Project 2 – Research Review: “Mastering the game of Go with deep neural networks and tree search[1]”

1 Introduction and Overview

In this review, we give an overview of the key findings of the paper mentioned in the title, which presents various improvements to the state-of-the-art Go playing artificial intelligence (AI) agents. The proposed agent called “AlphaGo” made use of these improvements and was able to defeat the European Go champion by 5 games to 0. The techniques shown are based on the use of deep neural networks (DNNs), which made the “AlphaGo” program very efficient in practice.

Like Chess and Checkers, Go is a 2-player strategy board game of perfect information, which can be solved by using an exhaustive tree search algorithm that explores every possible move at a particular stage in the game and chooses the best possible move assuming the opponent would do the same. However, the complexity of such extensive search is inefficient for even smaller board games with just a few possibilities. The game of Go had been one of the biggest challenges in AI development prior to this work, because of its large size of search space. Past AI research has presented various algorithms for game playing, like alpha-beta pruning, iterative deepening minimax search and Monte-Carlo tree search (MCTS).

The paper proposes the combination of a supervised learning (SL) policy neural network (p_σ), which is trained by using a database of human expert moves, a reinforcement learning policy network (p_ρ), which optimizes p_σ to achieve the final goal of winning the game rather than accurate prediction of the ongoing move, and a value function neural network (v_Θ), which predicts the winner of games played by the RL policy network against itself. Additionally, for “AlphaGo” the MCTS technique is integrated into that combination of NNs in order to achieve the record-breaking feat of beating a human Go champion. DNNs in SL policy learning contribute to the novelty of the presented work.

2 Summary of the Key Results

There were two versions of “AlphaGo” considered. One used a single machine (40 search threads, 48 CPUs, 8 GPUs). Another one was distributed across multiple machines (40 search threads, 1202 CPUs, 176 GPUs). The program was evaluated against other Go playing agents that are based on high-performance MCTS algorithms. While only 1 game out of 495 games played against other competitive agents was lost by the single-machine “AlphaGo”, the distributed “AlphaGo” version won 77% of all its games against the same single-machine version and every game against other

Thomas Dang, 2017

competitive agents. However, the most impressive achievement was the victory over a professional human Go player in a tournament, with 5 games to 0, against Fan Hui.

References

[1] Silver, D., Huang, A., et al.; January 2016. “Mastering the game of Go with deep neural networks and tree search”; Nature vol. 529.