Using deep reinforcement learning and self-play to master the game abalone

Ture Claußen, 1531067, ture.claussen@stud.hs-hannover.de

Hochschule Hannover Fakultät IV

Abstract. Perfect information games provide a great playing ground for the comparison of different algorithms, as they allow for direct comparison in a controlled environment. Whereas more classical approaches like minimax require human knowledge encoded in heuristics, modern approaches like (deep) reinforcement learning have excelled solely based on knowledge gained from self-play. To test how well this generalizes for other games, in the following, this is applied to the game abalone and compared with existing algorithms in a final tournament.

1 Introduction

Abalone is a fairly new game that was devised in 1987 by Michel Lalet and Laurent Lévi. Nevertheless, with more than four million global sales it has established itself as a classic game [4]. Abalone is a two-player game consisting of a hexagonal board with 61 fields and 14 marbles for black and white respectively. The abstract nature of the game requires the player to plan ahead and find the right strategy in the plethora of moves.

As it is a two-player perfect information game it can be solved, in theory, by backward induction. However, in search of the optimal move it is not possible to expand all of the possible paths the game could take, even for modern computers. Hence, more sophisticated approaches for navigating the search space are needed. The simple yet complex nature of this type of game makes it very suitable for theoretic analysis. [9, p. 1]

Ranging from game theory to artificial intelligence, games have sparked theoretical inquiry with many real world applications in economics, psychology, mathematics, neuroscience and many more. [15, pp. 46]

2 Background

The comparison between a human's and a computer's proficiency in a game is particularly interesting as it allows for very tangible conclusions about the current state of affairs. IBM's "Deep Blue" (DB) win against Gary Kasparov [11] in chess caused big public interest just as the win of "AlphaGo" (AG) against Lee Sedol [8] in the game Go. These two milestones also represent a shift in technology. Whereas Deep Blue relied on minimax, AlphaGo relied on a combination of multiple deep neural nets trained in self-play. [17]

Building on this success DeepMind, the company behind AG, further improved the architecture. "AlphaGo Zero" and the generalization "AlphaZero" (AZ) learn *tabula rasa*, without the help of human knowledge and surpassed the performance of AG significantly. Since then the architecture has been applied to Chess, Shogi and Atari games by removing the last piece of human knowledge in the system: The rules of the game. [16]

There is a formidable body of work regarding classical game-playing agents and some based on learning algorithms. The most significant contributions are:

1. "Algorithmic fun-abalone" (2002) Considers foundational heuristics for the game and analyzes minimax and its refinements in the form of (heuristic) alpha-beta pruning. Furthermore it sheds light on the performance differences between those. [5]
2. "Abalearn: Efficient Self-Play Learning of the game Abalone" (2003) Utilizes TD-Learning to create a self-learning agent that is on par with intermediate players. [6]
3. "A Simple Intelligent Agent for Playing Abalone Game: ABLA" (2004) Implementation of a game-playing agent with minimax, alpha-beta pruning and some custom heuristics. The evaluation of the performance is done by comparing the agent to existing software in the form of ABA-PRO and RandomSoft. [**?**]
4. "Constructing an abalone game-playing agent" (2005) Provides a very thorough explanation and analysis of the game's fundamentals, such as the state space, rules and positions. In regards to the alpha-beta pruning it also explains strategies for ordering the nodes and performance concerns. [13]
5. "Implementing a computer player for abalone using alpha-beta and monte-carlo search" (2009) This master thesis is a very exhaustive analysis of the game, alpha-beta pruning and Monte Carlo tree search, conferring many of the previous results. [7]

Considering the date of the publication, interest in the game has cooled off. However there are some smaller student publications of the more recent years that apply Q-Learning and TD-Learning. [14] [12]

3 Aims and objectives

The thesis has the objective to modify the architecture of AlphaZero (c.f. fig. 1 & 2) such that it can be applied to the game of Abalone. As the computational resources are not as vast as those of Google, it has to be evaluated how the architecture works on smaller networks with less training duration.

