Research Review

Sunil Thakur

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

The goal of the research was to play the game of Go at expert level. Go is an adversarial board game with 19x19 grid. The problem was challenging due to its enormous search space. The general game search tree contains approximately bd 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≈35, d≈80) and Go (b≈250, d≈150), exhaustive search is infeasible, but the effective search space can be reduced by two general principles. First, the depth of the search may be reduced by position evaluation. Second, the breadth of the search may be reduced by sampling actions from a policy p(a|s) that is a probability distribution over possible moves a in position s.

The strongest current Go programs are based on MCTS, enhanced by policies that are trained to predict human expert moves.

The novel approach taken by the DeepMind team was to use deep neural networks to reduce the effective depth and breadth of the search tree: evaluating positions using a value network, and sampling actions using a policy network. Board positions are passed as a 19×19 image and convolutional layers are used to construct a representation of the position.

The neural network-training pipeline consists of many machine-learning stages.

1. **Supervised learning (SL) policy network:**

   This provides fast, efficient learning updates with immediate feedback and high quality gradients.
2. **Reinforcement learning (RL) policy network**

   This network improves the SL policy network by optimizing the outcome of games of self-play. This adjusts the policy towards the correct goal of winning games, rather than maximizing predictive accuracy.
3. **MCTS Algorithm:**

   Train a value network that predicts the winner of games played by the RL policy network against itself. AlphaGo combines the policy and value networks in an MCTS algorithm that selects actions by look ahead search.

Results

AlphaGo evaluated thousands of times fewer positions than Deep Blue did in its chess match against Kasparov; compensating by selecting those positions more intelligently, using the policy network, and evaluating them more precisely, using the value network—an approach that is perhaps closer to how humans play. AlphaGo also had 99.8% winning rate against other computer Go programs, demonstrating its dominance.

DeepMind’s research also revealed the level of computational power required to conquer such a task. The final version of AlphaGo used 40 search threads, 48 CPUs, and 8 GPUs. Though a distributed version with 40 search threads, 1,202 CPUs, and 176 GPUs was also implemented,

DeepMind’s research has provided hope that, by similarly leveraging AlphaGo’s novel techniques, human-level performance can be achieved in artificial intelligence domains that were also previously seen as currently unconquerable. 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.

References:

(1) Mastering the game of Go with deep neural networks and tree search, Nature.
https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf