CPSC 540 Literature Survey:
Modelling Human Behaviour in Board Games

Greg d'Eon, Shivam Thukral, Abner Turkieltaub

April 1, 2020

Past advances in artificial intelligence have often been closely linked with board games. Two prolific examples are Deep Blue and AlphaGo , which both represented milestones in AI research with their respective victories over world-class players in chess and Go. However, such programs are typically created with the goal of reaching superhuman performance. In our project, our goal is instead to develop a **predictive model that can play chess like a human**.

To help us toward this goal, we use this literature review to investigate two broad topics. First, we review state-of-the-art machine learning techniques for board games. Our aim is to understand existing ideas, methods, and architectures designed for chess that we can leverage in our model. Second, we explore the literature on chess players' cognitive processes, which documents how humans perceive the board and make decisions during games. We hope to use findings from this psychological research to help us more accurately predict human moves.

1 Machine Learning for Games

Games, such as Go, chess, Atari games, and Hanabi, are one of the most well-studied domains in artificial intelligence. Initial attempts to solve these problems combined sophisticated search techniques, domain-specific adaptations, and hand-crafted features. Over the years, these algorithms have shifted away from these types of pre-existing knowledge, replacing them with deep learning and self-play, making methods more applicable to real-world settings. In this section, we discuss the deep learning methods that have achieved superhuman performance in these challenging domains.

1.1 AlphaGo: Deep Neural Networks for Go

In 2016, DeepMind began working on Go with AlphaGo [1]. Their approach broadly consists of three parts. The first is a **policy network**, which predicts which moves are most promising in any position of the game. This policy network was initially trained through supervised learning to predict expert players' moves from 30 million online games. Then, it was refined through reinforcement learning by having AlphaGo play against previous instances of itself.

The second component is a **value network**, which predicts each player's chance of winning from a given state of the game. To train the value network, they used supervised learning, predicting the outcomes of 30 million self-play games. Lastly, AlphaGo uses **Monte Carlo tree search** to find strong moves, using the policy network to guide its search and the value network to quickly predict a game's outcome. We view this search procedure, combining deep learning models with existing MCTS algorithms, as the most significant contribution of this work.

AlphaGo was significantly stronger than all existing programs, winning 494/495 games against other programs. The program gained much appreciation when it defeated the European champion 5 games to nil, being the first computer program to beat a professional Go player.

1.2 AlphaGo Zero: Learning from Scratch

In the following year, DeepMind introduced AlphaGo Zero [2]. The key change in AlphaGo Zero is that it is not provided any human knowledge about Go beyond the rules. Specifically, there are four key differences between AlphaGo Zero and AlphaGo:

- **Self-play**: AlphaGo Zero is only trained through self-play, and human games are not used to initialize the network.
- **No handcrafted features**: AlphaGo had many predefined features that were handcrafted for Go. The new version only uses black and white stones from the board as input features.
- **Single network**: The policy and value network were combined into a single network that performs both tasks.
- **Simplified MCTS**: It uses a simplified version of tree search that relies upon this single neural network to evaluate the position and sample moves, with Monte-Carlo roll-outs.

AlphaGo Zero was trained using 40 days of reinforcement learning on specialized hardware, including 64 GPU and 19 CPU servers, generating 29 million self-play games. One of the major drawbacks of this project is the cost of this hardware, including custom components, which was quoted around $25 million.

AlphaGo Zero was evaluated using an internal tournament against two previous versions of AlphaGo, along with other previous Go programs. AlphaGo Zero outperforms all the existing algorithms, even beating the strongest version of AlphaGo by 100 games to 0. This neural network was more powerful than AlphaGo: it no longer relies on human expertise, and instead learns from itself.

1.3 AlphaZero: One Algorithm, Many Games

AlphaGo Zero's successor came out in 2018 as AlphaZero [3]. Unlike both of the previous editions, which were designed for Go, AlphaZero's architecture can also be trained to learn chess and Shogi. It also included two key changes from AlphaGo Zero. First, AlphaZero was modified to account for draws, which are common in chess and Shogi, but virtually non-existent in Go. Second, AlphaZero no longer exploited invariances to rotating or reflecting the game board, which provided useful data augmentations in Go, but do not apply in chess or Shogi.

Aside from these differences, AlphaZero uses the same convolutional neural network architecture and Bayesian optimized hyperparameters as in AlphaGo Zero. During training, 5000 first-generation TPUs were used to generate self-play games, and 16 second-generation TPU trained the network. Training took 9 hours for chess, 12 hours for Shogi, and 13 days for Go.

