Research Review
Mastering the game of Go with deep neural networks and tree search

Rafael Correia Nascimento

In the paper the authors introduces a new novel approach to playing the game Go, AlphaGo. The game of Go has long been viewed as the most challenging of classic games for artificial intelligence owing to its enormous search space and the difficulty of evaluating board positions and moves [from the original article]. Theirs approach uses Deep Neural Networks, 'value networks' and 'policy networks', combining supervised learning and reinforcement learning. The authors also introduces a new search algorithm combining Monte Carlo simulation with value and policy networks that achieved 99.8% winning against other Go programs and were able to defeat the human European Go champion by 5 games to 0. This were the first time a computer defeated a human professional player creating a new mark in the AI field.

Like in isolation, the game of Go could be solved by recursively computing the optimal value function in a search tree, containing approximately b^d possible sequences of moves, where b is the game's breadth (number of legal moves per position) and d is its depth (game length). However in the game of Go b ~ 250 and d ~ 150 what makes exhaustive search infeasible. Two general approaches to reduce the space are (1) position evaluation to reduce depth and (2) sampling actions from a policy to reduce breadth. Inspired by the recent good results of deep convolutional neural networks (CNN) in the computer vision field. The authors passed a 19x19 image of the board to a CNN to construct a representation of the positions and to reduce depth and breadth of the search tree.

The authors first trained a supervised learning (SL) policy network to predict expert moves. This SL policy network used a 13-layer CNN with rectified nonlinearities. The network were trained on randomly sampled (30 million positions from KGS Go Server) state-action pairs, using stochastic gradient ascent. The network had maximum accuracy of 57.0% using all input features. To improve the results of the policy network, the authors trained the policy network by policy gradient reinforcement learning (RL). This was done playing games against randomly selected previous iteration of the policy network. The reward function was zero for all non-terminal time steps and +1 for the terminal wining state (perspective of current player) and -1 for losing. The weights of the network were then updated at each time step using stochastic gradient ascent. The RL policy network were able to win more than 80% of games against the SL policy network. Also, it won 85% of games against Pachi (strongest open-source Go program).

The final stage of training was the value network that estimate the value function for theirs strongest policy, using the RL policy network, to evaluate positions. This value network has a similar architecture than the policy network but outputs only one prediction. The weights were trained using regression on state-outcome (win or loss), using stochastic gradient descent. To reduce overfitting the authors generated a new self-play data set consisting of 30 million distinct position, each sampled from a separate game. The mean squared error (MSE) was 0.226 on the training data and 0.234 on the testing data.

AlphaGo combines the policy and value networks in an Monte Carlo tree search algorithm (MCTS). The tree is traversed by simulation, starting from the root state. Once the search is completed, the algorithm chooses the most visited move from the root position. AlphaGo uses an asynchronous multi-threaded search that executes simulations on CPUs, and computes policy and value networks in parallel on GPUs.

AlphaGo was evaluated in a tournament against other Go programs, including the strongest commercial programs Crazy Stone and Zen, and the strongest open source programs Pachi and Fuego. The program GnuGo was also included. All programs had 5 seconds per move. AlphaGo won 494 out of 495 games (99.8%). AlphaGo was also evaluated against Fan Hui, a professional Go player, winner of the 2013, 2014 and 2015 European Go championship. Alpha Go defeated Fan Hui 5 games to 0. This was the first time a computer Go program defeated a human professional.