AlphaGo: the first AI to achieve the master level in Go

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

A goal of the paper was to present the techniques that were used to create a Go playing AI that was capable of beating all existing Go AIs and even master level human players. It was a feat that at the time was considered to be not possible for at least another decade.

Go is similar to chess in a way that both are two player board games with each player taking turns and no element of randomness is present during the game. However, differently from chess, Go has a much larger number of possible game states and more actions available in each state. The combination of these factors made the Go playing AI that performed at the master level hard to attain. The AIs before AlphaGo could only achieve amature level play.

Games like Go and Chess involve search techniques that build game state trees and then search for the best moves by using various traversing optimization techniques. Since evaluating a full state tree is impractical, the game tree is iterated up to a certain level and then the state is evaluated with a heuristic that is usually built over time via research and game analysis that can accurately interpret the usefulness of the board for a player.

Move selection and position evaluation is a very critical piece in the successful game playing AIs. This is where AlphaGo differed from the other Go AIs by introducing a novel method for game state evaluation and effective move selection.

It combined a learned value and policy networks with Monte Carlo tree search (MCTS) rollouts for a very effective search algorithm to traverse the game state tree. The effectiveness of the networks came from it first being built using supervised learning by classifying and learning from human expert Go games but then it followed that by reinforced learning from self-play. Instead of learning to predict expert human moves like previous AIs did, it learned the moves it should use from the self-play. Reinforcement-learned value networks strengthened the network produced from supervised learning which led to its dominance against existing Go AIs and eventually a human expert players.

Evaluating policy and value networks was very computationally intensive operation, much more so than compared to a more traditional search heuristics. To accomplish this in a reasonable time, policy and value networks were evaluated on GPUs while in parallel MCTS simulations were run on the CPUs. The AI also could be configured to run in a distributed configuration across multiple machines where it achieves even more impressive win rates and Elo ratings.

Event without using any search and relying on just reinforced learned networks, AlphaGo was able to defeat the most sophisticated Go AIs of that time 85% of the time. Eventually the researchers measured its effectiveness using just Monte Carlo tree search, just the policy network, just the evaluation network, and then the combinations of the three. The most effective results came from all three being used in playing the game where it defeated all the existing AIs by a wide margin (99.8% winning rate). And eventually it participated in a highly publicized match against Lee Sedol, winning by the count of 4-1 and cementing its place in AI history forever.