**Summary of „Mastering the game of Go with deep neural networks and tree search”**

Karolina Sala

In this review, I present the goals, techniques and results of the research described in the article „Mastering the game of Go with deep neural networks and tree search”. The result of the study was to create a computer program to play the game of Go that were able to beat a human. Created a program called AlphaGo.

The main goal was to find techniques that are successful in the challenging of classic games with massive search space and the difficulty of evaluating board positions and moves. Go has a high branching factor. The number of nodes in the game tree of Go can be $b^d$, which given the branching factor b=250 and depth of the game tree d=150. Go has an enormous search space. The depth of the search may be reduced by position evaluation: truncating the search tree at state s and replacing the subtree below s by an approximate value function $v(s) \approx v^*(s)$ that predicts the outcome from state s. The breadth of the search may be reduced by sampling actions from a policy p(a|s) that is a probability distribution over possible moves a in position s.

The methods that have been used are ’value network’ for evaluating board positions, and selecting moves using ’policy network’. The search space is reduced using this methods. They were trained policy networks to identify possible moves using the board position as a 19x19 image and convolutional layers to construct a representation of the position. The goal was to reduce the effective depth and breadth of the search tree.

They begin by training a supervised learning policy network from human expert games and then they trained a fast policy that can rapidly sample actions during rollouts. The next step was reinforcement learning from games of self-play by optimizing the final outcome of this games. Finally, they trained a value network that predicts the winner of games by the reinforcement learning policy network against itself. They introduced a new search algorithm combines Monte Carlo simulation with the ‘value’ and ‘policy’ network.

AlphaGo uses an asynchronous multi-threaded search that executes simulations on CPUs, and computes policy and value networks in parallel on GPUs. The final version of AlphaGo used 40 search threads, 48 CPUs and 8 GPUs.

AlphaGo is better than all other programs achieved a 99.8% winning rate against other Go programs and won the tournament as 5 games to 0. It was the first program which won against a professional Go player. The game of Go had been one of the grand challenges in the field of AI. A computer beating a human at the game Go was a remarkable breakthrough in the most modern games playing agents.

Bibliography:

“Mastering the game of Go with deep neural networks and tree search”, Nature vol 529, January 28, 2016