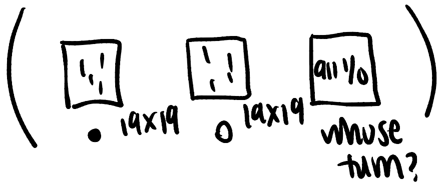
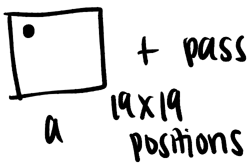
LECTURE 20

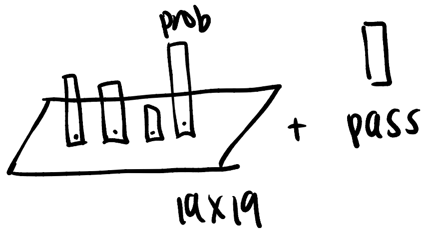
Alpha GO

state = 

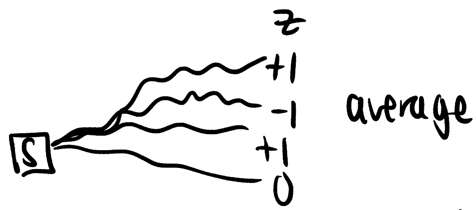
S

Action ∈ 

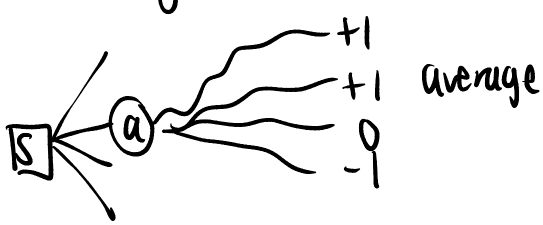
policy: $P(a|s)$



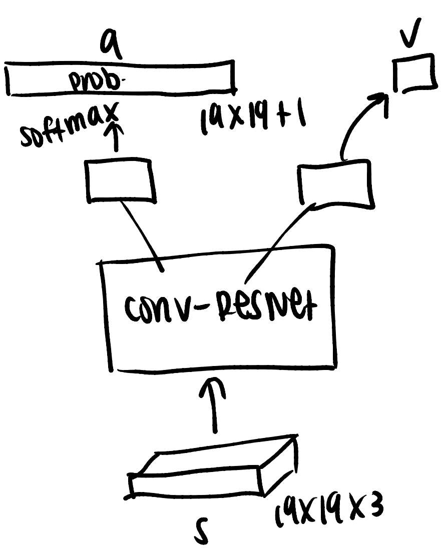
value: $v(s) = E_{policy}(z|s)$



$Q(s,a) = E_{policy}(z|s,a)$



Deep RL



supervised:

human teacher

$$\Delta \sigma \propto \frac{\partial \log p_\sigma(a|s)}{\partial \sigma}$$

RL:

RL ↓ supervised (fixed) ↓

$$\boxed{s} \xrightarrow{p_\rho} (a) \rightarrow \boxed{s'} \xrightarrow{p_\sigma} (a') \rightarrow \boxed{s''} \xrightarrow{p_\rho} (a'') \rightarrow \dots \rightarrow z$$

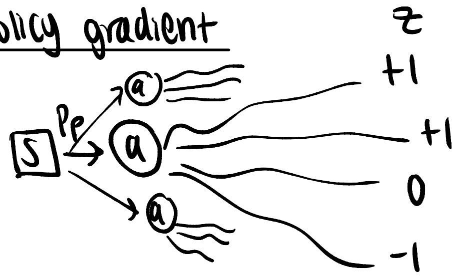
self p_ρ

$$\Delta \rho \propto \frac{\partial}{\partial \rho} \log p_\rho(a|s) \cdot z \quad \begin{matrix} +1 \\ -1 \\ 0 \end{matrix}$$

↓ reinforcement

stochastic gradient ascent on $E(z)$
↳ try to max expected payout

policy gradient



- if action good, ↑ prob. of action
- if action bad, ↓ prob. of action

$$\max E_{p_\rho(a|s)}[z] = \sum_a z \cdot p_\rho(a|s)$$

average over payoffs of all trajectories

$$\frac{\partial}{\partial \rho} E(z) = \sum_a z \frac{\partial}{\partial \rho} p_\rho(a|s)$$

$$= \sum_a \left[z \cdot \frac{\partial}{\partial \rho} \log p_\rho(a|s) \right] \times p_\rho(a|s)$$

$$= E_{p_\rho(a|s)}\left[z \cdot \frac{\partial}{\partial \rho} \log p_\rho(a|s) \right]$$

γ
$\Delta \rho$

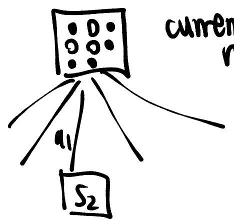
learn $V_\theta(s)$

$$\boxed{s} \xrightarrow{p_\rho} (a) \rightarrow \boxed{s'} \xrightarrow{p_\sigma} (a') \rightarrow \boxed{s''} \xrightarrow{p_\rho} \dots \rightarrow z$$

$$\Delta \theta \propto -\frac{\partial}{\partial \theta}(z - V_\theta(s))^2$$

↳ gradient descent

play the game



$a_1 \sim p_\rho(a|s_1)$ (impulse/reflex)

OR

a_1 to max $V_\theta(s_2)$ (desire)

fast thinking

Which one is more accurate?

