Research Review

For this assignment, I have chosen the paper AlphaGo by the DeepMind Team.

In this paper, the AlphaGo Team presents their take on the problem of creating a Go agent capable of beating some of the best human players. A feat previously thought to a decade away, due to the sheer size of the problem. The Go agent is known as AlphaGo.

The problem of Go

The game of Go is played on a 19 x 19 board, where each player takes turns placing stones until a player runs out of moves or resigns. The size of the search tree in Go is approximately 250^{150} or 5 x 10^{359} nodes, compared to chess' approximately 35^{80} or 3 x 10^{123} nodes.

Current state of the art

Modern techniques to solving Go includes `Monte Carlo Tree Search` (MCTS) or specialized handcrafted branching techniques, thereby only expanding *interesting* nodes. However, none of these techniques go above the rank amateur in the world of Go players.

Deep Convoluted Neural Networks

Recent developments in deep convoluted neural networks - a technique very successful in the domain of image recognition led the authors to try to apply these to the problem of Go.

AlphaGo is based on MCTS combined with neural networks for branching and evaluation.

Neural Network usage in AlphaGo

AlphaGo utilizes several neural networks during the game. Two policy networks and one value network. Given a board state, the output of the policy networks is a probability distribution of board positions used in the selection of next possible moves.

For MCTS, AlphaGo utilizes a special fast rollout (FR) policy network, with a prediction accuracy of 24,2% and 2microsecond evaluation time.

For the actual game tree, AlphaGo utilizes a 13-layer policy network (SL) with a 57% prediction accuracy and a 3ms evaluation time.

Training the networks

AlphaGo's neural networks go through several stages of training.

The first stage trains the FR and SL policy networks using Supervised Learning. The networks were trained to predict human expert moves with real player date from the KGS Go Server. About 30 million training points where used.

The second stage uses policy gradient reinforcement learning with an RL policy network. This network has the same architecture and values as the SL network from the first stage. To train the network, games were played between the RL policy network and a randomly selected previous iteration of the RL policy network.

Finally, the value network is trained to predict game outcomes by letting the RL network generate a dataset distinct positions. Each position was sampled from games played between the RL policy network and itself until the game terminated.

Conclusion

The authors managed to combine MCTS with several specialised Neural Networks to create an excellent Go player. AlphaGo defeated the human European Go champion by 5 games to 0. AlphaGo also managed to beat the hitherto strongest computer Go players, Crazy Stone, Zen, Pachi and Fuego.