AlphaZero was evaluated against top engines in each of the three games. In Go, AlphaZero defeated AlphaGo Zero, winning 61% of games. In chess, AlphaZero defeated Stockfish by winning 155 games and losing 6 games out of 1000 games played. In Shogi, AlphaZero defeated Elmo by winning 98% of games when playing black and 91% overall. In particular, its chess playstyle is distinctive and unorthodox, yet creative and dynamic. This playing style is unlike any traditional chess engines, showing how a single algorithm can discover new knowledge in a range of settings.

1.4 MuZero: Learning and Planning in Unknown Environments

Planning algorithms, like those used in AlphaZero and its predecessors, are useful only when we have information about the dynamics of the environment. This issue limits their use in real-world applications, where dynamics are ever-changing. To cope with this problem, model-based reinforcement learning approaches initially learn a model of the environment dynamics, then learn how to plan with respect to the learned model.

MuZero [4] is a model-based RL approach which is able to play Atari games, but can still learn board games such as chess and Go. MuZero extends from AlphaZero using its search-based policy iteration algorithm, but incorporates a learned model of the game's rules in the training phase.

The input to the model is an image of the game board or Atari screen, which is transformed into a hidden state. The hidden state is updated iteratively through a recurrent process that receives previous hidden state and proposed next action. At each step model gives three values: which move to play, the predicted winner and the reward (points scored by playing a move). The model is trained to accurately output these three values, so as to match the compared estimates of policy and value generated by search tree as well the observed reward.

MuZero Reanalyze achieved 713% median normalisation score compared to 192%, 239% and 431% for previous state-of-art model free approaches IMPALA, Rainbow and LASER respectively. This approach does not require any knowledge of the game rules or environment dynamics, giving a way to apply such architectures to real-world problems.

1.5 The Hanabi Challenge

Hanabi is a cooperative card game for 2 to 5 players. It is a kind of collective solitaire: each player can see other players' hands, but not their own. The game revolves around giving limited hints to other players, giving them partial information about their hands. Since the game is cooperative, our objective of developing an AI capable of playing like a human is a natural goal.

In 2019, Bard et al. [5], proposed Hanabi as a new challenge for AI. Unlike competitive games, like Chess, Go, Shogi, Poker, or Starcraft, where AI has been really successful, it is not as clear what means to be good at Hanabi. For instance, a human player could be good at playing with a group of friends, but perform poorly with others. With this issue in mind, the authors propose two settings:

1. **Self-play setting:** find conventions for the entire team that achieve good scores. This approach is often used by human players, who use conventions to encode additional information in the hints.
2. **Ad-hoc team setting:** develop agents capable of learning from, adapting to, and playing well with new teammates.

The paper motivates the problem well, explains the game rules and how human players approach it, describes the two settings aforementioned, and summarizes previous work. They also develop agents using reinforcement learning that achieve good results in the self-play setting, but don't beat the best hand-coded bots and perform very poorly in the ad-hoc team setting. For future work, there is still some room to improve in self-play, while ad-hoc teams are an open problem.

2 Human Play in Chess

Each of the artificial intelligence approaches described above were focused on playing the best possible moves. However, the goal of our project is to play "human-like" moves, and humans rarely play optimal moves. To understand how humans deviate from perfect play, we reviewed psychology papers on the cognitive processes used in chess, hoping to find how amateur players consider and select their moves to inspire our machine learning approaches.

2.1 Laboratory Studies: Perceiving Pieces and Solving Puzzles

Chess has long been studied in psychology laboratories, as it provides a well-defined environment to study how experts solve difficult problems. Prior experiments have focused on two specific components of this problem-solving process. The first is players' **perception** of the board, and how they process and encode the state of the game into their working memory. The second is players' methods to **search** for good moves, and how experts' and novices' thought processes differ.

Chase and Simon's study of perception in chess [6] is perhaps the most influential work in this area. In their experiments, players were asked to recreate a chess position, either working from memory after seeing the board for 5 seconds, or by glancing aside at another prepared board. Their results showed that highly skilled players were much better than novices at recreating typical positions from memory, but this advantage disappeared when the position consisted of randomly placed pieces. Further, after taking one glance at the board, skilled players tended to place pieces with specific relationships to each other – for example, chains of same-coloured pawns defending each other. Together, these results suggest that chess experts perceive the board as a collection of "chunks", where each chunk consists of structures that commonly appear in games.

