Review on playing GO

1. Supervised learning of policy networks

Goal: predict what a master would do given a board state

method $\quad s \rightarrow \boxed{\sigma} \rightarrow p_\sigma(a|s)$ $\quad (a, s)$: training data

gradient ascent

$$\Delta\sigma = \alpha \frac{1}{m} \sum_{i=1}^{m} \nabla \log p_\sigma(a^i | s^i)$$

2. Reinforcement learning of policy networks

Goal: make moves that win games

method: start from prior network. Play self games

sample $i \rightarrow z_i$ outcome $= \pm 1$

$$\Delta\sigma = \alpha \frac{1}{m} \sum_{i=1}^{m} \left[\sum_{t=1}^{T_i} \nabla \log p_\sigma(a_i^t | s_i^t) \right] z_i$$

Policy Gradient Theorem $\rightarrow$ first order step to maximize expected reward

MCTS

Reinforcement learning of value networks

Predicts $\overbrace{\text{win or lose}}^{\text{value}}$, quickly!

- goal: make it very fast
  does not need to be super accurate

method: use network trained to play games

goal: predict game value

both players use network

$$s \to \boxed{\;\theta\;} \to v_\theta(s)$$

gradient descent

$$\min E(z - v_\theta)^2$$

$$E(z_t \mid s_t = s, a_{t,\ldots,T} \sim p)$$

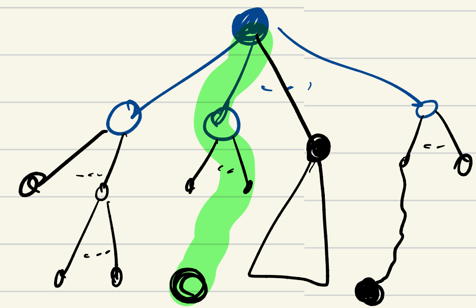
MSE

$$\Delta_\theta \propto \nabla_\theta(v)\,(z - v_\theta)$$

skipping value networ

Monte-Carlo Tree Search (MCTS)

- Balance between exploration
  exploitation



root = start of game

s

edge selected action a

how? $\rightarrow a = \arg\max \{Q(s,a) + u(s,a)\}$

$$u(s,a) = \frac{P(s,a)}{1 + N(s,a)}$$

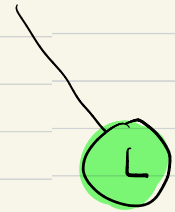
$Q(s,a)$ = estimate of value in subtree

$P(s,a)$ = prob of win

$N(s,a)$ = # of visits

leaf of current tree → evaluate
→ backup
→ expand

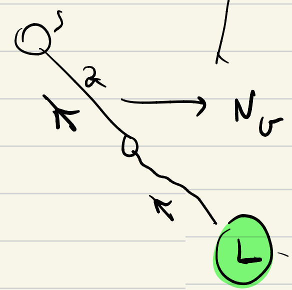
Generic MCTS



evaluate: how traditional fast rollouts

$v(L)$

backup



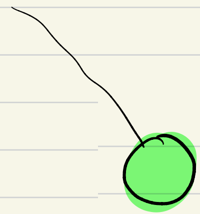
$$N_v(s,a) = N_v(s,a) + 1$$

$$Q(s,a) = \frac{1}{N_v(s,a)} \sum_{i=1}^{m} v^i$$

define

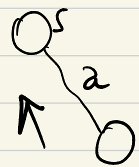
Expand L when visit count is "large enough"

AlphaGO



fast rollout

trained network (prediction of game value)



$$Q(s,a) = \lambda \frac{W_r(s,a)}{N_r(s,a)} + (1-\lambda)\frac{W_\sigma(s,a)}{N_\sigma(s,a)}$$

Expand

40 threads 48 CPUs 8 GPUs

→ for $v_\theta(L)$ leaf evaluation

40 threads 1202 CPUs 176 GPUs

"methods require several orders of magnitude more computation than traditional heuristics)

- AlphaGo "many dans stronger than any other previous program"
- Defeated (2015) Fan Hui 5-0