In detail the steps to achieving that objective are to:

- adapt the architecture of AlphaZero and to implement it,
- build a self-training pipeline,
- answer the question: *How well does the architecture of AlphaZero generalize?*

4 Methodology

A key element for the success of AlphaZero might be advancement in computational resources, wherefore this project requires sufficiently powerful hardware to adequately replicate the conditions. For smaller scale testing in the initial phase there is the possibility to use a personal workstation with a GTX 1660S and 6 core processor. Moreover, the Hochschule Hannover might be able to provide a more powerful workstation. For the larger scale and final experiments an application for the usage of Google's TPU Research Cloud has been successfully granted. It allows for the usage of "up to 5 on-demand Cloud TPU v3 devices, 5 on-demand Cloud TPU v2 devices, and 100 preemptible Cloud TPU v2 devices for free for 30 days" [2]. This can be combined with the free $300 credit for new Google cloud accounts. [1]

To assess the performance of the trained algorithm (variants) it will be compared with other algorithms in a tournament. All of the algorithms devised in related papers are not freely accessible wherefore access has been requested via email. As baseline there are the implementations created in a previous project [3] which would need further improvement to be more competitive. Namely that would be a parallelization of the Minimax algorithm by switching to principal variation splitting or an alternative that preserves pruning. [10]

5 Timeline

Week	Item
[-2, 1)	Foundational courses on usage of tensorflow and notation in RL
[1, 2]	Reading and gathering of related material
[3 - 4]	Planning of software, including research on more suitable game engine
[5]	Implementation of first design and small tests
[6]	Iteration over first design and preparation for distribution of training across multiple TPUs tests
[7]	Preparation of final experiment tests
[8 - 9]	Training of large model and tournament tests
[10-12]	Final writing, printing and submission

6 Preliminary table of contents

1. Introduction
2. Related work
3. Architecture
   (a) Structure of neural network
   (b) Self-learning mechanism
   (c) MCTS search
4. Empirical performance evaluation
   (a) Game engine
   (b) Experimental setup
   (c) Results
5. Conclusion

A AlphaZero architecture

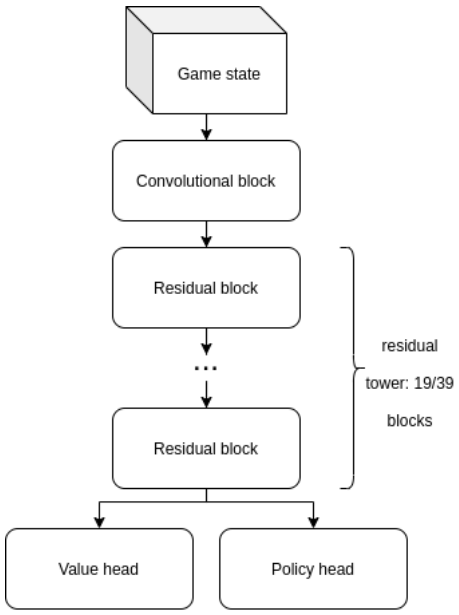
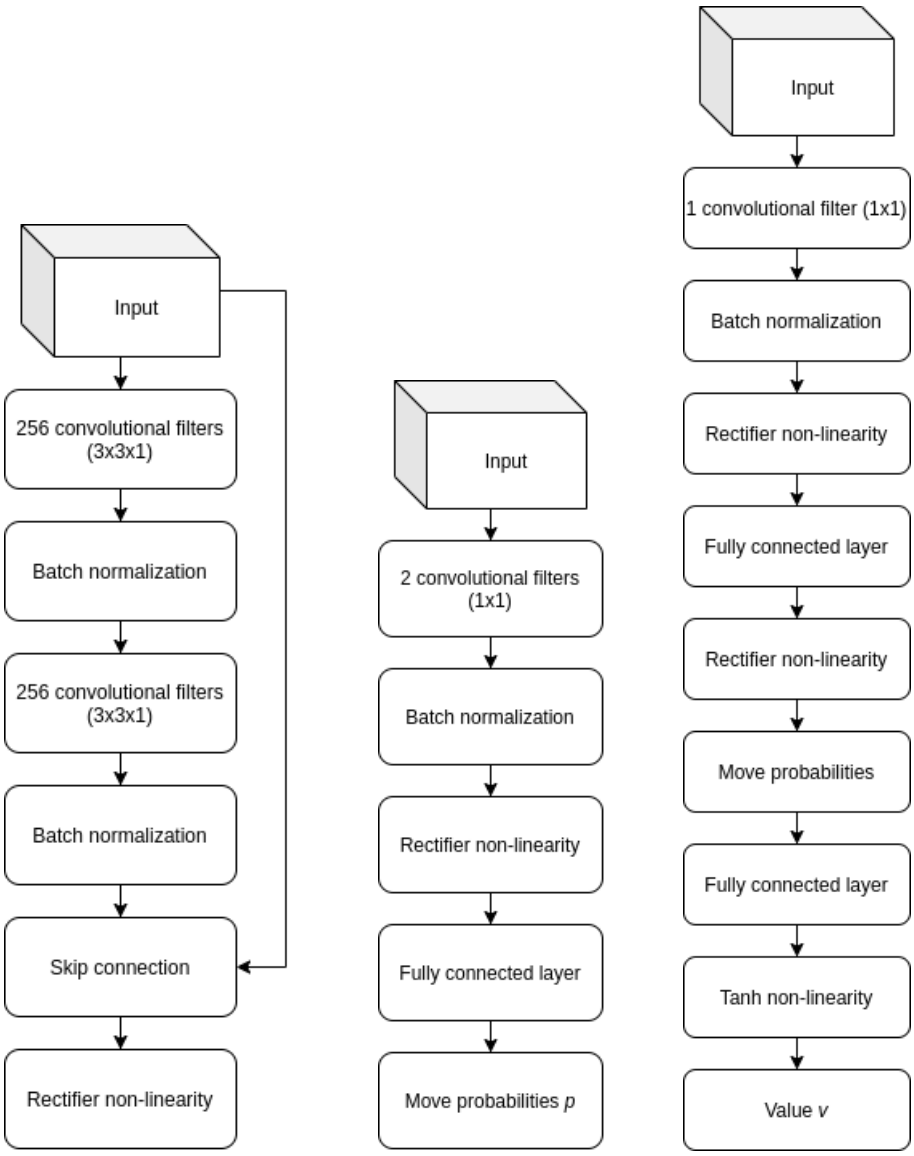


