DeepMind AlphaGo (a paper review)

At the time of this write-up (08-2017), AlphaGo represents the state of the art agent for playing the game of Go, and it is one of the most exciting recent breakthroughs in artificial intelligence.

The game of Go is considered extremely challenging for artificial agents due to its vast search space (branching factor 250^150). Previous to AlphaGo, the best approaches to play Go were based on Monte Carlo Tree Search (MCST). This technique aims at reducing the depth and width of the search tree by using Monte Carlo rollouts to estimate the value of each state in the search tree. A Monte Carlo rollout of an action policy "p", is a game simulation under sampling actions from the policy “p” for both players. During the search, the average of multiple rollouts provides the position evaluation. As more simulations are executed, the search tree grows larger and the values of each node converge to the optimal value function for the node (which determines the outcome of the game for every board position under the assumption of perfect play by both players).

The core idea of AlphaGo is to enhance MCST with reliable predictors of the value of the current board position **(termed value networks)** and of the best move to take **(action policy networks)**.

In addition to this, the design of AlphaGo aims at:

- Learning policy networks and value networks from data using as little domain knowledge as possible
- Demonstrating that artificial agents can be trained by a mixture of human labeled data and simulated data

There are four types of neural networks which are all based on variations of deep convolutional neural networks:

A **supervised learning policy network (SL network)** which predicts then next move given a board configuration. It is trained from 30 million pairs of positions and moves played by human experts.

A **reinforcement learning policy network (RL network)**, which predicts the best next move to take given the current board position. While similar to the SL network, it differs in the training procedure and the meaning of the output label: the former predicts a move that a human would take while the latter predicts the best move to take in order to win the game. In practice, the RL network is trained using policy gradient learning by generating data obtained letting the previous iterations of the network playing against itself (at the first iteration this network is initialized with the SL network).

A fast rollout policy network (FR) which is similar to the RL network (and trained in a similar fashion), but it is shallower and thousand times faster to train and evaluate. It is used as policy network for the monte Carlo rollouts.

In addition to these, a **value network** predicts the probability of current position to result in a win or loss under the assumption of two agents which play perfectly. When trained on human game plays it severely overfitted, hence it was then trained on a dataset sampled from different games played by RL networks.

During the game, the agent runs a search procedure for a certain amount of time to estimate the action value function of the next possible moves: *Q(s,a)* (exploration phase). Then, the agent takes the action which maximizes the action value function (exploitation phase).

During the exploration phase, firstly the agent chooses the leaf node with the highest current value for Q and explores that branch *(selection phase).* Secondly, it evaluates the slow SL policy network to come up with a move which leads to tree expansion *(expansion phase)*. Thirdly, to assess the new positions, it runs the value network and a Monte Carlo rollout with the FR network and averages these estimates. These new leaf estimates are then propagated to update of the function Q for the overall tree.

Results:

The paper presents a comprehensive evaluation of the algorithm against artificial and human agents. In a nutshell AlphaGo significantly outperforms previously existing Goplaying AIs and is better than the current best human player. Remarkably, even just using the value network, AlphaGo performs almost as well as other AIs. Finally, it is demonstrated that the algorithm can be implemented on a cluster which improves performance accordingly.