Research Review:
Mastering the game of Go with Deep Neural Networks and Tree Search

Rizwaan Adil

Summary

Go is a classic 2 player strategy board game where players try to surround each other while capturing territories on the board. Due to a high branching factor, enormous search space and difficulties in evaluating positions, Go was considered highly complex for AI agents. This research paper presents unique approach to overcome this complexity in Go gameplay. The approach outlines a search algorithm which combines Monte Carlo simulation with deep learning and reinforcement learning networks.

Techniques

The key to AlphaGo's success is the unique combination of value & policy networks and Monte Carlo simulation. Value networks are used to evaluate board positions and policy networks for move selection. Human expert games are used for training the value networks and reinforcement learning is employed for games of self-play.

The approach manifests in the form of a pipeline of machine learning stages as below

Step 1: Policy Network Training

A 13 layer policy network (SL) was trained on randomly sampled state-action pairs from 30 million position from KGS Go server (a popular online game host). This network produced the probability of each move being the actual next move on the board during gameplay.

While being only 55% accurate on held out test sets, the SL policy network was comparatively more accurate than the then state-of-the-art result of 44.4%.

Step 2: Reinforcement Learning for Policy Network

To further improve the SL policy network, another policy network (RL) was constructed with the same 13 layers and even the same initial weights. Overfitting was prevented by randomizing from a pool of opponents (one of the previous RL iterations) and weights were adjusted using stochastic gradient ascent.

In head to head games with the SL policy network, RL network posted a handsome 80% win rate. RL policy network also defeated Pachi, the strongest open source Go player backed by a sophisticated Monte Carlo search algorithm, winning 85% games without using search

Step 3: Reinforcement Learning of Value Networks

The final stage of the pipeline involved training a value network (VL), identical in structure to the RL network, which produced a single prediction for every move, instead of a probability distribution. The network weights were trained by regressing on the state-outcome pairs using stochastic gradient descent to minimize the MSE between the prediction and the outcome. The resulting network suffered from overfitting due to the strong correlation between successive moves of the KGS dataset. To overcome this, the value network was instead trained on a dataset consisting of 30 million distinct

moves sampled from several different games. The significant drop in MSE (down to 0.226 from 0.37 on test set) was a strong indicator of minimal overfitting.

Step 4: Search with Policy and Value Networks

The core idea came together in the MCTS search which combined the PL and VL networks and implemented a look ahead search. A simulation would traverse the search tree, starting from the root, and update the action value Q and visit count N at each edge. After running multiple such traversals, the algorithm would select the most visited move from the root position.

Results

AlphaGo became the first ever computer Go program to defeat a human champion (Fan Hui, Winner, 2013-2015 European Go Championship) winning 5 to 0. This feat was believed to be atleast 10 years away at the time of the tournament owing to the complex nature of Go.