Amichai Antebi
AIND Project 2, Build a Game-Playing Agent
June 3, 2017

Research Review Summary:

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

The Game of Go has about 250^{150} possible sequences of moves, In order to narrow the search to a beam of high-probability actions, the AlphaGo program used Monte Carlo tree search (MCTS) combine with 'value networks' to evaluate board positions and 'policy networks' to select moves, those are deep neural networks that trained by combination of supervised learning from human expert games, and reinforcement learning from games of self-play.

The researchers train 3 policy networks, the input is a representation of the board position, passes it through many convolutional layers and outputs a probability distribution over legal moves.
2 policies are trained to predict human expert moves in a data set of positions - A fast rollout policy (accuracy of 24.2% to predicted expert moves, using just 2 μ s) and a supervised learning (SL) policy network (accuracy of 57.0% using 3 ms).
In addition there is a reinforcement learning (RL) policy network that initialized to the SL policy network, and is then improved by policy gradient learning to maximize the outcome against randomly selected randomly selected of the policy network.

They use a reward function that is zero for all non-terminal time. The outcome is the terminal reward at the end of the game from the perspective of the current player at time step: +1 for winning and −1 for losing. Weights are then updated at each time step by stochastic gradient ascent in the direction that maximizes expected outcome.

The RL policy network won more than 80% of games against the SL policy network and 85% against the strongest open-source Go program, Pachi.

In addition to the policy networks, there is a value network, it's used in the final stage, and focuses on position evaluation. It has a similar architecture to the policy network but outputs a single prediction instead of a probability distribution. They train the weights of the value network by regression on state using stochastic gradient descent to minimize the mean squared error between the predicted value and the corresponding outcome.

AlphaGo combines the policy and value networks in an MCTS algorithm that selects actions by lookahead search. Each edge of the search tree stores an action value, visit count, and prior probability. The tree is traversed by simulation, at each time step an action is selected so as to maximize action value plus a bonus.

When reaches a leaf node, the leaf node may be expanded. The output probabilities are stored as prior probabilities for each legal action.
The leaf node is evaluated by the value network and by the outcome of a random rollout played out until terminal step, using the fast rollout policy.
Each edge accumulates the visit count and mean evaluation of all simulations passing through that edge.
Once the search is complete, the algorithm chooses the most visited move from the root position.

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 performed best.

AlphaGo won 99.8% against other Go programs and won 5 games to 0 against Fan Hui, a professional 2 dan, first time that a computer Go program has defeated a human professional Player.