HOCHSCHULE
HANNOVER
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS
–
Fakultät IV
Wirtschaft und
Informatik

Mastering the game of Abalone using deep reinforcement-learning and self-play

Ture Claußen

Bachelor thesis in "Applied computer science"

November 17, 2021

Author	Ture Claußen Matriculation number: 1531067 tu.cl@pm.me
First examiner:	Prof. Dr. Adrian Pigors Abteilung Informatik, Fakultät IV Hochschule Hannover adrian.pigors@hs-hannover.de
Second examiner:	Prof. Dr. Vorname Name Abteilung Informatik, Fakultät IV Hochschule Hannover email-Adresse

Declaration of authorship

I hereby declare that I have written this thesis independently without any help from others and without the use of documents or aids other than those stated. I have mentioned all used sources and cited them correctly according to established academic citation rules.

Hannover, November 17, 2021 Signature

Contents

List of Figures

List of Tables

Abstract

Explanation

1 Introduction

In the field of computer science board games have been a popular environment to test the capabilities of state of the art methods against human opponents. Many board games are widely known making them a tangible measure of performance. The most prominent examples are the games of Chess and Go. For both the defeat of the current best players by machines have been representative of fundamental progress in computing.

IBM's "Deep Blue" defeated Gary Kasparov in 1996 [Hig17] by utilizing search to look ahead into the game tree and choose the move that maximizes a heuristic function. This approach is a prime example for symbolic AI approaches, "good-old-fashioned-AI" ("GOFAI") [Hau85], which rely on logic and search on symbolic representations.

However, these knowledge-based approaches are severely limited by our ability to properly model the problem correctly and exhaustively. For example, in the case of Deep Blue it requires us to encode our knowledge about ches in a heuristic function to evaluate the board. Only then we can search for actions that maximize this function. Problems with large complexity would require tremendous efforts, which just become unfeasable at a certain point. A different approach would be devising (general) methods to learn the necessary domain knowledge from scratch, *tablula rasa*. As Alan Turing put it:

> Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain. Presumably the child-brain is something like a note-book as one buys it from the stationers. Rather little mechanism, and lots of blank sheets. [...] Our hope is that there is so little mechanism in the child-brain that something like it can be easily programmed. [TUR50]

The recent success of "AlphaGo" in 2016 against the long-time world-champion Lee Sedol [Dee] in the game Go is a milestones that perfectly demonstrates this shift towards "bottom-up" or subsymbolic methods. [Nil98] The increasing availability in computational power (and data) has enabled two subsymbolic methods to find large success in unclaimed territory such as copmuter vision or natural language processing. Namely those are neural networks and (stochastic) gradient descent. Combined they provide a general function approximator, that can be trained in a process akin to the learning described by Turing.

In the case of Go designing a powerful heuristic function was deemed not possible for humans. AlphaGo used (deep) neural networks and gradient descent to train a evaluation function based on a large database of expert moves. With the help of Monte Carlo Tree Search they used this function to play against itself and improve further. [SSS+17]

Building on this success DeepMind, the company behind AlphaGo, further improved the architecture. "AlphaGo Zero" and the generalization "AlphaZero" learn, without the help of the database of expert moves and surpassed the performance of AlphaGo 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. [SAH+20]

At this point our endeauvor begins, as the purpose of this writing is to apply the methods of AlphaGo to the game of Abalone.

2 System architecture

2.1 Software

2.1.1 Training framework

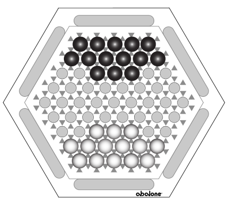
As there are existing frameworks that have implemented the system described in the AlphaZero paper in a more general and adaptable fashion, it has to be considered building on their foudation.

3 Analysis

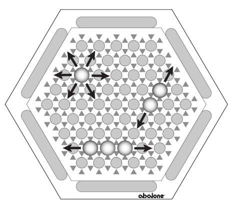
3.1 Environment

3.1.1 Abalone rules

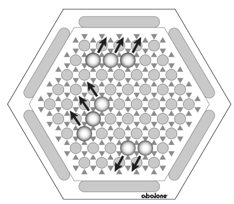
The goal of the game is to push six of the opponent's marbles off the playing field. The game's starting position is depicted in figure 3.1 (a). One, two, or three adjacent marbles (of the player's own color) may be moved in any of the six possible directions during a player's turn. We differentiate between broadside or "side-step" moves and "in-line" moves, depending on how the chain of marbles moves relative to its direction, which is shown in figure 3.1 (b) and (c).



