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

The ancestral game of Go has been considered as a very challenging one in the real of Artificial Intelligence, because of it huge search space, as well as, the difficulty to evaluate the board's moves and positions. The authors of this research article have introduced new approaches to address this issue by using "value networks" in order to evaluate positions and "policy networks" to select moves. Then, these neural networks are trained by combining supervised learning, which originate from human expert games with reinforcement learning from self-play games. Lookahead search is not used. Instead, neural networks play Go at the level of state-of-art Monte Carlo tree search programs. They simulate thousands of new self-play games. The authors also introduce a new searching algorithm, which combines policy and value networks, with Monte Carlo simulations. Therefore, this approach has allowed AlphaGo to reach a 99.8% winning rate against other Go programs. Furthermore, it has defeated the human European Go champion by 5 games to 0. As a result, it is the first time a computer program has won a human professional in a full-sized Go game. The following paragraphs describe these new contributions and outcomes in more detail.

Supervised learning of policy networks

During the first part of the training stage, the authors built on prior work on predicting expect moves in the game of Go using supervised learning. This SL policy network alternates between convolutional layers with weights and rectifier nonlinearities. Then, a final softmax layer outputs a probability distribution over all legal moves. The input to the policy network is a simple representation of the board state. The policy network is trained on randomly. A fast rollout policy and supervised learning policy network are trained to predict human expert moves in a data set of positions.

Reinforcement learning of policy networks

This stage of the pipeline is focused on improving the policy network by policy gradient reinforcement learning. The RL policy network has the same structure as the SL policy network. The authors played games between the current policy network and a randomly selected previous iteration of the policy network. Overfitting prevention is done by randomizing the opponents pool. A reward function, which is zero is used for all non-terminal time steps. The outcome is the terminal reward at the end of the game from the perspective of the current player at time step t: +1 for winning and -1 for losing. Moreover, weights are then updated at each time the step t by stochastic gradient ascent in the direction that maximizes the expected outcome.

Reinforcement learning of value networks

The final stage of the training pipeline is focused on position evaluation. This estimates a value function, which predicts the outcome from a certain position, based on games played using a specific policy for both players.

AlphaGo's strength evaluation

AlphaGo was evaluated by running an internal tournament among different AlphaGo variants and many other Go programs. All of these programs are based on high performance Monte Carlo tree searches. Also, the authors included open source programs; such as, GnuGo. All of these programs were allowed 5 seconds of computation time per move. The results of this tournament suggest that single machine AlphaGo is many dan ranks stronger than previous Go programs. In fact, the distributed version of AlphaGo won 77% of games against single-machine AlphaGo and 100% of its games against other programs.

Bibliography

Hassabis, Demis, Mastering the game of Go with deep neural networks and tree search, 2016