Artificial Intelligence Nanodegree Alexandru Korotcov
Project: Research Review of a paper on Game-Playing March 15th, 2017

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

David Silver, Aja Huang, Chris J. Maddison, Arthur et al.
Nature 529, 484–489 (28 January 2016)

Goals: One of the most challenging games for artificial intelligence (AI) is the game of GO. The challenges come from a very big search space (depth 150, average plies in a game 30, branching factor 250) and difficulty of evaluating board positions (board size 361) and moves, thus an exhaustive search is infeasible. The effective search space can be reduced by two general principles: position evaluation (truncating the search tree) and sampling actions from a policy (probability distribution). Strongest Go programs are based on Monte Carlo tree search (MCTS) enhanced by policies that are trained to predict human expert moves. The problem is that most of these programs are limited to shallow policies or value functions based on a linear combination of input features. The goal of this work was to use the advantages of convolutional neural networks (NN) and find a way of reducing the effective depth and breadth of the search tree by evaluating positions using a 'value network', and sampling actions using a 'policy network'.

Techniques: Authors train NN using a pipeline consisted of several stages of machine learning. Firstly, a supervised learning (SL) 'policy network' was trained directly from expert human moves providing fast and efficient learning updates with immediate feedback. Second step was to train a reinforcement learning (RL) 'policy network' thus improving SL policy network by optimizing the final outcome of self-play games. Finally, authors trained a 'value network' predicting the winner of games played by the RL policy network against itself. The final program efficiently combined the policy and value networks with MCTS method.

Results: The approach of using combination of deep neural networks and tree search chosen by the authors was used to build a program called AlphaGo. In order to evaluate AlphaGo, authors ran an internal tournament among variants of AlphaGo and several other Go programs, including the strongest commercial programs Crazy Stone and Zen, and the strongest open source programs Pachi and Fuego. The AlphaGo exhibited a very impressive performance (winning 494 out of 495 games, which is 99.8%) and able to play at the level of the strongest human players (program had defeated a human professional player in the full game of Go), thus achieving one of artificial intelligence's "grand challenges". This is the first time in the literature when the effective move selection and position evaluation functions for Go were based on a deep neural networks that were trained by a novel combination of supervised and reinforcement learning. Also, authors introduced a new search algorithm that successfully combines NN evaluations with MCTS, thus turning AlphaGo into a high-performance tree search engine.

Summary: Authors introduced a novel approach to computer Go that uses 'value networks' to evaluate board positions and 'policy networks' to select moves. The deep neural networks trained by a novel combination of supervised learning from human expert games, and reinforcement learning from games of self-play. The new search algorithm that combines Monte Carlo simulation with value and policy networks achieved a 99.8% winning rate against other Go programs, and defeated the human European Go champion by 5 games to 0.