MASTERING THE GAME OF GO WITH DEEP NEURAL NETWORKS AND TREE SEARCH

Summary:

Mastering the Game of Go with Deep Neural Networks and Tree Search paper new search algorithm that combines Monte Carlo simulation with value and policy networks. AlphaGo, uses value networks to evaluate board positions and policy networks to select moves for the game. The deep neural networks are trained by supervised learning from human expert games, and reinforcement learning from games of self-play. 99.8% winning rate is achieved compared against other Go programs and defeated the human Go champion by 5 games to 0.

Implementation:

Stage 1: Supervised learning of policy network

SL Policy Network is a 13-layer policy network 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: 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.

Stage 2: 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: RL policy network won more than 80% of games against the SL policy network.

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. 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) to avoid overfitting. 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 which indicates minimal offer fitting.

Stage 4: 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.