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

Summary:
The goal of this paper is to provide a summary of the strategies and configurations that DeepMind's Go winning AlphaGo used to tackle the challenges of Go and achieved unprecedented success in the field of this game using a combination of MCTS (Monte Carlo Tree Search) and DNN (Deep Neural Network).

Challenges:
The search space for Go is extremely high, valued at (b^d with $b \approx 250$, and $d \approx 150$), much greater than other chess games. Such large search space has made many traditional search algorithms such as minimax with alpha-beta pruning, negamax and its variants intractable, also made the existing combinations of searching and evaluation techniques such as MCTS + Policy Network + Hand-crafted evaluation function used in other Go programs such as Pachi, Zen and CrazyStone.

Strategies overview:
AlphaGo incorporates a dual stream Search and Evaluation process both run asynchronously on numerous CPUs and GPUs. The Search is accomplished by MCTS with fast rollout, and two separate deep neural networks that train the policy and value functions, which then, are deployed for move predictions.

Components overview:
Search: AlphaGo devised its own variant of MCTS named APV-MCTS (asynchronous policy and value), which consists of four stages: selection, expansion, evaluation and backup. It is asynchronously simulated on CPUs.
Rollout policy: A fast, linear Softmax policy returning optimal move from the current state based on incrementally computed features. It takes into account of some hand-crafted "hints" and recognition patterns as well as caching existing move trees for future use. It is used in parallel with Policy and Value networks, although less accurate, due to its fast nature, it is used along for the optimum final speed.
Policy network: The policy network is a neural network which returns a probability distribution function for a move given a state, it consists of a 19 by 19 image stack with 48 feature planes (network layers).
Value network: The value network is similar to policy network in terms of network structure, but instead it returns a single value for a move given a state, this is AlphaGo's version of "evaluation function", which removes the biases presented in traditional heuristics such as Opening Book, Progressive Widening, by purely relying on the trained results. Value network takes the trained policy networks and regresses to an optimum model which predicts the final outcome (whether the current player wins).

Training phase:
The program is trained heavily in the beginning to obtain desirable rollout policy, as well as policy network and value network.
Rollout policy: It is trained from 8 million human play games with its weight Ω updated using stochastic gradient descent as well as with some hand-crafted rules.
Policy network: It is trained in two stages, with first being classification trained with expert games in KGS (Go data server) and the second being reinforcement learning with games played with its previous iterations.
Value network: It is trained by self-playing with randomly selected time step (time to make a move) and move from the previously trained policy networks.

Result:
With the combination of all above techniques and innovations, AlphaGo has successfully beaten Fan Hui, first time AI beats human in a full, formal game and achieve an Elo (Go ranking) score of over 3000, around 60% improvement over the second Go program under the same time settings.