluation of a playing position [Tord Romstad (2018)].

5 Project Goal

The main project goal is to train an neural network which is able to play at least on amateur level which corresponds to ≈1500 elo rating on `lichess.org` [Duplessis (2018)]. Our milestones of the project are the following:

1. Integrate all Crazyhouse's rule to the system
2. Train a deep neural network using either supervised or only reinforcement learning
3. Optimize the model architecture and learning procedure

One major challenge will be the availability of computing resources. Therefore we might have to use a more shallow network compared to the *Alpha Go Zero* version.

References

Droste, S. & Fürnkranz, J. (2008). *Learning of piece values for chess variants* (Tech. Rep.). Tech. Rep. TUDKE2008-07, Knowledge Engineering Group, TU Darmstadt.

Duplessis, T. (2018). *Chess analysis board.* Retrieved 2018-05-12, from `https://lichess.org/analysis`

Linscott, G. (2018, May). *leela-chess: A chess adaption of GCP's Leela Zero.* Retrieved 2018-05-12, from `https://github.com/glinscott/leela-chess` (original-date: 2017-12-30T23:23:25Z)

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., ... Hassabis, D. (2016, January). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489. Retrieved 2018-05-12, from `http://www.nature.com/articles/nature16961` doi: 10.1038/nature16961

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., ... Hassabis, D. (2017, December). Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning

Algorithm. *arXiv:1712.01815 [cs]*. Retrieved 2018-05-12, from http://arxiv.org/abs/1712.01815 (arXiv: 1712.01815)

Tord Romstad, J. K., Marco Costalba. (2018, May). *Stockfish: UCI chess engine.* official-stockfish. Retrieved 2018-05-12, from https://github.com/official-stockfish/Stockfish (original-date: 2014-06-18T19:59:53Z)

Wang, F. & Rompf, T. (2018, February). From Gameplay to Symbolic Reasoning: Learning SAT Solver Heuristics in the Style of Alpha(Go) Zero. *arXiv:1802.05340 [cs]*. Retrieved 2018-05-12, from http://arxiv.org/abs/1802.05340 (arXiv: 1802.05340)