AlphaGo by the DeepMind Team

AlphaGo is a unique machine learning system combining deep neural networks, multi-layer neural nets, and Monte Carlo simulation to search for, predict and evaluate, and select game actions with a search tree. Efficiently combining these approaches has enabled the system to consistently best all other previously state-of-the-art Go playing systems, and even the highest ranked professional human players.

The development team sought to demonstrate the utility of deep and conventional neural nets when combined with state of the art techniques for producing superhuman performance in other games, and those producing amateur performance in Go. Previously, some of the state of the art techniques for tree searching had been intractable for Go given its relatively enormous state space volume. More conventional techniques include depth-limited search with evaluation of truncated sub-trees, probabilistic policies for using actions given a distribution of possible states, and Monte Carlo 'rollouts' to find probability distributions of actions across states. Seamlessly combining these with tailored neural networks enabled AlphaGo's breakthrough efficiency.

Conventional tree searching techniques were combined with distinct multi-layer neural networks trained through supervised as well as reinforcement learning, and a convolutional neural network for handling board states as images with various levels of abstraction. The neural networks performed action evaluation on different time scales (micro- vs. milliseconds for fast or standard policy networks) for exploration within the search tree. Evaluative policy-finding neural networks were combined with Monte Carlo search to enable rapid sampling of actions during rollouts of fast policy, achieving 24% accuracy in just 2 microseconds of time per selection. For more optimal move selection policy, pattern recognizing networks using supervised learning with expert moves as targets were able to achieve 57% accuracy, with 3 millisecond run times. Reinforcement learning was then used on a copy of this expertly trained supervised neural net to encourage convergence towards optimal single moves or 'beams' of patterns by facing off against earlier iterations of itself, preventing overfitting to the policy it favored at any one time. By using only endgame-dependent values in this reinforcement training phase, this version of the valuation net was able to best the regular supervised learning net 80% of the time, and win against Pachi, a commercial Go program, 85% of the time. For reference, the next best state of the art supervised learning convolutional neural net had only achieved win rates of 11% and 12% against Pachi and Fuego, respectively.

The exploration-exploitation dialectic showed up when comparing the apparently superior performance of the reinforcement learning network to that of the supervised learning network in matches with human professionals. Whereas the RL network tends to focus on just the very best moves, this comes at the cost of missing opportunities to counter the more diverse strategies of humans. The SL network's tendency to predict wide ranges of outcomes rather than select just the best ones proved valuable in this regard, allowing the system to counter more creative opponents. A distributed, multi-machine version of AlphaGo was also developed, overhwhelmingly outperforming all other state-of-the-art Go playing systems, the single-machine version of AlphaGo, and the very best human professionals. The complementary systems of relatively slow evaluation and fast policy rollouts combined with parallelization afforded by many core architecture (to which neural nets lend themselves well) culminated in the 5-0 victory against the world-class Fan Hui a decade before most thought it possible.