# **RESEARCH REVIEW by *Sean Alexander Frenn***

# *Mastering the game of Go with deep neural networks and tree search*

## Goals and Techniques

In the paper studied, deep neural networks and tree search are used together to create a game playing agent more performant than humans in a game such as Go. Reducing the search space of all possible moves using as less knowledge as possible about the domain and reinforcement learning using deep learning and supervised learning over millions of samples.

Given an enormous search space, classical AI methods couldn’t be applied to the problem to find a solution relatively easily. Therefore, couldn’t evaluate board positions and moves of the game of Go due to the games complexity. The paper presents a new approach based on multi-stage machine learning and a state-of-the-art tree search algorithm that enables a winning rate to achieve 99.8% against other Go programs and the defeat of the human European Go champion by 5 games to 0. On top of that, the program defeated in March 2016 the world’s best Go player. Which is an achievement that was not expected before at least a decade away.

The AlphaGo paper presents a way of using deep learning and neural networks to train a model that can assign values to board positions and decide which branch of the game tree to select during the play using human-designed heuristics.

Given the fact that supervised learning uses a lot of labels, but they couldn't develop a system performant enough to beat top Go players.

However, they used reinforcement learning, to make the system evaluate its performance against other instances of itself, they managed to train a really good game‐playing agent.

First, they used a supervised learning policy network trained with human expert moves, then reinforcement learning policy network that evaluates the agent’s current output. Followed by a fast rollout policy network which was trained to predict the next most likely output given the predicted next move (a technique that is a thousand times faster than the supervised learning policy but not as accurate). Finally, a value network estimates the probability of winning or losing given the current position and current player. Using the reinforcement learning technique, the team has been able to avoid an overfitting problem by training the program with 1.5 billion self‐play games instead of about 30 million human games.

To explore the game tree, the agent captures the goodness of a potential move in 4 different phases. First, it chooses the child with the highest value and explores down that branch. When it reaches a leaf node of that branch, it creates a new branch and performs a supervised learning policy network to come up with a good candidate. Then runs a value network to evaluate this position, and uses the fast rollout policy network to play from that position to the end of the game given a time limit. Then propagates the information up the search tree. After the give time limit running out, the agent chooses the child with the highest value found.

## Results

2 different implementations of the agents have been presented: distributed and not-distributed. All the 2 implementations won against the top Go players in the world even without using all of its full potential. It has been shown that a value network allows the agent to perform almost as well as other Go playing AIs. And the more computer power is provided, the better the agent performs.