Review of AlphaGo

Summary

The win of Google Deep Mind's Go algorithm, AlphaGo [1], against Fan Hui, a 2. dan professional player, has been a major breakthrough in artificial intelligence (AI).
After Deep Blue beat Kasparov in chess in 1996-97 [2], the win of AlphaGo put AI back into the general publics mind. Go was considered very challenging given the complex nature of the game. The huge search space of Go-a 19x19 board has 2.08e170 legal game positions [3], the large amounts of possible moves in every state (=250), makes it a very challenging game. Combined with the simplicity of the rules, which allow for huge freedom, and thus makes it very difficult to come up with good heuristics to approximate the outcome of a game from a given state.

Due to this complexity, algorithms, such as minimax and alpha-beta pruning, proven to be successful in chess or checkers, failed in Go.
Monte Carlo tree search (MCTS) in contrast to minimax has shown to be successful in playing go, here, the value function and the optimal policy is estimated using a huge amount of repeated plays (roll-outs).

Unlike, these earlier methods AlphaGo does not rely on complicated roll-out strategies for the MCTS. AlphaGo uses a clever combination of different techniques from AI and machine learning (ML) to estimate a policy and a value function using deep convolutional neural networks (CNN). At first alpha go uses CNNs to represent the spatial position of the game. Then it uses supervised learning based on an extensive database of human plays to train winning policies, and further refines these policies via reinforcement learning (RL). The resulting policy network is then a mapping from a current state of the game to a probability distribution of moves leading to a win.

In the next steps, the self-play of the policy network via RL to get an approximation for the value function, i.e. the chance of winning of a given state, using regression. Here, in order to avoid the bias introduced by autocorrelation, 30 million games were played, and only one sample from each game was taken. Thereby, the reinforcement learning pipeline was used to its fullest.

In the last step MCTS was used to combined the two networks, using the policy network to guide the strategy and the value network for early stoppage. The two networks were further supplemented by full roll-outs, thereby utilizing the advantage of in-depth knowledge of the game state. The computational framework used to implement this state of the art framework reflects how one can utilize available computational power to its fullest by combining the most advanced techniques in ML and AI.

Result

As a first test, the authors compared AlphaGo against the most advanced Go algorithms. Here, AlphaGo outperformed all of them, interestingly, even without roll-outs, the value network alone outperformed all of the leading AI algorithms.

Next, to comparing AlphaGo against other AI algorithms, the authors matched AlphaGo against human Go champions, here, AlphaGo managed to beat FanHui, a 2 dan professional player, in 5 out of 5 matches. AlphaGo could repeat this success against a human player in its match against LeeSedol, a rank 9 dan, where it has won 4 out of 5 matches. This showed that using the combination of AI and ML allowed to build algorithms, which repeatedly are able to compete against humans in games complex as Go.

References

[1]: Silver et al., Mastering the game of Go with deep neural networks and tree search., 2016, Nature, 529, 484-489

[2]: https://en.wikipedia.org/wiki/Deep_Blue_versus_Garry_Kasparov , retrieved on 03/16/2017

[3]: http://tromp.github.io/go/legal.html , retrieved on 03/08/2017