Research Review: AlphaGo

Georg Maerz

2018-05-06

Solving Go is a challenging decision-making task, covering an extremely large search space leading to infeasibly direct approximation of a solution.
GOs branching factor of 250 with an approximate depth of 150 demonstrates the scope of the challenge and why GO has been considered one of the major milestones in AI development, for exceeding the complexity of chess with an branching factor of 35 and an approximate depth of 80.

The overall success has been based on the combination of deep neural networks and tree search. For this purpose Google developed a methodology to effectively move to a selection, but also to evaluate the current position itself utilizing a combination of both supervised and reinforcement learning. Furthermore they introduced a new algorithm that successfully combined the evaluation of neural networks with the monte carlo algorithm

Due to the high complexity of the game Go, Alpha Go has been made efficient in selecting from a small amount of positions using the policy network, utilizing its computation on a precise evaluation using the value network. The authors comment, that this might be, how humans themselves actually think, execute and play.

Further human moves are predicted by the use of the generation of new data set by playing games of self-play with the RL policy network. This network has 13-layer for the SL policy network with 30 million positions. One of the game changers have been proved to be the small improvements in playing strength improvement, by just small improvements in accuracy of predicting moves.

Empirically it seemed that to many filters in the policy network are rather bad than good. For the training they take into account, that a quick win is a goal, by using a reward function r(s) for all non-terminal time steps t<T. The final stage is the position evaluation. It predicts the outcome by using a policy for both players.

Alpha Go tackles the issue of overfitting by acknowledging that successive positions are strongly correlated, causing the naïve approach for predicting game outcomes from data consisting of complete games to overfitting. They overcome this by generating new self-play data set. Additionally they encourage the exploration by adding decays in the selection of each time step t of each simulation an action a is selected from state t.