Charness et al. [7] later validated this notion of "chunking" by tracking players' eye movements as they solved a number of chess puzzles. They found that intermediate players tended to fixate on each of the individual pieces, while expert players were much more likely to fixate on empty squares or the most tactically relevant pieces. As before, these results imply that experts perceive the board as a combination of several local structures instead of a grid of unrelated pieces.

On the search side, several experimental papers have studied how players pick moves. These experiments typically have players think out loud as they work through a chess position. One of the earliest of these studies is by De Groot [8], who found that experts don't analyze more variations, but spend more of their time analyzing good moves. Charness [9] only partially reproduced these results, finding that more skilled players could sometimes search much deeper than novices. Campitelli and Gobet [10] reconciled these differences by showing that experts can read deeper, but only do so in positions that need huge search trees to solve. In sum, while skilled players can read deeper when they must, their search process is also strongly guided by heuristics about which moves to consider.

Lastly, Van der Maas and Wagermakers [11] developed a standardized test for chess proficiency. They compiled a list of questions, ranging from choose-a-move tactics puzzles and predict-the-move scenarios from real games to questionnaires about players' motivation and verbal knowledge. They found that players' answers to these questions were strongly correlated with their ELO, allowing them to use the test to estimate a player's skill level. In a sense, this test is a flipped version of our project's goal: we hope to use player's skill level to predict their moves in a variety of games.

2.2 Online Chess: Studying Full Games

In the past 20 years, online chess databases and servers have steadily grown in popularity. As these archives document a great amount of detail about every game, they provide an exciting new source of data, making them an ideal "research vehicle" for psychologists [12]. This data is even useful without considering individual games: Harreveld et al. [13] compared players' ELO ratings across different time controls to study the importance of fast and slow thinking processes.

We are aware of two papers that have extended this type of analysis to individual games. First, Sigman et al. [14] found that weaker players spend less time on middlegame moves and more time on endgame moves, and that players tend to move faster when their opponent does too. They also used empirical win rates to show that, for example, having several extra seconds in the endgame can be as valuable as having an extra piece. Second, Slezak and Sigman [15] showed that players change their strategy when facing stronger opponents. Their data suggested that people play slower, more accurate moves when they are outranked by their opponent.

Overall, it is clear from prior work that studying board games, such as chess, has given psychologists unique insights about human cognition. However, it appears that there is still plenty to be learned from online chess databases, which give researchers access to a huge amount of in-the-wild data. This failing gives us confidence that a predictive model of human moves would be a novel and valuable contribution to the field of cognitive science.

References

[1] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[2] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[3] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[4] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *arXiv preprint arXiv:1911.08265*, 2019.

[5] Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

[6] William G. Chase and Herbert A. Simon. Perception in chess. *Cognitive psychology*, 4(1):55–81, 1973.

[7] Neil Charness, Eyal M. Reingold, Marc Pomplun, and Dave M. Stampe. The perceptual aspect of skilled performance in chess: Evidence from eye movements. *Memory & cognition*, 29(8):1146–1152, 2001.

[8] Adriaan D. De Groot. *Thought and choice in chess.* The Hague: Mouton, 1946/1978.

[9] Neil Charness. Search in chess: age and skill differences. *Journal of Experimental Psychology: Human Perception and Performance*, 7(2):467, 1981.

[10] Guillermo Campitelli and Fernand Gobet. Adaptive expert decision making: Skilled chess players search more and deeper. *ICGA Journal*, 27(4):209–216, 2004.

[11] Han L. J. Van Der Maas and Eric-Jan Wagenmakers. A psychometric analysis of chess expertise. *The American journal of psychology*, pages 29–60, 2005.

[12] Nemanja Vaci and Merim Bilalić. Chess databases as a research vehicle in psychology: Modeling large data. *Behavior research methods*, 49(4):1227–1240, 2017.

[13] Frenk Van Harreveld, Eric-Jan Wagenmakers, and Han L.J. Van Der Maas. The effects of time pressure on chess skill: an investigation into fast and slow processes underlying expert performance. *Psychological research*, 71(5):591–597, 2007.

[14] Mariano Sigman, Pablo Etchemendy, Diego Fernandez Slezak, and Guillermo A Cecchi. Response time distributions in rapid chess: a large-scale decision making experiment. *Frontiers in neuroscience*, 4:60, 2010.

[15] Diego Fernandez Slezak and Mariano Sigman. Do not fear your opponent: Suboptimal changes of a prevention strategy when facing stronger opponents. *Journal of Experimental Psychology: General*, 141(3):527, 2012.