Review <the game of Go with deep neural networks and tree search>

Goals：

Alpha Go achieved a 99.8% winning rate against other Go programs, and defeated the human European Go champion by 5 games to 0. This is the first time that a computer program has defeated a human professional player in the full-sized game of Go, a feat previously thought to be at least a decade away.

Techniques :

Deep neural networks: have achieved unprecedented performance in visual domains: for example, image classification, face recognition, and playing Atari games; it uses many layers of neurons, each arranged in overlapping tiles, to construct increasingly abstract, localized representations of an image.

Supervised learning of policy networks: The SL policy network $\mathrm{p}_\sigma(\mathrm{a}|\mathrm{s})$ alternates between convolutional layers with weights σ, and rectifier nonlinearities. A final softmax layer outputs a probability distribution over all legal moves a. The input s to the policy network is a simple representation of the board state. The policy network is trained on randomly sampled state-action pairs (s, a), using stochastic gradient ascent to maximize the likelihood of the human move a selected in states. $\Delta\sigma \propto \frac{\partial \log p_\sigma(a|s)}{\partial \sigma}$

Reinforcement learning of policy networks: The RL policy network p_ρ is identical in structure to the Supervised learning policy network, and its weights ρ are initialized to the same values, ρ=σ. $\Delta\rho \propto \frac{\partial \log p_\rho(a_t|s_t)}{\partial \rho} z_t$

Rollout policy: The rollout policy $p_\pi(\mathrm{a}|\mathrm{s})$ is a linear soft-max policy based on fast, incrementally computed, local pattern-based features consisting of both 'response' patterns around the previous move that led to states, and 'non-response' patterns around the candidate move a in states. Each non-response pattern is a binary feature matching a specific 3×3 pattern centered on a, defined by the color (black, white, empty) and liberty count (1, 2, ≥3) for each adjacent intersection. Each response pattern is a binary feature matching the color and liberty count in a 12-point diamond-shaped pattern centered around the previous move.

Reinforcement learning of value networks: estimating a value function $v^p(s)$ that predicts the outcome from position s of games played by using policy p for both players. This neural network has a similar architecture to the policy network, but outputs a single prediction instead of a probability distribution.

$$v^p(s) = E[z_t|s_t = s, a_{t\ldots T}\sim p]$$

Monte Carlo tree search: uses Monte Carlo rollouts to estimate the value of each state in a search tree. As more simulations are executed, the search tree grows larger and the relevant values become more accurate. The policy used to select actions during search is also improved over time, by selecting children with higher values. Asymptotically, this policy converges to optimal play, and the evaluations converge to the optimal value function.

Results:

By combining tree search with policy and value networks, Alpha Go has finally reached a professional level in Go, providing hope that human-level performance can now be achieved in other seemingly intractable artificial intelligence domains.