Research Review of the Paper: Mastering the game of Go with deep neural networks and tree search

Joe Minichino

Goals and Techniques introduced

The paper introduces the usage of deep neural networks to create a game playing agent capable of superhuman performance in a game such as Go, which is extremely complex by virtue of its large set of possible moves (the board is a square grid, 19x19), therefore exploring the entire game tree, or even a useful subset of it, is impossible, or prohibitively long.
The authors of the paper suggest an innovative approach that leverages deep neural networks, specifically the usage of three neural networks:

1. A supervised learning (SL) policy network trained with human expert moves
2. A reinforcement learning (RL) policy network that evaluates self-play outcomes of the current state of the game
3. A RL value network that predicts the winner of games played with network number 2

At SL stage, a 13-layer network is trained using image representations of the board, with moves taken from the KGS Go server (30 million samples). This is effectively an approach similar to more famous convolutional neural networks like Google's ImageNet.
The first neural network performs a classification which produces a policy gradient. This second network produces a probability distribution of moves, and passes this into the third neural network which performs a regression to establish the winner of the self-play with the given set of moves.
This approach is used in combination with Monte-Carlo tree search (MCTS), which utilizes random sampling of the tree with evaluation of each game tree branch. At each visit evaluations are stored so as to quantify a "bonus" for each branch that effectively determines the most "promising" moves. The value for each explored node is the mean of an evaluation function over the number of visits on that node.

Paper results

The paper shows that this method has not only obtained a superhuman performance in the game, but also - thanks to the combination of policy and value networks with tree search - that the game agent has evaluated far fewer moves than DeepBlue did in its chess match

against Kasparov, while beating human experts at a game that is far more complex than chess. (cntd.)
Additionally, unlike DeepBlue, evaluation functions in AlphaGo are not handcrafted, rather standard machine learning methods, in particular, stochastic gradient ascent (to maximize the likelihood of winning a game), rectifiers (linear and non-linear), tanh, logistic functions and bayesian logistic regression (see Neural Networks Architecture section and Evaluation section).