Fig. 1: The coarse architecture of AlphaZero [17]



(a) Residual block (b) Policy head (c) Value head

Fig. 2: The different components of AlphaZero in detail [17]

References

1. Free Trial and Free Tier. https://cloud.google.com/free
2. TPU Research Cloud. https://sites.research.google/trc/

3. Ture / abalone. https://gitlab.com/CampFireMan/abalone
4. Abalone (board game). https://en.wikipedia.org/w/index.php?title=Abalone_(board_game)&oldid=994557581 (Dec 2020)
5. Aichholzer, O., Aurenhammer, F., Werner, T.: Algorithmic fun-abalone. Special Issue on Foundations of Information Processing of TELEMATIK **1**, 4–6 (2002)
6. Campos, P., Langlois, T.: Abalearn: Ecient Self-Play Learning of the game Abalone (2003)
7. Chorus, P.: Implementing a Computer Player for Abalone Using Alpha-Beta and Monte-Carlo Search. Master's thesis, Citeseer (2009)
8. DeepMind: Match 1 - Google DeepMind Challenge Match: Lee Sedol vs AlphaGo. https://www.youtube.com/watch?v=vFr3K2DORc8&t=7020s
9. Demichelis, S., Ritzberger, K., Swinkels, J.M.: The simple geometry of perfect information games. International Journal of Game Theory **32**(3), 315–338 (Jun 2004). https://doi.org/10.1007/s001820400169
10. Gao, Y., Marsland, T.A.: Multithreaded Pruned Tree Search in Distributed Systems p. 11
11. Higgins, C.: A Brief History of Deep Blue, IBM's Chess Computer | Mental Floss. https://web.archive.org/web/20170803130439/https://www.mentalfloss.com/article/503178/brief-history-deep-blue-ibms-chess-computer (Jul 2017)
12. Lee, B., Noh, H.J.: Abalone –Final Project Report. Tech. rep. (2005)
13. Lemmens, N.: Constructing an abalone game-playing agent. In: Bachelor Conference Knowledge Engineering, Universiteit Maastricht. Citeseer (2005)
14. Mizrachi, R., Golran, G., Jacobi, O., Zats, R.: Introduction to artificial intelligence Final Project. Tech. rep., The Hebrew University of Jerusalem (2017)
15. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education, Inc, fourth edn. (2021)
16. Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T., Silver, D.: Mastering Atari, Go, chess and shogi by planning with a learned model. Nature **588**(7839), 604–609 (Dec 2020). https://doi.org/10.1038/s41586-020-03051-4
17. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., Hassabis, D.: Mastering the game of Go without human knowledge. Nature **550**(7676), 354–359 (Oct 2017). https://doi.org/10.1038/nature24270