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

Stephan Kurpjuweit

Goals

The paper discusses a new approach to computer Go (AlphaGo). Go has long been viewed as the most challenging classic game for artificial intelligence.

Approach and techniques

State-of-the-art: Computer-based Go programs rely on Monte Carlo tree search (MCTS), to estimate the value of each state in a tree search. However, this approach is limited (only shallow policies or value functions based on linear combinations of input features) and has only led to strong amateur play.

New approach: The new approach described in the paper uses deep neural networks: *value networks* to evaluate board positions and *policy networks* to select moves. These convolutional networks construct a representation of the position based on a 19 x 19 image of the board.

The neural networks are trained by a combination of supervised learning (from human expert games) and reinforcement learning (from games of self-play) using a machine learning pipeline with multiple stages:

1) Supervised learning of policy networks: The 13-layer SL policy network predicts expert moves and outputs a probability distribution over all legal moves. To train the network, 30 million positions from the KGS Go server were used. The prediction accuracy of the network is 55.7% (compared to the state-of-the-art of 44.4%).
2) Reinforcement learning of policy network: The RL policy network has an identical structure to the SL network and is initialized to the same weights. It aims at improving the policy network. To prevent overfitting to the policy, games between the current network and a randomly selected previous iteration are played and the weights are adjusted. The RL policy network wins more than 80% of the games against the SL policy network.
3) Reinforcement learning of value networks: The value network focuses on position evaluation. The network output is a single prediction instead of a probability distribution. The network estimates the value function for the RL policy network. To generate the training data consisting of 30 million distinct positions, the RL network plays against itself. The training data must be sampled from multiple games, because successive positions in a game are strongly correlated and without the sampling step, the network would memorize entire games (leading to overfitting).

To select a move during game play, a new search algorithm has been developed that successfully combines neural network evaluations with Monte Carlo rollouts. The algorithm selects actions by lookahead search. To drive this algorithm, the final version of AlphaGo utilized 40 search threads, 48 CPUs (to execute simulations) and 8 GPUs (to compute policy and value networks) or 1202 CPUs and 176 GPUs in a distributed version which exploited multiple machines.

Results

The new approach achieved a 99.8% winning rate against other Go programs and defeated a human Go champion by 5 games to 0. The value network alone reaches the accuracy of Monte Carlo rollout, but uses 15000 times less computation.