Research review of AlphaGo

Go has long been viewed as a hard problem to solve for computers as it's hard evaluating the space. In fact the game length (b^d) is b=250 and d=150 making the search tree very large. The team was able to find a way to prune the search tree notably using two principles

1) **Position evaluation:** replacing the subtree below the current state by an approximation
2) **Reduce breadth:** the breadth can be reduced by assigning probabilities to the possible moves in the space using Monte Carlo enabling maximum depth branching

The team went a step further and used Neural Networks to evaluate an image of the board to further reduce the depth and breadth of the search tree

The Neural Network was trained using supervised learning (SL) by comparing the performance to expert games which provides fast instant feedback. Further, a Reinforcement Network (RL) is trained which plays against SL to improve the model further. Finally a network is trained using the RL against the SL. This is combined with the Monte Carlo search tree.

Supervised network
This network was trained using 30M samples from a database and achieved 57% accuracy. The network maximized the likelihood that a human state was selected in a given state.

Reinforcement learning
The Reinforcement Network plays games against the previously trained Supervised Network. This network over time won 80% of games against the supervised network.

Reinforcement learning based on value
The last and third step assigns a value to each option at the current state using the learnings from the RL network above. Initially this set suffered by overfitting but the team simulated 30M additional games to overcome this

The team evaluated the performance against other open source Go programs and won in 494 out 495 cases. In the final match against Fan Hui AlphaGo ended up evaluating thousands of times fewer positions that Deep Blue did against Kasparov. However, the evaluated positions were chosen more carefully.