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

I chose the Alpha Go article "Mastering the game of G with deep neural networks and tree search" since it uses a technique that replaces the traditional hand-crafted heuristics used from most of zero sum games such as chess or isolation, which is a different approach to the evaluation function used in the AI Nanodegree.

Paper's Goals:

This paper introduced a new approach using Monte Carlo tree search programs along with policy, value networks and the neural networks pipeline used to defeat a human professional player in the game of Go.

Techniques introduced:

The new technique introduced was using Monte Carlo tree search (MCTS) with value and policy networks. MCTS was used by the strongest Go programs at that moment, as this technique used Monte Carlo rollouts to search maximum depth without branching, by simulating actions until reach the terminal states. I will briefly mention the components used in AlphaGo, which uses a variant from MCTS.

Convolutional Neural Networks (CNN): We used CNN to train and to approximate the value function, policies or Q values. CNNs are useful to construct abstractions from an image (in this case the game board), so it can understand the state and recognized patterns.

Supervised Learning (SL) policy network: A policy trained using expert moves, which output a probability distribution over all legal moves. Also, a faster but less accurate network was trained, "rollout policy".

Reinforcement Learning (RL) policy network: A policy with an identical architecture to the SL policy and trained by playing with previous iterations of itself, which result in adjusting the policy towards the goal of winning instead of maximizing precision.

Value policy network: It allows evaluate a position by outputting a value, which predicts the expected outcome (win or loose) from position and playing following the RL policy.

AlphaGo MCTS: One of the innovations was the modification of the MCTS, such as the evaluation of the leaf node (when a node is expanded) using both the value network and the outcome of a rollout following the fast rollout policy. It is worth mention that before this evaluation the SL policy network processes the leaf position, and the output probabilities are stored as prior probabilities for each legal move.

In a nutshell, AlphaGo uses CNNs to train the policy and value networks that will be used in the modified Monte Carlo Tree Search. We trained the SL policy from expert moves (29,5 million positions from 160,000 games), the RL policy adjusted the parameters from the SL policy by playing against itself. Then, it used the RL policy to approximate the value function in order to estimate how much value has a position

given a game state. Also this value is complemented by the outcome of the rollout using fast rollout policy. In this way we can tie the components mentioned above.

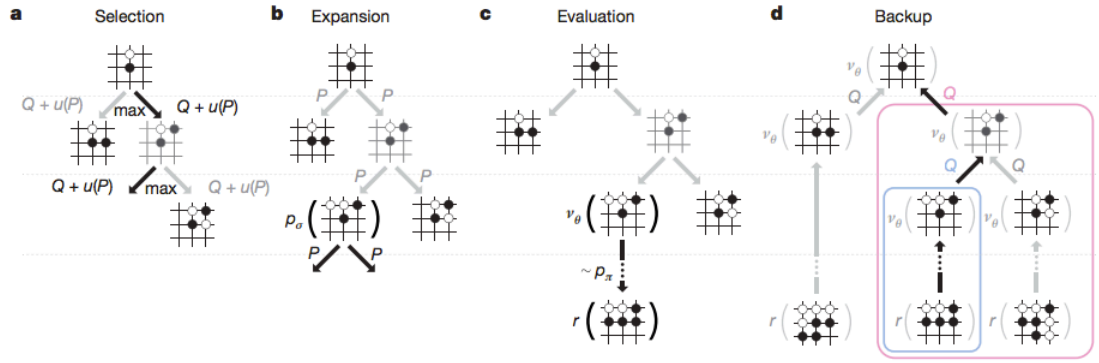

Fig1. Monte Carlo Tree Search in AlphaGo: a. The simulation transverse the whole tree starting from the root, it selects the node with the maximum Q + u(P) value which is the action value plus a bonus. b. When it reaches a leaf node, it expands to children nodes outputting the prior probabilities P for each legal move by following the SL policy. c. This leaf node is evaluated in two ways: using the value network and by the outcome of the rollout following the fast rollout policy. d. It backups the values obtained up to the expanded leaf node.

Paper's results:

AlphaGo won 99.8% of games against other Go games such as Crazy Stone, Zen, Pachi, Fuego y GnuGo. Also, variants of AlphaGo were assessed using just value network or just rollouts (at the moment of the position evaluation) and the result was that a mixed evaluation performed better than other combinations, (i.e. a weight of 50% of relevance for the value network and rollouts).

Comparing with IBM's DeepBlue match, AlphaGo evaluated thousands of times fewer positions against Fan Hui, since it selected the positions more intelligent using the policy networks and evaluated those positions using the value network and the Monte Carlo rollouts.