Alpha Go

Summary

Alpha Go is computer program that uses a combination of deep neural networks and game tree search techniques. Alpha Go could play at the level of strongest human players and could defeat may state of the art Go engines out in the market.

Alpha Go used two neural networks namely policy networks and value networks. Former being used to evaluate board positions and latter to select a move on the board. These neural networks are trained using a combination of supervised learning (SL) from labelled data of human expert games and reinforcement learnings (RL) from self-played games.

In Alpha Go, three convolutional neural networks (CNN) are trained. Two of them are policy networks and one is a value network. The input to these networks are the game positions. Value network predicts a scalar, the probability of the computer player to win the game given the current game position. Policy networks guide Alpha Go to take the best action for a given game position. For each valid move from a given game position, value network outputs a probability value which represents the chance of win if we are taking that move.

Alpha Go employs a pipeline of machine learning stages to achieve its goals. The first stage is a policy network trained using supervised learning on 30 million positions from the games played by human experts. This network alone was giving 57% accuracy.

The goal here is to achieve high percentage of winning games. So another policy network is trained using a reinforced learning technique to improve on the actions learned using the first stage of the pipeline. Alpha Go achieves this by letting the networks play against each other and using the outcome for training.

The third stage is a value network trained on 30 million game positions obtained during the reinforcement learning of policy networks.

Finally Alpha Go combines value and policy networks using a Monty Carlo Tree Search (MCTS) algorithm that selects action using a look ahead search. The tree is traversed using simulation and the best moves are updated along with the prior probability value.

Results

Single node setup of Alpha Go used 40 threads, 48 CPUs and 8 GPUs where as distributed version used 40 threads, 1202 CPUs and 176

GPUs. Alpha Go has been evaluated by conducting a set of tournaments with other popular as well as commercial Go programs based on MCTS algorithms with 5s computation time per move. Single machine Alpha Go won 494 out of 495 games (99.8 %) in the tournament where as distributed version of Alpha go won 100% of the games against other Go programs and 77% of the games against single node Alpha Go. Even without the rollouts, alpha go out run other Go programs. This suggests that value networks can serve as an alternative to Monte Carlo evaluation technique in terms of Go. Finally, a tournament of 5 games has been conducted with a professional Go player having highest rank and won all the games (100%).