Research Review of AlphaGo
(https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf)

The paper was published by Google's DeepMind team after AlphaGo became the first Computer Go program to beat a human professional Go player without handicaps on a full-sized 19×19 board[1] in late 2015. AlphaGo has won even more matches since then.

The issues:

- Compared to chess, which has about 35 legal moves per position and game length of 80. Go has 250 legal moves per position with game length of 150. This makes exhaustive search infeasible.
- While utilizing heuristic evaluation to reduce the search depth had worked for games such as chess, it doesn't work well for Go due to its complexity.
- Reducing the search breadth through Monte Carlo rollouts provide effective position evaluations, which achieves great performance in games like backgammon, but only "weak amateur level play" in Go.

AlphaGo approach and result :

1. Use supervised learning to predict expert moves using 30 million positions from the KGS Go Server. This is a 13-layer convolutional neural network, called the policy networks. The input is the board position as 19x19 images and the output is the probability for each legal move. Deeper network achieved better accuracy but lack of efficiency.
    - This results in 57% accuracy on the test set using all input features. It takes 3 ms to select a move. A faster network, with 24.2% accuracy achieves just 2 μs to make a move.
2. Use reinforcement learning to predict expert moves using identical structure as SL policy network, with inputs from policy networks playing against itself. Opponents are randomly chosen and weights of the network are updated according to the outcome of the game.
    - RL policy network won more than 80% of games against the SL policy network, and 85% of games against Pachi - the strongest open-source Go program with a sophisticated Monte Carlo search algorithm.
3. Use reinforcement learning on position evaluation with similar structure as the policy networks, but output a predicted game outcome given the current game state. This is called the value networks. This is similar to the heuristic evaluation functions except they are learned through RL instead of explicitly designed by human experts. To avoid overfitting, RL policy networks play against itself to the end of the game using newly generated self-play datasets.
    - The MSEs are .226 and .234 for training and test set.
4. Monte Carlo tree search with large number of simulations guided by the policy and value networks. Each simulation starts from the root state, store and update action values, visit counts and prior probabilities.
    - SL policy network performed better than the stronger RL policy network. They explained this is because humans select a diverse beam of promising moves, whereas RL optimizes for the single best move.
    - Value function derived from the stronger RL policy network performed better than the SL policy network.
    - AlphaGo ran on 48 CPUs and 8 GPUs and the distributed version of AlphaGo ran on 1202 CPUs and 176 GPUs.

The final evaluation[2], at the time the paper is written:

[1] AlphaGo (https://en.wikipedia.org/wiki/AlphaGo)
[2] Google DeepMind's AlphaGo: How it works (https://www.tastehit.com/blog/google-deepmind-alphago-how-it-works/)

AI name	Elo rating
Distributed AlphaGo (2015)	3140
AlphaGo (2015)	2890
CrazyStone	1929
Zen	1888
Pachi	1298
Fuego	1148
GnuGo	431

[1] AlphaGo (https://en.wikipedia.org/wiki/AlphaGo)
[2] Google DeepMind's AlphaGo: How it works (https://www.tastehit.com/blog/google-deepmind-alphago-how-it-works/)