AlphaGo Article Summary
Masa Watanabe

AlphaGo is the most advanced Go program developed as it defeated the European Go Champion 5 out of 5 times, becoming the first computer program to beat a professional human player in the game. Played on a 19 x 19 board with limited restrictions on each player's moves, Go has a significantly high number of potential moves attesting to the game's complexity. The program uses a combination of deep neural networks and Monte Carlo tree search (MCTS) to increase efficiency and accuracy of its predecessor programs which have been unable to overcome the issue of evaluating the vast positions in a timely manner.

The program takes advantage of the recent advancements in understanding vision, passing images of the board and producing convolutional layers to represent the players' positions. This neural network technique is able to more efficiently evaluate a position of the search tree. Two networks,policy and value, are trained in three stages. In the first stage, using supervised learning(SL) techniques, a 13-layer policy network was trained with 30 million positions from expert human moves. Using all input features, this increased the accuracy by 11.3% compared to other research groups at the time. The second stage reinforced the policy network (reinforcement learning or RL) and reduced overfitting by playing against random iterations of itself and using stochastic gradient descent in the direction which maximizes the number of times the current policy network won. The RL policy network won 85% of games against Pachi, the strongest open-source Go program. The final stage of the neural network estimated the value function using the RL policy network trained in stage one and two. A value network with weights was trained using stochastic gradient descent, minimizing the mean-squared error between the prediction and actual outcome. To offset overfitting, a data set of 30 million distinct positions was created from samples of self-play games. This led to the program being able to accurately evaluate a position as the original MCTS but using 15,000 times less computation.

The MCTS stores an action value, visit count and prior probability at each edge. The visit count is used to decay the value of the search node so as to encourage evaluation of nodes. The prior probabilities are for each legal action of the leaf node. The value is determined by a combination of the value network and a random rollout played out to the end of game. In order to maintain the mass amount of computation, AlphaGo uses asynchronous multi-threaded search and distributed versions of the program across multiple machines.

AlphaGo's strength was assessed by running a tournament against other high-performing Go programs. The tournament resulted in showing that AlphaGo was significantly better than any previous program, winning all but 1 of 495 games. Even with a four-stone handicap, the program won over 77% of games. AlphaGo passed one of artificial intelligence's "grand challenges" of defeating three-time Go Champion Fan Hui, a feat said to be at least a decade away.

AlphaGo introduced a new method of combining evaluations via neural networks and MCTS. Evaluating fewer positions more accurately, AlphaGo is perhaps the closest program to mirror how humans play the game and even how humans make decisions. Games, like Go, are practical methods of researching the bounds of artificial intelligence and tackling the challenges of understanding the complexity of human decision-making. They result in artificial intelligence programs like AlphaGo which achieved human-level performance and provide hope in achieving the same in other domains.

10/23/2017