Literature Survey

Thien Nguyen

26th October 2017

1 Introduction

1.1 Problem Background

The purpose of this project is to explore approaches to tackle the game of English Draughts via the use of contemporary machine learning techniques. First, we will study previous historical successes in the field, and look at the compoments that helped build their systems. Then, we will look at contempoary methods of computer science that could be used to evolve the historical systems.

1.2 Terms

- **Genetic Algorithm** An evolutionary method that solves optimisation problems. This is based on Darwin's theorem of perpetually evolving populations of solutions.
- **Machine Learning**: computer algorithms related to a branch of computional learning theory. It allows computers to learn.
- **Neural Network** A computational model based on the operations of "interconnected processing elements, which process information by their dynamic state response to external inputs."
- **Draughts** In this project we will be using the British Draughts (or, American Checkers) rules. For the sake of clarity these arguments will be enforced:
  1. The game is played on an 8x8 checkerboard.
  2. Jumps are enforced in the event that it is possible for a player to make one.
  3. Multiple jumps are enforced.
  4. In the event that a piece performs a multiple jump, if it lands on a promotion row (where a piece is promoted into a king), then its move is terminated.
- **Ply**: a ply refers to one turn taken by one of the players.

- **ELO:** A rating system devised by Arpad Elo. It measures the relative skill levels of players in competitive games. It is commonly used in Chess, American college football and Scrabble.
- **Complete Information**: Compoments and the status of the game are wholly shown throughout, to the players.
- **Zero-Sum Game**: A situation where one players benefit is offset by the loss of another player.
- **Game Theory**: The science of strategy, or in more general, the decision-making of independent and competing actors in a strategic setting.

2 Themes

2.1 History

The use of machine learning and zero sum board games is not novel; Research within this particular field has currently moved on to tackle games with larger search spaces, most notably Chess [17], and Go[25]. Draughts, however, occupies a fundamental place in game theory - It is the most complex game ever solved to date. [18, 28] Historically, Draughts has been used as a testing ground for artificial intelligence and computional performance since the early introduction of computers. An early design of machine learning to play Draughts was devised by Samuel (1959)[22] (who also coined the term 'Machine Learning'), where his algorithm revolved around using a linear combination of weighted features the board contained, such as the number of pawns, number of pawns positioned along the central diagonal, and so on. During its design, Samuel also described an early concept of Alpha Beta Pruning. The weights were then trained by the algorithm playing itself in a form of genetic programming. A later investigation by a checkers magazine evaluated Samuel's player to "below Class B",[23, 12] where Class B being categorised with an ELO range of "1600-1799".

Schaeffer et al. took the concept of solving Draughts further, and produced a paper that describes how the game could be be weakly solved using mathematical calculations in 1996, [24] with their introduction of Chinook. This was made possible by using a vast database of games pre-played from grandmaster tournaments. Chinook also uses a dictionary of end games, containing all possible moves for a board containing at most 8 pieces.

Schaeffer took this further, proving a weak solution to Draughts after 10^{14} calculations of end-game positions. Using numerical calculations, it was mathematically proven that for two draughts players making no wrong moves, the game always ends in a draw [18].

While it is not disputed that Chinook is a landmark in computational limits - as of writing, Chinook remains the worlds strongest American checkers player with an ELO of 2890 - Chinook was largely made possible through human intervention; the player was effectively told to play in a particular manner.

Fogel et al. (2001) used Samuel's system as a base; improving it's evaluation method, using feed-forward convolutional neural networks instead of a linear combination. Their resulting player, Blondie24[13], was then taught to play without priori knowledge.

This was accomplished by using a convolutional neural network that takes in 32 inputs (each input representing a position on the board), with a single output. Two hidden layers exist between the input and output layers. This resulting neural network was used to evaluate the status of the board, determining the overall performance scaling from -1 to 1 that suggests which player has the advantage.

Blondie24 reached an average ELO rating of 1901.

Since Blondie24, the field of machine learning has dramatically evolved, most notably since the 2010's.

2.2 Research

Since then, there has been a non-trivial number of papers within Checkers, Genetic Algorithms and Convolutional Neural Networks.

Sergei Perez (UC Irvine) describes the use of genetic algorithms to evolve weights for a convolutional neural network as a more performant alternative to back-propagation, in nearly all situations, other than when the number of generations generated for neural networks are small.[21]

In 2017 Lingxi Xie et al. took it a step further and discussed using genetic algorithms to compute optimal deep learning structures automatically, using an image recognition dataset as the benchmark for competitiveness against the networks.[29]

Kusiak et al. (2007) have devised the use of pure genetic algorithms as the premise to evaluate the board. [16] they describe the use of 25 heuristics that, when used in unison, can determine the evaluation of a given board. They continue to go further on and specify that various combinations of heuristics, when used in different stages of the game, prove to be the most promising of outcomes.

Cobbe, et al. improves Blondie24 by using an evaluation function that consists of Support Vector Machines, which achieves locally optimal performance in approximately a third as many generations than its typical evolution. [10] Interestingly, they suggest the use of the ELO algorithm as the tournament rating system. This could prove to be useful background information as it would provide an reliable relative rating of a given agent, using fewer games.

Fogel's board was used as the premise for Al-Khateeb's Thesis (Ph.D)[1], conducting a through evaluation of the different components that make Blondie24, including the Look-ahead Depth[4], The tournament approaches [3], the value of piece differences [2].

