Part 2: Research Review:

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

Typically the board on which the game Go is played has a size of 19x19. This leads to an average branching factor b of 250 (Chess ~ 35) and an average game length d of 150 (Chess ~80). The complexity of the complete search tree can be approximated with b^d possible sequence of moves which makes an exhaustive search infeasible.

goals or techniques introduced

The innovative approach of the authors is to use several neural networks.
Starting with a supervised learning (SL) policy network trained with moves of human experts. This 13 layer is trained with 30 million samples from the KGS Go Server.
The second is a reinforcement learning (RL) policy network improving the outcome of the SL policy network by optimising the final outcome of games of self - play. With this information the result can be adjusted to winning sequences of moves.
Those values are again improved by a value network which predicts the winner of the results of the previous RL policy network.
These policy and value networks are combined in a Monte Carlo Tree Search algorithm selecting the best actions using lookahead search. The most promising subtrees selected by random sampling are then explored further.

results

A Go agent has been developed by combining SL, RL and value networks playing at the level of the strongest human players. This agent evaluated much fewer positions than deep blue did in its chess match agains Kasparov. Furthermore this agent did not need any handcrafted evaluation functions but just standard machine learning algorithms.