Karolis Jankauskas, March 2017

Review of "Mastering the game of Go with deep neural networks and tree search" by Silver, Huang et al. (2016)[1]

The enormous search space, the difficulty of evaluating board positions and moves of the game of Go have made it very challenging to apply the classical artificial intelligence methods to solve it. Nevertheless, Silver, Huang et al. (2016) presented a new approach based on multi-stage machine learning and a state-of-the-art tree search algorithm that enabled their program, AlphaGo, achieve a 99.8% winning rate against other Go programs and defeat the human European Go champion by 5 games to 0. AlphaGo also defeated the top Go player in the world in March 2016, Seoul, South Korea. Such achievement was thought to be at least a decade away.

The key to the approach presented by Silver, Huang et al. (2016) is a combination of policy and value neural networks, and Monte Carlo tree search. A policy network outputs a probability distribution predicting the next move whereas a value network approximates the outcome of the move. Both types of networks are used to guide the Monte Carlo tree search: a policy network helps to reduce the breadth while a value network reduces the depth of the search. AlphaGo selects the move that is most successful in simulation.

The training of deep neural networks consists of several stages. During the first stage, a supervised learning (SL) policy convolutional neural network is trained directly from expert human moves. The SL policy network predicted the expert move 57% of the time using only a raw board position. For the second stage, the SL policy network is improved using the policy gradient reinforcement learning (RL). The RL policy network alone (without the help of the tree search) won 80% of games against the SL policy network and 85% of games against Pachi, an open-source, Monte Carlo search based Go program. The final stage focuses on position evaluation, estimating (by regression) a value function that predicts the outcome from position s of games played by using policy p for both players. The policy networks are used to train the value networks by RL from games of self-play. The value network is trained by using stochastic gradient descent to minimize the mean squared error between the predicted value and the corresponding outcome of the position.

For the efficient combination of the deep neural networks with Monte Carlo tree search, the team of AlphaGo used an asynchronous multi-threaded search (40 threads) that executed simulations on CPUs (48), and computed policy and value networks in parallel on GPUs (8). A distributed version of AlphaGo was also implemented by exploiting multiple machines, i.e. 1,202 CPUs and 176 GPUs. The distributed version of AlphaGo won 77% of games against the single-machine version and 100% of its games against other programs.

---

[1] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484-489.