(a) Starting position



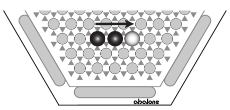
(b) "In-line" moves



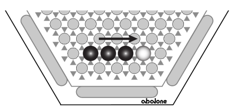
(c) "Side-step" moves

Figure 3.1: Basic moves [S.A]

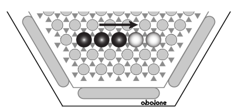
A move pushing the opponent's marbles is called "sumito" and comes in three variations, as shown by figure 3.2. Essentially, the player has to push with superior numbers and the opponent's marbles can not be blocked. This is the game mechanic that allows for pushing the marbles out of the game and winning.



(a) "2-push-1" sumito



(b) "3-push-1" sumito



(c) "3-push-2" sumito

Figure 3.2: Sumito positions allow pushing the opponent's marbles [S.A]

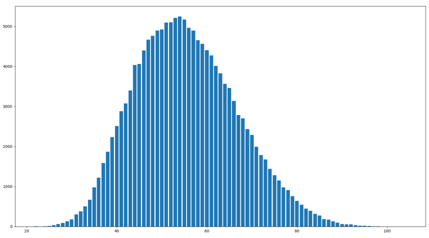


Figure 3.3: Counts of moves available for random for random player in 5 games

3.1.2 Abalone complexity

An important characteristic of a game environment is its complexity, which can be described in two relevant dimensions.

State space complexity The state space is the collection of all possible states the agent can be in.[RN21, p. 150] For Abalone this means we have to consider all possible board configurations with different numbers of marbles present. Additionally, we would have to correct duplicates that arise from the symmetries of the board. Ignoring this fact the following gives a good upper bound:

$$\sum_{k=8}^{14} \sum_{m=9}^{14} \frac{61!}{k!(61-k)!} \times \frac{(61-k)!}{m!((61-k)-m)!}$$

Game tree complexity The game tree defines the dependencies between board positions (nodes) and moves (edges). First we consider the branching factor (how many moves are possible in one position) of the game tree, which is on average 60. We combine that number with the height of the tree to get the total number of leaves. As the length of a game varies greatly, we use the average length of a game which is 87: 60^{87} [Lem05]

Putting Abalone's complexity in relation with other popular games, its state space complexity is on the same level as Reversi, whilst its game tree surpasses chess in complexity (c.f. table 3.1)

Game	state-space complexity (log)	game-tree complexity (log)
Tic-tac-toe	3	5
Reversi	28	58
Chess	46	123
Abalone	24	154
Go	172	360

Table 3.1: Abalone in comparison with other games [Cho09]

Bibliography

[Cho09] Pascal Chorus. Implementing a computer player for abalone using alpha-beta and monte-carlo search. Master's thesis, Citeseer, 2009.

[Dee] DeepMind. Match 1 - Google DeepMind Challenge Match: Lee Sedol vs AlphaGo. https://www.youtube.com/watch?v=vFr3K2DORc8&t=7020s.

[Hau85] John Haugeland. *Artificial Intelligence: The Very Idea*. MIT Press, Cambridge, Mass, 1985.

[Hig17] Chris Higgins. A Brief History of Deep Blue, IBM's Chess Computer — Mental Floss. https://web.archive.org/web/20170803130439/https://www.mentalfloss.com/article/5 history-deep-blue-ibms-chess-computer, July 2017.

[Lem05] NPPM Lemmens. Constructing an abalone game-playing agent. In *Bachelor Conference Knowledge Engineering, Universiteit Maastricht*. Citeseer, 2005.

[Nil98] Nils J. Nilsson. *Artificial Intelligence: A New Synthesis*. Elsevier, April 1998.

[RN21] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, Inc, fourth edition, 2021.

[S.A] Abalone S.A. Abalone rulebook. https://cdn.1j1ju.com/medias/c2/b0/3a-abalone-rulebook.pdf.

[SAH+20] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, December 2020.

[SSS+17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, October 2017.

[TUR50] A. M. TURING. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460, October 1950.