AlphaGo Paper summary

The paper introduces a novel approach to computer GO for a problem that was expected to be solved in at least a decade or more. It uses two approaches to (1) give a value to the current state of the board and (2) to define the next moves: “value networks” and “policy networks” respectively and combines them with a powerful searching technique that replaces minimax-AB pruning techniques used in chess/checkers in the past: Monte Carlo Tree Search. The policy and value networks are deep neural networks trained combining supervised learning using expert knowledge and reinforcement learning as the ones used for games of self-play. MCTS is combined with deep learning to drastically improve the search and the information in every state.

Techniques

Supervised learning of policy networks: alternates between convolutional layers with weights sigma and rectifier nonlinearities. The input to the policy networks is the representation of the board state while the network was trained using stochastic gradient descent on state-action (s,a) pairs to maximize the likelihood of the human move ‘a’ in the selected state ’s’. The dataset used to train these networks was the KGS (Kiseigo Go Server) data set which contains 29.4 million positions from 160000 games played by KGS 6 to 9 dan human players. The data set was split in a test set of 1 million positions and a training one with the remaining 28.4 million positions. Final training took around 3 weeks for 340 million training steps.
Reinforcement learning of policy networks: The goal of this step was to improve the policy networks using policy gradient reinforcement learning. It assumes the same structure of the SL policy networks and inherits the weights obtained during the supervised learning stage. The network plays against previous iterations of itself in a random fashion to avoid overfitting which makes training more stable. The policy networks were trained for 10000 mini batches of 128 games, using 50 GPUs, for one day.
Reinforcement learning of value networks: This stage focus in position evaluation. For this purpose an approximate value function is calculated to validate the strongest next policy (or move). A similar architecture to the one used in the RL policy networks was used. However, the output is a single prediction (the value of the position) instead of a probability distribution. The weights are trained using stochastic gradient descend over state-outcome pairs trying the minimize the MSE (Mean Square Error) between the predicted value and the outcome.
Searching with policy and value networks: The previous deep neural networks (policy and value networks) are combined with a tree search technique to improve the performance of the Go players. The tree technique in this case is Monte Carlo Tree Search (MCTS). Each node (s,a) of the search tree stores an action Q(s,a), visit count N(s,a) and a prior probability P(s,a).Then the tree is traversed by simulation starting from the root. After each simulation the actions Q and visit count N of every node are updated. Since policy and value networks require several order of magnitude more computational power than traditional heuristics, it was used an asynchronous multi-threaded search that executes simulations on CPUs and computes policy and value networks in parallel in GPUs (40 search threads, 48 CPUs and 8 GPUs for the single machine version of alphaGo)

Results

AlphaGo was evaluated in a tournament agains other Go programs: commercial Crazy stone, Zen and open source Pachi and Fuego. The single machine AlphaGo version won 494 or 495 games (99.8%). In handicapped matches AlphaGo won 77%, 86% and 99% of the time against Crazy stone, Zen and Pachi respectively. In a series of matches between the AlphaGo single machine and distributed (40 search threads, 1202 CPUs and 176 GPUs) versions the distributed version won 77% of the time and 100% versus other programs. The biggest highlight tough was the match between AlphaGo and a professional 2 dan human player, Fan Hui. AlphaGo won 5-0, which was the first time a computer Go program won against a human professional without handicap.