**A Summary of the AlphaGo Paper**

Rob Straker

The AlphaGo paper[[1]](#footnote-1) introduced a new approach to computer Go using neural networks to reduce both the effective depth (game length) and breadth (number of legal moves per position) of the search tree.

The new approach relied on two general principles that have been successful in reducing the search space for large games like chess and Go:

1. Search depth can be reduced by evaluating the potential that a particular position will result in a win, effectively replacing the search subtree below a particular state with an approximate value function that predicts the outcome from the state.
2. Search breadth can be reduced by sampling actions from a policy that identifies probabilities associated with all possible moves for a particular position.

Prior to AlphaGo, the most successful Go programs were based on Monte Carlo tree search (MCTS) improved through policies trained to predict human expert moves. However, these utilized shallow policies or value functions based on a linear combination of input features.

AlphaGo combines MCTS with policy and value networks based on convolutional neural networks (CNNs). CNNs use layers of neurons to build a representation of an image. In AlphaGo, the board position is converted into an image and transformed via ‘convolutional layers’ (each includes a set of learnable filters) into an abstract representation of the image.

Neural network methods often use some form of online machine learning. For AlphaGo, two such methods were utilized for policy networks (supervised learning and reinforcement learning), and one for value networks (reinforcement learning). While supervised learning has been used for other Go programs, the combination with reinforcement learning is novel.

**AlphaGo’s Results**

AlphaGo is the first Go program to reach a professional level, and the first to defeat a human professional player in a full-sized game of Go, beating the European Go champion 5 games to 0. Compared to Deep Blue, AlphaGo evaluated thousands of times fewer positions and did not depend on a handcrafted evaluation function.

AlphaGo has achieved a 99.8% winning record against other Go programs, winning 494 out of 495 games against the strongest commercial programs, Crazy Stone and Zen, and the strongest open source programs, Pachi and Fuego, as well as the open source program GnuGo, which used state-of-the-art methods preceding MCTS.

1. David Silver et al., “Mastering the game of Go with deep neural networks and tree search,” *Nature* 529 (2016): 484. [↑](#footnote-ref-1)