Research summary: Mastering the game of Go with deep neural networks and tree search by Silver, Huang et al.

Silver, Huang et al., describe the use of deep neural networks in creating a computer agent to play 'Go', one of the most challenging games for the artificial intelligence community due to its enormous search space and the difficulty of evaluating board positions and moves. Without any look ahead search, these networks can play at the level of cutting edge Monte Carlo Tree Search (MCTS) algorithms. Combined with MCTS algorithms, these networks are capable of playing at professional human levels.

Cutting edge Go programs are based on MCTS algorithms supplemented with policies trained to predict human expert moves in order to narrow Monte Carlo search space. These policies tend to be quite shallow and Silver, Huang et al. extends this with deep convolutional neural networks trained using both human expert played games (supervised learning) and self-played games (reinforcement learning). The steps are described below:

1. Training a supervised learning (SL) policy network directly from 30 million positions (state-action pairs from KGS Go server) to predict from a simple representation of board state a probability distribution representing the chance a human expert player will select a move from all legal moves. This was a 13-layer policy network that predicted moves with 55.7% accuracy, 11.3% better than other state of the art research groups.
2. Training a fast policy that can rapidly sample actions during rollouts using a linear softmax of small pattern features with 24.2% accuracy using 2μs rather than 3ms.
3. Training a reinforcement learning (RL) policy network that improves the SL policy network by optimising the final outcome of games of self-play. The RL begins with the same structure as the SL and plays against a random previous iteration of the policy network using stochastic gradient ascent to maximise a terminal reward function. This managed to beat the strongest-open source Go program 85% of the time, a 74% improvement over other state of the art neural networks which were only based on supervised learning of convolutional networks.
4. Training a value network that predicts the winner of games played by the RL policy against the network itself. Training is on a self-played data set containing 30 million distinct positions from separate games. The resulting neural network is constantly able to out predict Monte Carlo rollouts using fast policy and compete with Monte Carlo rollout using the RL policy network using 15000 times less compute.
5. Combining the policy and value networks in an MCTS algorithm that selects actions by lookahead search. Simulations are run and actions are chosen to maximize action value and exploration is encouraged. Leaf nodes are evaluated using a linear combination of the value network (4) and the outcome of a random rollout played out using the fast rollout policy (2). Once the search is complete, the most visited move from the root position is chosen. The above algorithm running on a single machine is able to win 494 out of 495 games against other computer agents. Running on a distributed network, this algorithm is the first to defeat a human professional player without a handicap in a full game of Go.

This is the first time deep neural networks trained on a combination of general-purpose supervised and reinforcement learning methods are used as evaluation functions for move selection and position evaluation functions for Go. With the introduction of a new search algorithm that combines these deep neural networks with tree search from Monte Carlo rollouts, AlphaGo is now able to play Go at the level of the strongest human players, a major breakthrough in computer Go and the artificial intelligence domain.