Alternative contemporary machine learning techniques have been exercised against Draughts. Franken et al. (2003) used draughtsboards as an evaluator for their rendition of an AI through Particle Swarm Optimisations.[14]

Quetzalcoatl Toledo-Marin et al. (2016) performed studies on attacking strategies using the game of checkers as a reference (which however can be applied to any other zero-sum game), where they mathematically prove that for an offensive player, maximising the offensive improves their probability to win.[28]

3 Proposed Direction

At the end of the chapter you should briefly explain how your own work builds on and differs from the work that has gone before it.

The approach taken would follow the a similar framework as described for Blondie24, but we will be considering seperate weights for the different stages of the game.

Like Blondie24, a convolutional neural network would be used to evaluate the board. However, it is undecided whether to use one network with three outputs, or to have separate neural networks that will evaluate the board contingent on the phase of the game.

The genetic algorithm would be used to improve the quality of the weights of the neural networks. We will need a population of neural networks (which have varying weights), with a round-robin style tournament as the evaluation function. Finalists are chosen as the basis for the next generation.

Experimentation of techniques would be conducted in these areas, and I could imagine that they would be tested independently of each other.

- Effectiveness of the min-max algorithm
- Genetic Crossover Algorithm
- Genetic Mutation Algorithm
- Effectiveness of the Neural Network
- When/where to perform the genetic algorithm.

An interface is also needed in order to test the AI. There are multiple options for this; It could be built for the web, which would interact with the AI, or we can simply use a program built locally for it.

In order to test it's performance, we could, like Chellapilla and Fogel's paper, make an account for the AI at a popular gaming site and act as the intermediary for it. This would provide a lot of benefits, such as a relative ELO amongst other players on the site.

Considerations will need to be made such that the game does not strictly overfit itself to play in a particular manner.

References

[1] Al-Khateeb, B. *Investigating evolutionary checkers by incorporating individual and social learning , N-tuple systems and a round robin tournament.* PhD thesis, University of Nottingham, 2011.

[2] Al-khateeb, B., and Kendall, G. The Importance of a Piece Difference Feature to Blondie24.

[3] Al-Khateeb, B., and Kendall, G. Introducing a round robin tournament into blondie24. *CIG2009 - 2009 IEEE Symposium on Computational Intelligence and Games 44*, 0 (2009), 112–116.

[4] Al-Khateeb, B., and Kendall, G. Effect of look-ahead depth in evolutionary checkers. *Journal of Computer Science and Technology 27*, 5 (2012), 996–1006.

[5] Al-khateeb, B., and Kendall, G. Introducing Individual and Social learning into Evolutionary Checkers. *IEEE Transactions on Computational Intelligence and AI In Games 4*, 4 (2012), 258–269.

[6] Allsop, D. D. U. *Artficial Intelligence Techniques Applied To Draughts Artificial Intelligence Techniques Applied To Draughts.* PhD thesis, Durham University, 2013.

[7] Boonzaaier, D. Training a Neural Network for Checkers.

[8] BÃck, T. Fogel on BÃck. *65.44.200.132* (1997), 26–27.

[9] Chellapilla, K., and Fogel, D. B. D. D. B. Evolving Neural Networks to Play Checkers without Expert Knowledge. *IEEE Transactions on Neural Networks 10*, 6 (1999), 1382–1391.

[10] Cobbe, K., Lee, P., and Gomez-Emilsson, A. Accelerating Checkers AI Evolution. *Cs229.Stanford.Edu*, 2–5.

[11] Elyasaf, A. Evolving Hyper-Heuristics using Genetic Programming.

[12] Fogel, D. B. Evolving a Checkers Player.

[13] Fogel, D. B. *Blondie24: Playing at the Edge of AI.* 2001. DOI: 10.1016/B978-155860783-5/50016-7.

[14] Franken, N., and Engelbrecht, A. P. Comparing PSO structures to learn the game of checkers from zero knowledge. *2003 Congress on Evolutionary Computation, CEC 2003 - Proceedings 1* (2003), 234–241.

[15] Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. Self-Normalizing Neural Networks. *arXiv* (2017).

[16] Kusiak, M., Walezedik, K., and Jacek, M. Evolutionary approach to the game of checkers. 600–661.

[17] LAI, M. Giraffe: Using Deep Reinforcement Learning to Play Chess.

[18] MULLINS, J. *Checkers 'solved' after years of number crunching*. 2007.

[19] NGUYEN, H. H., AND CHAN, C. W. Multiple neural networks for a long term time series forecast. *Neural Computing and Applications 13*, 1 (2004), 90–98.

[20] PAPPA, G. L., OCHOA, G., HYDE, M. R., FREITAS, A. A., WOODWARD, J., AND SWAN, J. *Contrasting meta-learning and hyper-heuristic research: The role of evolutionary algorithms*, vol. 15. 2014. DOI: 10.1007/s10710-013-9186-9.

[21] PEREZ, S. Apply genetic algorithm to the learning phase of a neural network.

[22] SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development 44*, 1.2 (2000), 206–226.

[23] SCHAEFFER, J. *One Jump Ahead*. 1997.

[24] SCHAEFFER, J., AND LAKE, R. Solving the Game of Checkers. *Games of No Chance 29* (1996), 119–133.

[25] SILVER, D., SCHRITTWEISER, J., AND SIMONYAN, K. Mastering the game of Go without human knowledge. *Nature 550* (2017), 354–359.

[26] STANLEY, K. O., AND MIIKKULAINEN, R. Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation 10*, 2 (2002), 99–127.

[27] THESIS, A. Combining Genetic Algorithms and Neural Networks : The Encoding Problem. *Thesis ANN p 67*, December (1994), 1–67.

[28] TOLEDO-MARÃN, J. Q., DÃAZ-MÃ©NDEZ, R., AND MUSSOT, M. D. C. Is a good offensive always the best defense? 1–12.

[29] XIE, L., AND YUILLE, A. Genetic CNN.