Research Review

MASTERNG THE GAME OF GO WITH DEEP NEURAL NETWORKS AND TREE SEARCH

Roberto Talamas | Artificial Intelligence, Udacity

Goals and Techniques

The paper introduces AlphaGo, a novel implementation of deep neural networks combined with tree search to create a game playing agent capable of superhuman performance in the game of Go.

The game of Go, similar to chess, is a game of perfect information meaning both players are perfectly informed of all the events that previously occurred in addition to the current state of the game. To solve a game of perfect information, an agent has to evaluate all possible legal moves to find the optimal value by recursive simulations. Simulation are represented by the game search tree which contains approximately b^d possible moves, where b is the number of legal moves at each node of the game tree and d is the depth of the search tree. To put things in perspective, the game of chess has roughly values of $b \approx 35$ and $d \approx 80$ (10^{123} possible game states) while Go has values of $b \approx 250$ and $d \approx 150$ (10^{360} different games). Given that the average multi-core gigahertz processor is capable of doing 10^9 operations per second, this means that evaluating all possible moves in Go would take longer than our Universe will exist! ($\frac{10^{360}}{10^9} = 10^{351}\ seconds$).

So how is AlphaGo capable of such superhuman performance? To solve the game complexity problem, the DeepMind team created AlphaGo using two different components: a tree search component and a convolutional neural networks (CNNs) component to guide the tree search procedure. One important aspect of AlphaGo is that the convolutional neural networks serve the same purpose as the evaluation function used by different game playing AI agents such as Deep Blue, the key difference is that they are *learned* and not *handcrafted*. This *learned* evaluation function provides the agent with a level of "intuition" for the game.

In total, three CNNs are trained of two different kinds: two *policy* networks and one *value* network. Both take as input the game state represented as an image.

The policy network was trained on 30 million positions from games played by human experts, available in the KGS Go server. The network was able to achieve an accuracy of 57% accuracy on withheld test-set, which is quite impressive given the complexity of the game. But the goal was not to create a system that is great at predicting human moves but rather to build a network optimized to win the game. Therefore, the policy networks were improved by allowing them to play with each other, using the output of these games as training signals. Then, the value network was trained on 30 million game positions obtained from the games played by the policy network. The function of the value network is to predict the likelihood of winning the game given the current state of the game. This is eerily similar to the purpose of an evaluation function except that it is learned from data instead of designed. Lastly, AlphaGo uses the combination of policy and value networks in Monte Carlo search tree in order to choose the best move.

Results

Performance of AlphaGo is widely known and recognized. To evaluate the strength of program, the team created an internal tournament among different variation of AlphaGo and other Go programs which included the strongest commercial programs at the time. The results suggest that AlphaGo ranks among the strongest programs with a win rate of 99.8% against other Go programs[1]. In 2015 it beat the European Go champion Fan Hui, marking the first time a Go program beat a professional Go player. It is wort pointing out that during the Hui games, AlphaGo evaluated thousands of times fewer positions than Deep Blue did during its match with Kasparov[2] suggesting that positions are selected more intelligently by the policy network and evaluated more accurately by the value network. This ultimate combination of tree search with policy and value networks has enabled AlphaGo to reach professional level status at the game of Go.

References

[1] Mastering the game of Go with deep neural networks and tree search, by David Silver et al @ https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf

[2] Campbell, M., Hoane, A. & Hsu, F. Deep Blue. Artif. Intell. 134, 57–83 (2002).