Research Review – Alpha Go

Yeshwanth Arcot | 06/14/17

Introduction:

The game of GO is known for it's enormous search space(b≈250, d≈150) when compared to chess(b≈35, d≈80). As exhaustive search is infeasible, computer programs till now relied on Monte Carlo Tree Search(MCTS) boosted by policies that were trained to predict human moves. However, this kind of play could only achieve strong amateur level play. All this changed when AlphaGo(developed at google) beat the European champion Fan Hui 5 games to 0 and 18 time world champion Lee Sedol 4 games to 1.

Working

The board is passed in as a 19x19 image (note that Go is played on a 19X19 grid) and convolutional layers are used to construct a representation of the position. Using a *value network* positions are evaluated and with a *policy network* actions are selected. These deep neural networks are trained by a novel combination of supervised learning from human expert games, and reinforcement learning from games of self play.

ML Pipeline

i) Supervised learning of policy networks:

A simple representation of board state *s* is passed to the policy network which alternates between convolution layers with weights σ and rectifier non linearities. A final softmax layer outputs a probability distribution over all legal moves.

ii) Reinforcement learning of policy networks

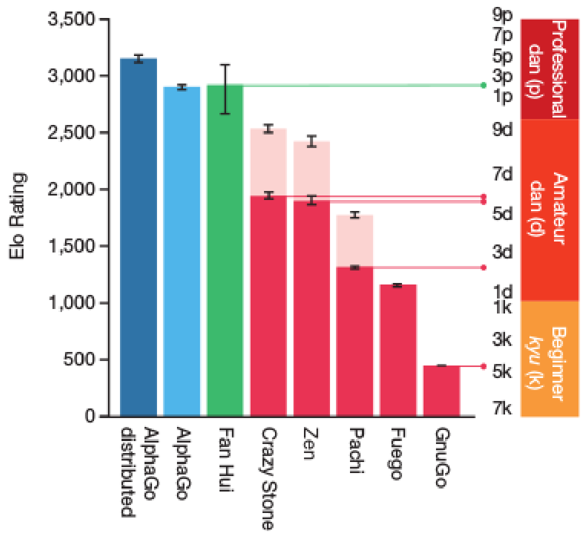
This stage aims at improving the policy network by policy gradient reinforcement learning. It's structure is identical to the SL policy network with its weights ρ same as σ. Games are played between current policy network and a randomly selected previous iteration of policy network. This randomization stabilizes training and prevents overfitting to the current policy.

A reward function that is zero for all non-terminal steps, +1 if the player wins at the terminal step and -1 if the player loses at the terminal step is used. Then weigths are updated at each time step by stochastic gradient ascent in the direction that maximizes expected outcome.

iii) Reinforcement learning of value networks

This stage focuses on position evaluation. Ideally we would like to know the optimal value function under perfect play. But here we instead estimate the value function of the strongest policy, using RL policy network. The value function is approximated using a value network initialized with weights θ. The weights are trained by regression on the state-outcome pairs (*s*, *z*) using stotchastic gradient descent to minimize mean squared error between predicted value and the corresponding outcome.

Results



Results of tournament between different GO programs (Fan Hui is the only human). Programs were evaluated on an Elo scale. A 230 point gap corresponds to 79% probability of winning. 95% Confidence intervals are shown.

Conclusion

During the match against Fan Hui, AlphaGo evaluated thousands of times fewer positions than Deep Blue did in its chess match against Kasparov. An approach that is perhaps closer to how humans play. Furthermore, while Deep Blue relied on a handcrafted evaluation function, the neural networks of AlphaGo are trained directly from gameplay purely through general-purpose supervised and reinforcement learning methods. Full paper available at https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf