Cailyn Craven

AlphaGo: The Power of a Hybrid System

For decades, artificial intelligence researchers have used board games as a platform to develop and test AI algorithms. At the beginning of the 2010s, AI systems had achieved success against many human games, including the milestone of IBM's Deep Blue defeating the world chess champion Gary Kasparov in 1996 [2]. However, beating a professional player in the highly complex, ancient game Go remained a long-standing, grand AI research challenge. Then, in October 2015, DeepMind's AlphaGo beat the European Champion Fan Hui, the first time a computer program defeated a professional Go player without a handicap. In March 2016, AlphaGo played a series of matches with Lee Sedol, the winner of 18 international titles. Despite the worldwide confidence that Sedol would win, AlphaGo won four out of the five games [3]. Given the rapid improvement in AlphaGo's ability to defeat higher levels of human players, spectators outside of the AI field may have hypothesized that DeepMind had discovered a novel new method. In reality, AlphaGo's success is a striking example of AI researchers creating hybrid systems that combine multiple techniques to create programs that perform better than the sum of their parts. DeepMind innovatively incorporated various research avenues that showed potential for playing Go, including Monte Carlo Tree Search, Reinforcement Learning, and Deep Learning. Even after developing an AI agent capable of defeating human Go champions, DeepMind continued iterating their program to remove the agent's dependence on human domain knowledge. Here's a glimpse into how DeepMind synthesized decades of AI research into AlphaGo.

One earlier approach that showed promise for a Go-playing AI agent involved the Minimax algorithm, a type of adversarial game-tree search used to defeat the world chess champion. Minimax incorporates an evaluation function that assigns values to game tree leaf nodes at a certain depth, and the algorithm searches through the game tree to pick the best possible move. AlphaBeta Pruning improves the search efficiency by ignoring portions of the search tree that don't impact the optimal action (Norvig and Russell p. 148 -155)[4]. Unfortunately, Minimax with AlphaBeta pruning is a weak option for Go because the branching factor means that the AI agent would be limited to looking ahead a small number of moves. Moreover, it is challenging to write a useful evaluation function for Go because so things stay in flux until the end of the game (Norvig and Russell p. 162)[4].

Researchers realized that a Monte Carlo Tree Search (MCTS) strategy makes it possible to use a game-tree search for Go that overcomes some of Minimax's weaknesses. MCTS builds a statistics tree that partially maps onto the entire game tree, estimating the win percentage of simulations, or rollouts, of complete games, starting from a state instead of using a heuristic evaluation function. A selection policy balances exploring states that have had few simulations run from them with exploiting states that have performed well to estimate their value better (Norvig and Russell p. 162-164)[4]. At the beginning of the twenty-first century, Go programs with MCTS methods began to achieve amateur levels [1].

With game tree search showing promise for playing Go, another AI research area also started burgeoning: deep neural networks. Researchers proposed neural networks that loosely simulated a system of neurons in the brain in the 1940s, and in the subsequent decades, research interest in neural networks periodically waxed and waned. In the early 2000s, the availability of large amounts of data and GPUs' parallel computational capabilities led to a deep learning revolution[5]. In 2014, researchers at the University of Edinburgh used supervised learning with datasets from human-played Go games to train a neural network to predict a human Go player's probability of making a possible move from a given position. Researchers found that a Deep Convolutional Neural Network method that played the most likely move for a skilled player outperformed several programs designed to play Go like GNU Go [6].

At this stage in AI research, researchers considered Monte Carlo Tree Search methods state of the art. Still, Deep Convolutional Neural Networks also showed potential for playing Go and sometimes performed better than MCTS methods[6]. In this environment, DeepMind turned its attention to developing a hybrid system that incorporated both techniques. The AlphaGo systems that played Hui and Sedol had three main components: two deep neural networks and tree search. The policy network trains on high-level human games to imitate those players and outputs the probability of making a particular move. The value network predicts the winner of matches played by the policy network against itself and gives the probability of winning in a specific place. MCTS uses the policy network to narrow down the search to high-probability moves and the value network to evaluate the tree positions[1]. Previous Go-programs that only used Monte Carlo Tree Search or Deep Convolutional Neural Networks failed to defeat professional Go players without a handicap. The hybrid system proved more robust than its parts and capably defeated human Go champions. Notably, instead of a new method, AlphaGo incorporated multiple fields that AI researchers had been investigating for decades.

Even after defeating a nine dan Go professional on a full-sized board without handicaps, DeepMind kept iterating, seeking to go further than existing AI research had gone. AI Go-programs had long incorporated a significant amount of human domain knowledge. Given the complexity of the game, there was a sense that an AI agent would benefit from incorporating some human domain knowledge and perhaps that such expertise was a necessity to defeat human Go champions [1]. For the AlphaGo Hui and Sedol versions, the policy network trained using supervised learning from human expert moves. Unfortunately, even a relatively large dataset of high-level human games only includes a relatively small number of moves for a game like Go. An AI agent trained through supervised learning on high-level Go games alone would have significant weaknesses regarding actions it didn't know how to play (Norvig and Russell p.789) [4]. After training the AlphaGo policy network with supervised learning, DeepMind refined the policy network with policy-gradient reinforcement learning to prevent deficiencies [1].

In 2017, DeepMind published a *Nature* paper announcing another critical AI milestone. They had developed AlphaGo Zero, where the neural network trained purely with a self-play reinforcement learning algorithm that used MCTS to play each move. AlphaGo Zero didn’t use supervised learning or human-generated data. While the algorithm was less accurate at predicting what moves a human would make, AlphaGo Zero performed better than human-trained AlphaGo versions [1]. Yet again, DeepMind’s experimentation with methods led to better performance.

In conclusion, to truly understand how DeepMind achieved superhuman Go performance, it is essential to understand what the AI research landscape looked like in 2014. After decades of AI research, both game tree search and deep learning methods showed promise for playing Go. Unfortunately, no technique seemed capable of defeating professional level players on its own. DeepMind solved a grand AI research challenge by designing a system that successfully combined methods from multiple subfields. Exploring DeepMind's approach, one can see how accomplishing the milestone of defeating a professional Go player represented the culmination of decades of AI research. A famous quote attributed to Isaac Newton says, "If I have seen further, it is by standing on the shoulders of giants." After defeating human champions, DeepMind sought to go further. A long-held belief was that human expertise would help a program play a complex game like Go. DeepMind continued experimenting with methods and achieved better performance using reinforcement learning without human knowledge.

Citations

1. **Mastering the game of Go without human knowledge.** Silver et al. *Nature* Vol 550, October 2017.
2. *Deep Blue vs. Gary Kasparov.* Wikipedia. https://en.wikipedia.org/wiki/Deep_Blue_versus_Garry_Kasparov
3. *AlphaGo The Movie: Full Documentary. https://www.youtube.com/watch?v=WXuK6gekU1Y*
4. Russell and Norvig. *Artificial Intelligence: A Modern Approach, 4th Edition.*
5. *Deep Learning.* Wikipedia. https://en.wikipedia.org/wiki/Deep_learning
6. **Teaching Deep Convolutional Neural Networks to Play Go.** Clark and Storkey. arXiv:1412.3409v2 December 2014.