Research Review

Mastering The game of Go with Deep Neural Network and Tree Search

AlphaGo uses value Network to evaluate board positions and policy network. These neural network are trained by novel combination of supervised learning from human expert games, and reinforcement learning from games of self-play.

Alpha Go Goal:

AlphaGo uses **neural networks** and **Monte Carlo Tree Search (MCTS)**. Three policy neural networks are used to decide which moves to investigate and whichones to play. They were trained to identify promising moves from a 19x19 image of the Go game board. A fourth neural network, called the value network,looks at one of those images and assigns a "goodness" value to the current player position (the equivalent of the evaluation function in our Isolation gameplaying agent). MCTS uses the four networks to evaluate the value of each game position in the search tree and identify the most promising move.

Techniques Introduced:

- Supervised Learning (SL) policy network:

Is a 13-layer deep CNN trained on 30 million Go game positions. Given a game position, it predicts the next move, It alternates convolutional layers followed by ReLU activations and is capped by a huge softmax that allocates a probability to each legal move.

- Reinforcement Learning (RL) policy network:

The second stage of the training pipeline improved the policy network by policy gradient reinforcement learning. The games were played between the then current policy network and a randomly selected previous iteration of the policy network. Randomizing from a pool of opponents in this way stabilized the training by preventing overfitting to the then current policy.

- Fast Rollout (FR) policy network:

Like the SL network it was trained to predict the next move, but it is a thousand times faster than the SL network. While not as accurate, but because it is so much faster, it is used to play out the rest of the game, hence, predicting the most likely outcome following the predicted next move.

- Value network:

Estimates the probability that the current position will lead to a win or a loss for the current player. When it was first trained on the samedata than the SL policy network, it severely overfitted. To improve its ability to generalize, the AlphaGo Team trained it on the games collected during the during reinforcement learning phase instead (~30M human games vs 1.5B self-play games).

Results:

A human professional Go player was defeated by a computer program (AlphaGo) for the first time in history and decades in advance of public expectation. Oft described as the last frontier for artificial intelligence in conquering full information games, a program was created that plays Go at the level of the strongest human players.
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, the program's competitiveness in terms of Elo rating exhibited diminishing returns. 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

References

[1] Mastering the game of Go with deep neural networks and tree search, by David Silver et al @ https://storage.googleapis.com/deepmindmedia/ alphago/AlphaGoNaturePaper.pdf