

# Research and Review: AlphaGo

April 17, 2017

## 1 Introduction

In the development of game playing AI there have always been certain games present a greater challenge than others, Go is one such game given the large action space and challenge of board evaluation[1]. The search space is unfeasibly large to search all possible moves and board evaluation at any stage is non simple. Human player take hundreds of games before being able to develop a rudimentary understanding of deeper strategy[1]. Previous techniques have used Monte Carlo search trees in an attempt to search action/state spaces in a timely manner[2].

Presented here is a review of AlphaGo[3], where novel deep learning techniques were applied to the move selection and board evaluation.

## 2 Move selection and board valuation

### 2.1 Deep learning

In contrast to Monte Carlo (random choice) methods AlphaGo uses deep neural networks[3] to efficiently limit move selection search space. Among the innovative approaches used by the AlphaGo team was the high level abstraction of the game state (presented in data table 2 of [3]). This representation of the game state is simple yet powerful, as demonstrated by the significant win rate of AlphaGo against both human and other state-of-the-art AI. The deep neural network used here was initially a supervised learner trained on a set of expert human plays. A duplicate of this network was then trained further, via reinforcement techniques, where the agent was playing against itself. It is worth noting that to prevent over fitting at this stage the network was also trained against previous iterations of itself. In a strange twist the agent used from which the valuation function was derived, in this case was not the reinforcement trained policy network but the supervised learner from which it evolved. This is interesting given that the reinforcement learner selected moves of greater utility than the supervised learner but was better at game valuation. It was postulated in [3] that this was due to the reinforcement enhanced policy network, while picking better moves, did not predict opponents move to the same accuracy, [3] says that this significantly affected win percentages.

## 2.2 Hybrid MCTS technique

[3] found that their best agent was a hybrid of their deep learners and a variation of MCTS where moves were selected at each layer via a fast policy network to sample moves during MCTS rollouts. Their fully hybrid agent achieved 95% win rate vs agents that favored one technique over the other.

## 3 In summary

By developing an agent that is better at Go than every other AI and the human player that they failed to defeat, AlphaGo is on the forefront of AI and has provided a proof of concept for AI applications that, this writer believes, demonstrates that AI can solve problems that are not typically suspected to be feasible for this problem domain.

## References

- [1] Louis Victor Allis et al. Searching for solutions in games and artificial intelligence. Ponsen & Looijen, 1994.
- [2] Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In International Conference on Computers and Games, pages 72–83. Springer, 2006.
- [3] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.