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

Research Problem

How to effectively reduce the search space of Go to achieve human-expert-level performance.

Guiding principles for search space reduction:
1) Reduction of search depth by position evaluation
2) Reduction of search breadth by move selection using action sampling

Approach

Use of deep convolutional neural networks using a 19x19 board image as input.

Implementation of search space reduction guiding principles using:
1) Value networks for position evaluation
2) Policy networks for move selection

Incorporation of reinforcement learning as well as supervised learning.

Combination of Monte Carlo simulation and tree search with value and policy networks.

Architecture

1) Rollout policy to rapidly sample actions

2) Supervised Learning (SL) policy network trained on expert human moves
   * 13-layer network alternating between convolutional layers and rectifier nonlinearities
   * Final softmax output layer outputting a probability distribution over all legal moves
   * Trained on randomly-selected state-action pairs *(s,a)* using stochastic gradient ascent
   * Maximization of the likelihood of selection of action *a* at state *s*

3) Reinforcement Learning (RL) policy network that optimizes the final outcome of self-play games generated using Monte Carlo simulation
   * Weights initialized to same values as for SL policy network
   * Prevention of overfitting via use of randomized pool of opponents
   * Reward function: i) 0 for all non-terminal steps; ii) +1/-1 for terminal outcome
   * Update of weights at each time step *t* via stochastic gradient ascent
   * Maximization of the expected outcome

4) Reinforcement learning (RL) value network that predicts the expected outcome
   * Single-value prediction of the outcome from position *s* by using policy *p*
   * Estimation of the value function using the RL policy network
   * Trained via regression on state-outcome pairs *(s,z)* using stochastic gradient descent
   * Minimization of the mean squared error (MSE) between prediction and outcome

Search Methodology

Combination of the policy and value networks in a Monte Carlo Tree Search (MCTS) algorithm that selects actions by lookahead search.
* Each node storing action value *Q(s,a)*, visit count *N(s,a)*, and prior probability *P(s,a)*
* a) Selection: Selection of edge with maximum action value *Q* plus bonus *u(P)* that depends on stored probability *P*
* b) Expansion: Expansion and processing of leaf node by SL policy network, with the output probabilities stored as prior probabilities *P* for each action
* c) Evaluation: Evaluation of leaf node i) by using RL value network and ii) by using rollout policy
* d) Backup: Update of action values *Q* with the mean of all evaluations in the subtree below the given action

Evaluation Setup

Tournaments among variants of AlphaGo, against other Go programs (mostly based on MCTS algorithms), and against human European Go champion.

Results

Single-machine AlphaGo version:
* 99.7% winning rate against other Go programs in regular tournament
* 77%, 86%, 99% winning rates against Crazy Sone, Zen, and Pachi, in handicapped version

Distributed AlphaGo version:
* 77% winning rate against single-machine AlphaGo
* 100% winning rate against other Go programs

Comparison among different combinations of rollouts, value network, and policy network:
* Combination of all 3 components, i.e., rollouts, value network, and policy network, achieving best performance, with >= 95% winning rate against other variants

Match against human champion:
* 5:0 winning rate for distributed AlphaGo

Contributions

Deep neural networks trained by a combination of supervised and reinforcement learning.
Scalable, high-performance search engine that integrates neural network evaluations with Monte Carlo rollouts.
Search evaluation of fewer positions (vs. Deep Blue) via more intelligent move selection using the policy network and more precise position evaluation using the value network.
Training based on general-purpose learning methods rather than handcrafted evaluation function (vs. Deep Blue).