Mastering the game of Go

Goal

The goal of the research is describing master the game of Go with deep neural networks and tree search. The program created for this challenge was named AlphaGo. The DeepMind team's goals was to find and apply techniques that are successful in games that have massive search spaces. A game of Go using a 19x19 board has a branching factor of 250, a state-space complexity of 10170, and a game-tree complexity of 10250, while chess has much smaller values of 35, 1047, 10123 . The massiveness of Go was previously thought to make the game unconquerable by 2 computers until decades in the future.
The authors of the paper suggest an innovative approach that leverages deep neural networks, specifically the usage of three neural networks:

1. A supervised learning (SL) policy network trained with human expert moves
2. A reinforcement learning (RL) policy network that evaluates self-play outcomes of the current state of the game
3. A RL value network that predicts the winner of games played with network number 2

Details of 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.
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.
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.
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 paper presents three sets of results for two different implementations of AlphaGo (one distributed, one not):

1. Both versions of AlphaGo significantly outperforms previously existing Go-playing AIs and are better than the best European player
2. Even without using all of its neural networks, just using the value network, AlphaGo performs almost as well as other AIs.
3. By throwing more hardware at the problem, AlphaGo performs even better

References

Mastering the game of Go with deep neural networks and tree search, by David Silver et al @ https://storage.googleapis.com/deepmind- media/alphago/AlphaGoNaturePaper.pdf