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

This paper describes how the team behind Alpha Go combines the use of novel deep neural networks with a Monte Carlo tree search algorithm to create an isolation game playing agent that outperforms all previous agents and the current European human Go champion.

Go is a game too complicated (250^{150} move sequences) to be solved by an exhaustive search technique within a feasible amount of time. Traditionally tree search algorithms like the Monte Carlo tree search (MCTS) reduce the search depth and breadth by approximating value functions below a certain depth level and by selecting actions from a predefined policy. These techniques let the search algorithm act as a beam search and allows the agents to achieve a good amateur level of Go play.

The introduction of deep convolutional neural networks led to a significant improvement in performance for the Go agents. Neurons in convolutional layers are used to create a representation of the game board state and the depth and breadth of the search are reduced by using a set of value and policy networks to evaluate board positions and sample actions.
The policy networks that are being used to select actions are created in a two-step process. Initially the network is trained using supervised learning (SL network) with a large set of moves from expert human players. The network is able to correctly predict the moves 55.7% of the time which is a big improvement over previous state-of-the-art traditional method results of 44.4%. To avoid overfitting to these pre-defined moves, a second step includes reinforcement learning (RL network) by utilising self-play games between the current network and a random previous iteration of the network. A network extended in this way has a win rate of 80% against the SL network and a win rate of 85% against leading search-based programs.
The value network is used to estimate a value function that can predict the outcome of the game from the current board position. It is very similar in structure to the policy network. Since successive positions on the board are strongly correlated, a network based on just learning from complete game data leads to overfitting since it tends to simply memorise the game outcome from board position to board position. To avoid this, extensive self-play data of the RL network against itself is included in the training process. This significantly reduces the overfitting and leads to a game outcome prediction error that is consistently lower than those of search-based methods.
For the searching, the policy and value networks are combined with a MCTS algorithm. The values in the tree are defined by using a combination of the value network and a prior probability. Initially the choice of moves depends highly on the prior probability but over time those moves with high action values are preferred.

When just using the neural networks, without combining it with the MCTS rollout scheme, the agent already outperforms all other available Go programs. The combination of the two brings another leap in performance, beating the human European Go champion 5-0 in an official match. Even though Alpha Go evaluates less positions than a traditional search based algorithm, it selects them more intelligently using policy networks and evaluates them more precisely using value networks and thus perhaps getting closer to the way humans play the game.