AlphaGo paper summary.

Aim :- To build a go playing agent which is capable of beating professional go players.

The agent was built using monte carlo tree search with multiple neural networks.

Supervised Learning Policy Network : A 13 layer convolutional network was trained using 30 million positions on KGS Go server. This policy network predicts the probability of the next step for the player given the current position. This network is helpful in decreasing the breadth of the search tree.

Supervised Learning Rollout Network : For Monte Carlo tree search, many games are to be simulated to evaluate the position. Instead of random move generation for better results a rollout network is used to simulate the game to the end multiple times. Original SL policy network is deep, hence takes longer time to compute. The rollout network is trained on the same data but is lot less deeper (less accurate). Hence is able to predict the next move thousands of times faster than SL network.

RL Policy network : It has the same structure as SL policy network. This is developed by making the network play against the previous versions of itself and using the gradients the network is improved. This network performs much better than SL policy network and won 85 % of the games against pachi (strongest open source version of go program) in comparison to SL policy network which only won 11 % of the games.

RL value network : This network is very much similar to the policy network. Except this is a regression network. So, this predicts a score for the board position. Board positions of self-play dataset (RL policy network playing against itself) were used to train this network.

Search using value and policy networks : Monte Carlo Search algorithm is used to search through the positions. SL network gives the prior probabilities of the each board state. Node for doing a rollout is picked based on the score associated with each node. It comprises of the number of simulated wins, prior probability, number of times the node has been selected (so as to aid exploration).

Results :- The created agent consistently performed much better than all previously existing go programs. It also won against the programs in the handicapped version (four free moves for the opponent). This version of the agent happens to be the first go program to win against professional go player.