AlphaGo Research Review

The DeepMind team designed an artificial intelligence named AlphaGo for playing the game of Go. It far exceeded all previous AI Go-playing programs and defeated the European human Go champion, Fan Hui, making it the first AI Go player to defeat a professional human player in a full game of Go, without handicap. Before AlphaGo, all state-of-the-art Go-playing programs were based on Monte Carlo Tree Search (MCTS), whereas AlphaGo successfully couples deep learning networks and MCTS to exceed all predecessors. AlphaGo not only achieved a 99.8% win rate against all other Go-playing programs, but it still exceeded the performance of other programs with its lookahead search disabled. The new technique deployed in AlphaGo entails first learning a policy network using supervised learning on a database of expert human moves. A reinforcement learning policy network is then trained in order to move the policy away from predictive accuracy and toward winning the game. One more network is trained, a reinforcement learning value network, which is then used in combination with the policy network to select actions based on an MCTS lookahead search.

The supervised learning policy network consisted of 13 layers and achieved 57% accuracy in predicting moves by expert human players from 30 million positions, using all feature inputs, with the best predecessor achieving about 44.4%. As accuracy rose from 50 to 57%, the win rate increased from less than 10% to 20 to 40%, depending on the number of filters used, demonstrating that small increases in accuracy led to more drastic increases in win rate. The reinforcement learning policy network that was subsequently trained achieved an over 80% win rate against the supervised learning policy network. When pitted against the MCTS Go-playing program Pachi, the reinforcement learning policy network won 85% of the time, without using lookahead search, with the best convolutional network previously performing at an 11% win rate against Pachi. In the lookahead search, an MCTS algorithm is used to select actions, in which leaf nodes are evaluated based on two criteria that are combined with a mixing parameter: 1) the value of the state as provided by the reinforcement learning value network, and 2) the outcome of a random rollout played until a terminal step, using the fast rollout policy. For the rollout policy, DeepMind employed another new technique, a generalization of the ‘last good reply’ heuristic, in which all moves from the search tree are cached, with similar moves being played during rollouts. In each step of the rollout, if the pattern context matches one in the cache, then the move mapped to it will be played with high probability.

AlphaGo depends on an asynchronous multi-threaded search, with simulations being carried out on CPUs and policy and value networks executed in parallel on GPUs. AlphaGo was implemented on a single machine, as well as being implemented in a distributed manner, across multiple machines. The distributed version of AlphaGo is the one that competed against the human professional Go player, Fan Hui. After achieving a 100% win rate against other Go programs, the distributed version of AlphaGo achieved 5 wins and 0 losses in 5 formal matches. With Go being considered intractable for AI in the past, having AlphaGo play well enough to defeat a human player is a huge step forward for the field of Artificial Intelligence.