**AlphaGo article.** This paper describes a new artificial intelligence game playing agent in the game of Go. It explains a new approach to designing a game playing agent which uses ‘value networks’ to evaluate the board and ‘policy networks’ to select moves. Deep neural networks are trained by supervised learning from human expert moves and reinforcement learning from self-play. This new agent was able to beat all other Go programs and a European Go champion. (Note: subsequent to this paper, AlphaGo played against a world champion and won 4 games and lost one)

**Background**

Although in theory every game of perfect information has an optimal value function v\*(s), for a game like Go, exhaustive search through the game space is infeasible. Such search tree would have *b^d* sequences of moves (b ~ 250, d~150, where b is breath and d is depth).

In general, search space is reduced by two techniques: (1) depth by position evaluation and (2) breath by sampling. A position evaluation function truncates the search tree and replacing the subtree below *s* and approximating value function v(s) ~ v\*(s). Breath of the search tree can be reduced by some kind of sampling action over available moves. For example, a policy function p(a|s) is a probably distribution over possible moves a in position s.  Monte Carlo rollout search is an example of such a sampling policy.  While these techniques have been able to achieve strong amateur play, they have some limitations.

AlphaGo’s architecture is similar to deep convolutional neural networks, which have achieved great successes in visual domains such as image classification and face recognition. AlphaGo uses many layers of neurons arranged in overlapping tiles, and constructs abstract, localized representations of the board, and reduces the search tree’s depth and breadth.

**AlphaGo Training and architecture:**

AlphaGo was trained by the NN using machine learning.  It was trained by a supervised learning policy network p from expert human moves.  It was also trained with a fast policy p that can rapidly sample actions during rollouts.  The next stage of the training was Reinforcement Learning (RL), which was to train on position evaluation and estimating a value function. The goal was to optimizing the final outcome of games in self-play.  And finally, a value network v was trained by regression to predict the the winner of games. AlphaGo then combines the policy and value networks with MCTS

**Result and Compared with Deep Blue**

AlphaGo won 99.8% against other Go programs and won against Fan Hui, a professional 2 dan. AlphaGo was developed with a combination of deep neural networks and tree search. Compared with Deep Blue, it evaluated thousands times fewer moves by selecting positions to evaluate the potential moves more intelligently. This game playing is more like how humans play. Deep Blue also relied on game specific evaluation function, but AlphaGo’s neural networks are trained from through general-purpose supervised learning and reinforcement learning.

As a result, AlphaGo was able to combine tree search with policy and value networks, and was able to achieve a professional level in Go which was believed to be at least a decade away.