Deep Reinforcement Learning for Chess

David Hui

March 6, 2018

Contents

Chapter 1

Introduction

Knowledge Engineering and Machine Learning are two subfields of Artificial Intelligence which has produced decades of orthogonal research. Knowledge Engineering relies on searching algorithms which manipulate hand-crafted representations, whereas Machine Learning is capable of inferring representations from data. Recent advances in Artificial Intelligence such as AlphaGo combine these two subfields to create a hybrid architecture used for general problem solving. Such hybrid architectures were analysed in the context of Chess. This approach yielded an agent comparable to Stockfish, a hand-crafted agent which required decades of tuning, and more powerful than a deep Q network trained using Reinforcement Learning.

1.1 Significance

Since the dawn of western civilizations in hallowed antiquity, Artificial Intelligence has ignited a Promethean flame within the intellects of pre-eminent luminaries. In Classical Greece, Homer recalls the bewitching golden handmaidens created by Hephaestus and the towering warrior-giant Talos – a pinnacle of creation so lofty it was only attainable by the Olympian Gods. The prospect of creating an intelligence in our own image to support and enhance our lives was one which would change our world, but never yet come to fruition. To this day, Artificial Intelligence and its prodigal son Machine Learning are renowned by many major media outlets. Recent attainments such as AlphaGo have been celebrated and heralded as generational attainments, made by mortal man and reproducible by any with access to computational resources, rather than by an exclusive dodekatheon.

In the previous centuries, some of the finest minds would formulate theories of mind seeking to explain the entwined caduceus of life and intelligence. René Descartes hypothesised the existence of a mind-body duality, anticipating the hardware and software gulf that would dominate the latter stages of the 20$^{\text{th}}$ century.

In opposition to this paradigm, Baruch Spinoza sought to expound upon the behavioural trait of responding to signals from the environment within his posthumous magnum opus *Ethics*. An agent would classify percepts from the environment into positive and negative signals, mediated by its internal desire.

The field of Reinforcement Learning weaves an elaborate counterpoint of these chasmic countermelodies within an elegant analytic formulation. Capable of operating within any environment with labelled signals, Reinforcement Learning has been claimed to be capable of teaching an agent optimal strategies through a didactic process similar to Pavlovian conditioning. To substantiate the centuries of abstract philosophy with the raw power of modern day computational resources, the relatively simple game of Chess was chosen as the environment in which an agent would be trained.

The strategy game of Chess has a similar venerable history amongst the Indo-European courts, undergoing many dramatic changes and revisals.

INCLUDE SHANNON'S LEGENDARY PAPER

John McCarthy, a founding father of Artificial Intelligence, considered Chess to be its *Drosophila Melanogaster*. [?] This was because Chess is a game where all the rules are known, yet is large enough not to be trivially solved. Training an artificially intelligent agent on Chess enables analysis of its effectiveness over a wide variety of strategies and situations. Many different chess-playing algorithms are begin constantly developed, even today, making it ubiquitous to understanding intelligent systems, just as the common fruitfly is to the biologist.

During his time in Bletchley Park, Alan Turing would often play chess, reasoning that his thought processes could be represented by a Turing Machine. [?] He, along with John von Neumann and Claude Shannon, would independently derive the *Minimax algorithm* to find the best possible next move. [?] [?] Used by nearly all chess engines, including the world-champion-beating Deep Blue, minimax analyses *game trees*, a subset of all possible game futures generated from *searching algorithms* [?].

As the game can produce very deep search trees, minimax is often limited by a specified depth. [?] A shortcoming of this approach is its reliance on an evaluation function used to determine the likelihood of victory given a state. [?] Many Chess engines rely on hand-crafted evaluation functions which may take years and sometimes a decade to develop. [?] Recent developments in Artificial Intelligence have shown that *Deep Learning* can be used to approximate such functions given enough labelled datapoints. [?] To reduce the size of datasets used, *Reinforcement Learning* algorithms enable a chess agent to learn by playing against itself. [?]

Inspired by behavioural psychology, [?] Reinforcement Learning is a promising methodology which has trained agents such as TD-Gammon or Deep-Q-Networks, which can outperform humans in Backgammon and Atari games. [?] [?] Both these architectures rely on one neural network. In contrast to this, AlphaGo, the engine which beat the World Champion in Go, is a hybrid architecture which combines reinforcement learning with minimax and searching algorithms. [?]

Such hybrid architectures harness the efficiency of classical searching algorithms with the large-scale parallel processing inherent within Deep Learning. This also means that the weaknesses of both strategies are apparent, such as the tendency of Deep Learning to overfit, or for search algorithms to generate large search trees. To analyse the effectiveness of this hybrid architecture, I will reimplement *Giraffe*, a predecessor of AlphaGo. Developed by an AlphaGo Scientist, Matthew Lai, Giraffe was trained on a CPU in Chess, achieving a strength comparable to that of a human grandmaster. [?]

Upon reimplementing Giraffe to be executed on a GPU, a successful agent would beat an opponent playing random moves within 40 rounds, to 95 % significance. I will also estimate the agent's strength with ELO, and evaluate finer aspects of its performance in the Strategic Test Suite. This suite is made up of board positions chess engines typically find difficult to play against, making it a more fine grained metric. [?]

If there are no problems in getting this far, an optional extension to this project would be to investigate the effects of changing the searching algorithm which generates the game tree. Giraffe is constructed around probabilistic search, whereas AlphaGo is based on Monte-Carlo Tree Search. [?] [?] Changing the searching algorithm and creating a successful agent similar to that of AlphaGo would highlight the flexibility of this architecture.

Overall, combining rule based algorithms with Deep Learning would unify the strengths of rule-based analysis with statistical data-based inference, perhaps providing a foundation of the direction of research to come. It is possible that this new architecture may yield a new strategy for playing Chess, enhancing the knowledge of the game as well as providing a new opponent for humans to train against.

Chapter 2

Preparation

Chapter 3

Implementation

Chapter 4

Evaluation

Chapter 5

Conclusion