Project 2:

AlphaGo Research Review

PAPER

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

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham2 , Nal Kalchbrenner, Ilya Sutskever2 , Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel & Demis Hassabis

Nature **529**, 484–489 (28 January 2016) | doi:10.1038/nature16961
Received 11 November 2015 | Accepted 05 January 2016 | Published online 27 January 2016

https://www.nature.com/nature/journal/v529/n7587/full/nature16961.html

https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf

GOALS & TECHNIQUES

This paper describes the approach the Deep Mind team has taken in developing AlphaGo and testing it against the best Go players in the world.

Of all the classic perfect-information games, Go is thought to be the most challenging for developing a computer agent/player. Unlike games like Checkers, exhaustive search is not feasible in Go given the complexity of the game which creates an incredibly high branching factor — i.e. the possible number of move sequences. This branching factor is represented by b^d where b is the number of legal moves per board position, and d is the length of the game ($\approx 250^{150}$). The search space for Chess is much smaller by comparison ($\approx 35^{80}$).

Prior Techniques

Prior to AlphaGo, techniques for developing a computer agent were unable to achieve expert level play in Go because they were not able to traverse this vast search space. Depth-first search using minimax and alpha–beta pruning has exceeded human level performance in chess and checkers, but it is intractable in Go. And, breadth-first search techniques have been largely ineffective in Go due to overly shallow policies or value functions, limited to linear combinations of input features. However, some progress had been made using Monte Carlo rollouts to estimate the value of each state in the search tree. Monte Carlo tree search (MCTS) with policy networks trained on human expert moves effectively narrows the search space to small set of high-probability actions. This improves the quality of play to a strong amateur level, but it's failed to produce expert-level performance.

New Techniques Applied by DeepMind

One of the novel aspects of the AlphaGo architecture is the utilization of Convolutional Neural Networks (CNN) to reduce the breadth and depth of the search space. The CNN constructs a representation of board positions board based on 19x19 images passed into it. The board positions are then evaluated using a value network, and actions are sampled using a policy network.

The policy network is used to select moves. It is comprised of supervised learning (SL) policy network that is trained on human expert moves. This data is also used to train a fast policy that can rapidly sample actions during rollouts, as had previously been done by other Go program developers. However, the Deep Mind team uses novel techniques to optimize action selection using MCTS, which is informed both by the search tree and the policy network.

The other major component of the policy network is a reinforcement learning (RL) policy, which outputs a probability distribution of legal moves based on a given board position. The key advantage of the RL policy is that it focuses on game outcomes. In other words, it improves the SL policy by optimizing the network for winning games, rather than maximizing predictive accuracy of the next board state.

The value network is then used to evaluate these board states. It does this by predicting outcomes of games that the RL policy network plays against previous versions of itself. The value network is trained by regression and outputs a scalar value that predicts the expected outcome of board positions in these "self-play" games.

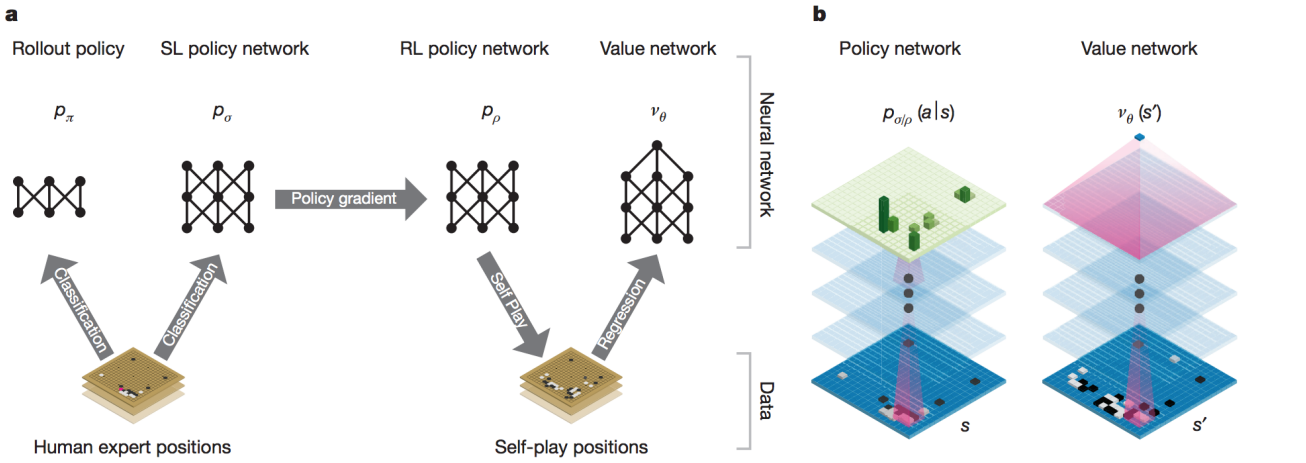

Figure 1 from https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf

RESULTS

AlphaGo emerged as the strongest Go program to date. It won 99.8% of its games (494 out of 495) against the world's top Go programs. Even when giving opponents a 4-move handicap, AlphaGo won 77% of its games against the next best program, Crazy Stone.

Testing variants of AlphaGo revealed that it still outperformed other Go programs even when it played without rollouts. This demonstrates that value networks are a reasonable alternative to MCTS for evaluating positions. However, the fact that AlphaGo performed best when utilizing both MCTS and the value network suggests that the two mechanisms complement one another — the value network is powerful but slow, whereas MCTS is much faster but comparatively weak.

The capstone came in Oct 2015, when AlphaGo became the first computer Go program to beat a professional human player. At the time, Fan Hui was the reigning 3-time European Go champion. AlphaGo won 5-0 in the formal match, and 3-2 in an informal match which had shorter time controls. In turn, the Deep Mind team achieved a milestone predicted not be reached for another 10 years.