Research Review

Mastering the game of Go with deep neural networks and tree search

Introduction and research objectives

The article “*Mastering the game of Go with deep neural networks and tree search”* introduces a new approach to computer Go that uses ‘value networks’ to evaluate board positions and ‘policy networks’ to select moves. These deep neural networks are trained by a novel combination of supervised learning from human expert games, and reinforcement learning from games of self-play. Using this new search algorithm, the program **AlphaGo** (developed by Google DeepMind team) achieved a 99.8% winning rate against other Go programs, and defeated the human European Go champion by 5 games to 0. This is the first time that a computer program has defeated a human professional player in the full-sized game of Go, a feat previously thought to be at least a **decade** away.

The outcome achieved by DeepMind is impressive, considering the space and time complexity of the problem they decided to solve. Go is a perfect information game, and these games may be solved by recursively computing the optimal value function in a search tree containing approximately b^d possible sequences of moves, where b is the game’s breadth (number of legal moves per position) and d is its depth (game length). In large games, such as chess ($b \approx 35$, $d \approx 80$)1 and especially Go ($b \approx 250$, $d \approx 150$)1, exhaustive search is infeasible.

DeepMind developed a strategy based on the concept of the **training pipeline**, composed by three main steps:

1. Step 1 – Supervised learning of policy networks: A 13-layer policy network, termed as SL Policy Network, that learns to predict the best possible move at a state of the game. The neural network takes input features from the board position and outputs the probability of each move on the board being the actual next move. The network predicted expert moves on a held out test set with an accuracy of 57.0% using all input features, and 55.7% using only raw board position and move history as inputs, compared to the state-of-the-art from other research groups of 44.4%.

2. Step 2 - Reinforcement learning of policy networks: The second stage of the training pipeline aims at improving the policy network by policy gradient reinforcement learning, to achieve the final goal of winning rather than prediction accuracy of current move which is not important.

3. Step3 - Reinforcement learning of value networks: The final stage of the training pipeline focuses on position evaluation, estimating a value function $v^p(s)$ that predicts the outcome from position s of games played by using policy p for both players.

The proposed AlphaGo program combines all these techniques with the Monte-Carlo tree search (MCTS) technique to achieve the record-breaking feat of beating a human Go champion. The use of deep neural networks for SL policy learning and value function evaluation contribute to the novelty of this work.

Results

The AlphaGo progam was evaluated against other Go programs. The results of the tournament suggest that single-machine AlphaGo is many *dan* ranks stronger than any previous Go program, winning 494 out of 495 games (99.8%) against other Go programs. To provide a greater challenge to AlphaGo, we also played games with four handicap stones (that is, free moves for the opponent); AlphaGo won 77%, 86%, and 99% of handicap games against Crazy Stone, Zen and Pachi, respectively. The distributed version of AlphaGo was significantly stronger, winning 77% of games against single-machine AlphaGo and 100% of its games against other programs.

A computer beating a human consistently at the game of Go was the feat achieved for the first time in history of the game of Go.

Go is exemplary in many ways of the difficulties faced by artificial intelligence: a challenging decision-making task, an intractable search space, and an optimal solution so complex it appears infeasible to directly approximate using a policy or value function. The previous major breakthrough in computer Go, the introduction of MCTS, led to corresponding advances in **many other domains**; for example, general game-playing, classical planning, partially observed planning, scheduling, and constraint satisfaction. By combining tree search with policy and value networks, AlphaGo has finally reached a professional level in Go, providing hope that human-level performance can now be achieved in other seemingly intractable artificial intelligence domains.