INFO8006: Summary report for the reading assignment

Grégoire Roumache[1] and **Colin Stoquart**[2]
[1] *gregoire.roumache@student.uliege.be (s162533)*
[2] *colin.stoquart@student.uliege.be (s161140)*

I. PROBLEM STATEMENT

The problem here was to build an A.I. capable of mastering the game of Go, which means capable of beating champions of the game.

a. Why is it challenging? No previous A.I. has been able to show professional skills at the game of Go, even less showing the capacity of defeating a human champion. The difficulty of the game comes from its gigantic size (19×19) and the complexity of the game. To illustrate this complexity, we can observe that there are about 250^{150} possible sequences of move which is much more than the number of atoms in the observable universe. In this kind of game, i.e. games where we have a perfect information, we aim to determine a value function $v^*(s)$ that determines the output of the game in state s. However, because of the huge number of sequences, this is clearly impossible to determine $v^*(s)$ exhaustively. Two methods may be used to reduce the search tree and were efficient in other games. However, these methods weren't really tractable in Go and it remains a challenge to approximate v^*.

The A.I. also has to come up with a move as fast as possible. It can't just compute the best action for days. It only possesses a limited amount of time to tackle the difficult task of evaluating board positions and moves.

b. Why is it important? The difficulties encountered in Go (intractable search space, difficult choice of heuristic,...) are rather representative of many problems that are encountered in AI. Solving these problems in the framework of Go could lead to breakthroughs in other fields of AI.

The complexity of Go can't be overlapped with "brute" force. We can't just evaluate as much sequences as we want to choose an action. We have to filter sequences that look promising. This way of thinking approaches a little bit more an agent that "acts rationally". This is how human would act, we aren't going to simulate all possible sequences in our mind but we'd rather assess some sequences that look great. In other words, we go closer from the definition of AI.

II. CONTRIBUTIONS AND RESULTS

a. Neural networks. Instead of attacking the gigantic search space head on, the A.I. usually uses two methods. First, it reduces the depth of the tree by truncating the search and approximating a value function $v(s) \approx v^*(s)$ for a state s that is not a terminal state. Second, we can reduce the breadth of the tree by sampling actions from a policy $p(a|s)$.

In order to do this, researchers have constructed two kinds of neural networks : value networks to evaluate board positions and policy networks to sample action. These neural networks actually are *deep convolutional neural networks* which receive the image of the board as input. The new approach is to use these two types of networks in symbiosis and train them with both supervised learning from human expert games and reinforcement learning from games of self-play.

The training goes like this :

- Firstly, train a policy network p_σ with supervised learning. This network aims to predict expert's move. It improves its predictions (by finding optimal weights σ) by training on random examples from the expert's games and using a stochastic gradients ascent method. The policy network p_σ possesses 13 layers which alternate between convolutional layers and rectifier non-linearities. There's also a final soft-max layer. It outputs a probability distribution over all possible actions.

- Secondly, train a *fast* policy network p_π for action sampling during rollouts. This network runs faster ($2\mu s$ instead of $3ms$ for p_σ) but is less accurate (24.4% instead of $\approx 56\%$). The rollout policy p_π uses a linear soft-max of small pattern features.

- Then, train a policy network p_ρ (initially, the weights of p_ρ are equal to the weights of p_σ) with reinforcement learning, i.e. games of self-play. We add to this network a reward function which means that it doesn't only consider the prediction of the expert but also the perspective of the current player for winning or losing. This network is also trained thanks to gradient ascent method.

- And finally, train a value network v_θ whose goal is to predict the winner of the games played by p_ρ against itself. The value network v_θ has the same structure as the policy network except it doesn't output a probability distribution but a single prediction. Once again, we consider gradient ascent method.

A major problem could occur at this step. Indeed, if you use complete game to find optimal weight θ for v_θ, it could lead to overfitting. It means that your network will memorize the outcome of the game instead of learning how to actually win the game. It results in accurate predictions for your training set but in bad results for

your test set. The "simple" solution is to use only data generated from different games.

b. Searching algorithm. So now that we know how we will represent the game, we have to determine how to choose the best actions. This will be done thanks to the search algorithm. The search algorithm is a combination of a Monte Carlo simulation with the value & policy networks. The algorithm proceeds in four stages :

- *Selection.* Starting from the root, the algorithm selects at each node the action that maximizes the action value plus a bonus. The action value characterizes how a state looks promising whereas the bonus will encourage to explore nodes with a lower prior probability (calculated thanks to p_σ) that are less explored.
- *Evaluation.* When arrived at a leaf node, this node is evaluated. The evaluation takes into account two results. First, the evaluation from the value network of the current state and then, the result of a rollout simulation where each opponent follow the "fast" policy p_π.
- *Backup.* When the evaluation is finished, we update the action and bonus values for each node before the leaf node in the tree.
- *Expansion.* When the visit count of a leaf node exceeds a threshold, we expand the tree with its successors.

We can verify that the policy used to select actions and the evaluation converge to optimal play and optimal value function asymptotically. The use of deep convolutional neural networks enables a stronger performance than the traditional value functions based on a linear combination of input features.

c. Results. In the end, AlphaGo was proved to be capable of defeating several other Go programs such as the strongest commercial programs and the strongest open source programs, all using advanced Monte Carlo tree search (MCTS). It also defeated GnuGo which uses methods that preceded MCTS. AlphaGo proved to be way stronger than its opponents, defeating them 99.8% of the time. It also showed that value networks can be an alternative to Monte Carlo evaluation because even without rollouts, it had great performances. Of course, using both rollouts and value networks provide a superior winning rate.

AlphaGo also defeated a professional human Go player 5 games to 0. This is something that had never been done before. It was even believed to be a decade away. The brand new development of effective move selection and position evaluation functions for Go which is based on deep neural network and the combination of supervised and unsupervised learning helped achieve the successes of AlphaGo.

III. DISCUSSION

The advantages of the paper are numerous. Not only does the text explain very well the ideas for people who haven't already studied neural networks but it also has a large quantity of illustration that make the concepts very easy to understand. Note that the method described at the end seem well detailed. So it should allowed others specialists to corroborate the results and to continue to explore the topic.

We could think of two shortcomings of the paper. The first one is the vocabulary used which, while usually not difficult to understand, can sometimes be a bit technical. However, it may be possible that it's more a problem coming from our side because we're not very experienced yet and lack a more technical vocabulary. Finally, We could wonder about the impartiality of the author seeing that they are explaining (and defending) theirs owns results.