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

Paper Summary by Group 5:
Alok Thatikunta (111498776), Rahul Sihag (111462160), Sweta Kumari (111497926)
{athatikunta, rsihag, swkumari} @cs.stonybrook.edu

The game of Go has long been viewed as the most challenging of classic games for artificial intelligence owing to its huge search space and the difficulty of evaluating board positions and moves. A naïve way to solve these problems is by recursively calculating the best move but is infeasible due to the huge search space and number of legal moves per position. The search tree contains $b^d$ sequences of moves, where b is the number of legal moves per position and d is the depth game should tell the perfect play in each position. For Go, b is around 250 and d is around 150 and thus exhaustive search is not possible.

Different approaches have been proposed to reduce the search space like an approximation to truncate the search at a certain node and replacing the subtree below with an approximation function, Monte Carlo rollouts that search maximum depth without branching at all, by sampling long sequences of actions for both players from a policy p. These approaches focus on reducing both the depth and the breadth of the game tree. But they were never able to achieve superhuman performance in the game of Go.

Since convolutional neural networks achieved a great performance in the visual domains by using many layers of neurons to construct abstract representations of an image, a similar approach is taken here to use a value network to evaluate board positions and policy networks to select the best move. AlphaGo uses deep neural networks to learn a value network used to reduce search depth, and a policy network used to reduce search breadth. The neural networks are trained using a three-stage pipeline. It comprises of training in a supervised manner from expert human moves, a reinforcement learning policy network improves upon the network from stage one by playing games against itself and a value network is trained to predict the winner of games played by the RL policy network of stage 2 against itself. Since position adjacent to each other is highly correlated which result in memorizing the game value instead of generalizing to new positions. This problem is solved by generating a new dataset of millions of different positions each dataset is then used to play a separate game which results in eliminating the overfitting problem. In supervised learning training, a 13-layer policy network was trained from 30 million positions. The network alternates between convolutional layers and rectifier non-linearities. The network predicted expert moves with an accuracy of 57%, and small improvements in accuracy led to large improvements in playing strength. The RL network is identical in structure to the SL network from stage one, with weights initialized to the same values. Games are played between the policy network and randomly selected previous iterations of the policy network. AlphaGo combines the policy and value networks in an MCTS algorithm that selects actions by lookahead search.

AlphaGo beat the European Go champion 5-0 using a distributed version of AlphaGo with 40 search threads, 1202 CPUs, and 176 GPUs. AlphaGo is able to win 99.8% of the matches against other Go programs, even without rollouts, which indicate that the value networks provide a valuable information on the game.