RESEARCH REVIEW

Mastering the Game of Go with Deep Neural Networks and Tree Search

Submitted by

Sumit Binnani

as part of Artificial Intelligence Nanodegree, Udacity

TABLE OF CONTENTS

BRIEF SUMMARY OF PAPER

The paper, *Mastering the Game of Go with Deep Neural Networks and Tree Search*, introduces an approach to a computer program, ***AlphaGo***, that uses ***value networks*** to evaluate board positions and ***policy networks*** to select moves for the game of Go. 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. The paper also introduces a new search algorithm that combines Monte Carlo simulation with value and policy networks. Using this search algorithm, program ***AlphaGo*** achieved a 99.8% winning rate against other Go programs and defeated the human European Go champion by 5 games to 0.

BRIEF DETAILS ON IMPLEMENTATION

1. STAGE1: SUPERVISED LEARNING

A 13-layer policy network, termed as SL Policy Network, is trained on randomly sampled state-action pairs from 30 million positions from the KGS Go Server. The neural network takes input features from the board position and outputs the probability of each move on the board being the actual next move.

Results: The network predicted expert moves on a held-out test set with an accuracy of 57.0% using all input features, and 55.7% using only raw board position and move history as inputs. Small improvements in accuracy led to large improvements in playing strength; larger networks achieve better accuracy but were slower to evaluate during the search. A faster but less accurate rollout policy, using a linear softmax; achieved an accuracy of 24.2%, using just 2μs to select an action, rather than 3ms for the policy network.

2. STAGE2: REINFORCEMENT LEARNING

The second stage of the training pipeline improved the policy network by policy gradient reinforcement learning. The games were played between the then current policy network and a randomly selected previous iteration of the policy network. Randomizing from a pool of opponents in this way stabilized the training by preventing overfitting to the then current policy.

Results: When played head-to-head, the RL policy network won more than 80% of games against the SL policy network. The RM policy network also won 85% of games against Pachi (*a sophisticated Monte Carlo search program, ranked at 2 amateur dan on KGS*) using no search at all.

3. STAGE 3: REINFORCEMENT LEARNING FOR VALUE NETWORK

The final stage of the training pipeline focuses on position evaluation, estimating a value function that predicts the outcome from a position of games played by using policy for both players. This neural network had a similar architecture to the policy network but outputted a single prediction instead of a probability distribution.

To mitigate the problem of overfitting, a new data consisting of 30 million distinct positions was generated using a set of self-play data set (each sampled from a separate game). Each game was played between the RL policy network and itself until the game terminated.

Results: Training on this data set led to MSEs of 0.226 and 0.234 on the training and test set respectively, indicating minimal overfitting.

4. STAGE4: SEARCH WITH POLICY AND VALUE NETWORK

AlphaGo combined the policy and value networks in an MCTS algorithm that selected actions by lookahead search. To efficiently combine MCTS with deep neural networks, AlphaGo used an asynchronous multi-threaded search that executes simulations on CPUs and computes policy and value networks in parallel on GPUs.

Results: The SL policy network performed better in AlphaGo than the stronger RL policy network. This was presumably because humans select a diverse beam of promising moves, whereas RL optimizes for the single best move. Also, the value function derived from the stronger RL policy network performed than the value function derived from SL policy.

RESULTS

The results of the tournament suggest that single machine AlphaGo is many *dan* ranks stronger than any previous Go program, winning 494 out of 495 games (99.8%) against other Go programs. AlphaGo also won 77%, 86%, and 99% of handicap games against Crazy Stone, Zen, and Pachi, respectively. The distributed version of AlphaGo was significantly stronger, winning 77% of games against single-machine AlphaGo and 100% of its games against other programs. Even without rollouts, AlphaGo exceeded the performance of all other Go programs, demonstrating that value networks provide a viable alternative to Monte Carlo evaluation in Go. However, the mixed evaluation (λ=0.5) performed best, winning ≥95% of games against other variants. This suggests that the two position-evaluation mechanisms are complementary: the value network approximates the outcome of games played by the strong but impractically slow, while the rollouts can precisely score and evaluate the outcome of games played by the weaker but faster rollout policy.

The distributed version of AlphaGo was evaluated against Fan Hui, a professional 2 *dan*, and the winner of the 2013, 2014 and 2015 European Go championships. Over 5–9 October 2015 AlphaGo and Fan Hui competed in a formal five-game match. AlphaGo won the match 5 games to 0. This had been the first time that a computer Go program has defeated a human professional player, without handicap, in the full game of Go—a feat that was previously believed to be at least a decade away.