Key learnings from reading the article "Mastering the game of Go with deep neural networks and tree search"

https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf

Article Summary

The goal of the article is to illustrate the the ground-breaking technologies and the related concepts used by AlphaGo to achieve an unprecedented winning rate of 99.8% against other prior Go programs.

The article first acknowledged that the huge search space (breadth and depth) of Go make it impossible to do an exhaustive search. Depth reduction by using utility approximation given a certain board state is not applicable to Go because of its complexity. Breadth (branching factors) could also be reduced by using Monte Carlo sampling without having to expand the breadth of search. It was mentioned that the prior best Go algorithm adopted this technology together with human training but the model is shallow linear function.

Technology Used

In AlphaGo, Convolutional Neural Network, which is best known for its performance on image, object, facial recognition is used. This system has 2 major functions/components: evaluating board positions using a value network, and sampling different actions using a policy network.

The neural network has a number of different stages of training:

- Using human expert to provide supervised training data for the policy (model). The softmax output probability of what a human expert would have done given a particular board state. The system is able to predict human expert moves by 57% accuracy.
- Reinforcement Learning that takes the outcome of the environment and correct/fine tune the course of the current policy. The system also used "self-play" games to create different environmental state so that it could learn from it. Winning a game would result in positive reward gradient (reinforcement) to the current model and update it to a better version. In another word, the policy/model keeps updating when the AlphaGo plays the game instead of just using one single policy during the whole game. Using no look-ahead-search at all, this reinforcement learning allow AlphaGo to beat Pachi, the strongest Go agent prior to AlphaGo by a winning rate of 85%.

- With a continuous update on the policy, the system is able to continuously predict the result of the game in run time which helps the system to foresee the final outcome of the game.
- A prediction model is being used to predict the final outcome of the game. The model for prediction is constantly updated to reduce the "mean square error" between the ground truth and the prediction made. The system generate new game for each sampling and hence able to reduce over fitting of the model to training set.
- AlphaGo also utilize an improved version of Monte Carlo Tree Search to same the depth of network to improve the accuracy of its prediction. This MCTS combines both policy and value network. The search used here is still expensive as it used a 48CPUs and 40GPUs for computation. AlphaGo also has a distributed version which utilize even more CPUs and GPUs.

Achievements

AlphaGo is able to beat all other Go programs by a winning rate of 99.8%. Even for free moves game, AlphaGo won 77%, 86%, and 99% of handicap games against Crazy Stone, Zen and Pachi, respectively. AlphaGo defeated Fan Hui a professional Go champion in a game of result 5-0 in 2015.