

[NOTE THIS IS STILL A WORK IN PROGRESS]

Review: Mastering the game of Go with deep neural networks and tree search

A review of the article published in Nature[link] by the Google DeepMind[link] team in which they describe how they built the famous AlphaGo.

This review is written in partial fulfillment of the Nanodegree in Artificial Intelligence (AIND) by Udacity[link to AIND].

Article Overview

- What is described in this article?
- How did they do what they described?
- What makes it special?
- Results

Goals

The goals laid out by the authors of this paper were two-fold. Firstly, they wanted to showcase the exciting possibilities of *deep learning* by improving on the current state of the art in GO game playing agents to such a level previously thought impossible. Secondly, they wanted to move beyond the realm of conventional game playing agents and implement the concept of *self playing* through *reinforcement learning*.

Design Implementation

The authors set out to achieve their goals by the introduction two new types of board evaluation methods: *value networks* and *policy networks*.

These evaluation methods were inspired by the recent advancements of convolutional neural networks (CNNs) in the domain of computer vision. The basic premise was to train a *value network* to assign values (or weights) to each board position, indicating which positions provide a positional advantage.

Then, make a decision about which move to make next, they built two policy networks:

- ρ_{σ} : Trained on annotated expert moves using Supervised Learning (SL)
- ρ_{ρ} : Improving ρ_{σ} through Reinforcement Learning (RL)

[TODO]

Results

The results presented in this paper are unprecedented and caused a global upset, with experts predicting these results would only be achievable within the next decade at best. Through the combination of the novel 'value networks' and 'policy networks' to evaluate and choose new board positions, AlphaGo defeated Fan Hui, the European Go champion by 5 games to 0, and boasts a 99.8% winning rate against other Go playing programs.