$V_\theta(s_{20})$ because

closer to the end of game

$\Rightarrow$ more accurate estimate of average score

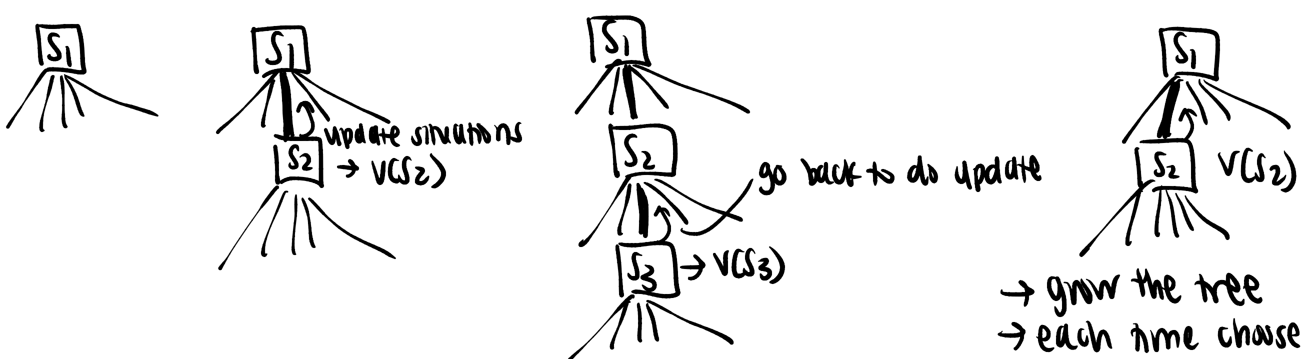
monte-carlo tree search

choose best a_1

by max $Q(s_1, a)$

or sample $a \propto N(s,a)$

MCTS

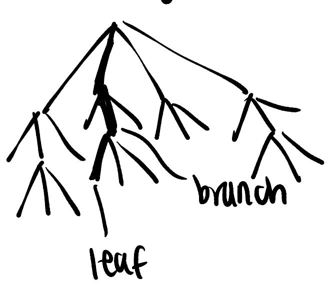


→ grow the tree

→ each time choose a_1 based on criteria

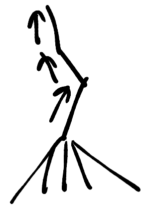
→ can't see full game tree, randomly select some branches

step 1: selection: go down a branch



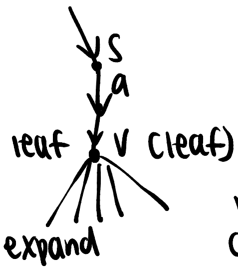
$\boxed{\text{step 2:}}$ expand

$\boxed{\text{step 3:}}$ back up



go back the branch

back-up



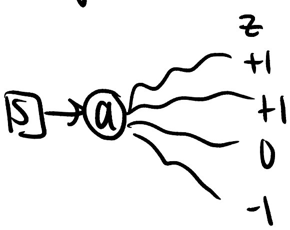
more accurate
(close to end)

$N(s,a) = N(s,a) + 1$ (# of visits)

$W(s,a) = W(s,a) + V$ (total score)

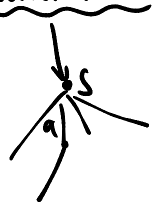
$Q(s,a) = W(s,a) / N(s,a)$

(average over leaf



average

selection



choose a by max

$Q(s,a) + U(s,a)$ → exploration

exploitation

↓

uncertainty

- balance btwn what know best and exploring

$U(s,a) \propto \frac{P_\sigma(a|s)}{N(s,a)+1}$

policy according to supervised learning

gives randomness

reduce breadth

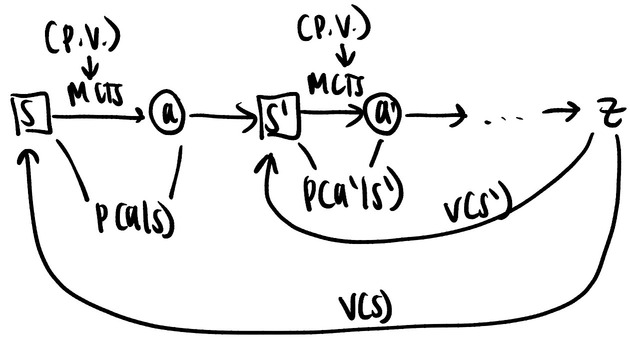
① train 2 policy networks — one supervised, one RL

② train value network

③ MCTS

Alpha Go Zero
- No human data (PG)
- self play by MCTS

- subconscious vs. conscious
  - (more explicit thinking
  - ↓ MCTS



RL in general
Markov decision process (MDP)

$$\boxed{s_0} \to \cdots \to \underbrace{\boxed{s_t} \xrightarrow{a_t} \boxed{s_{t+1}}}_{r_t} \underbrace{\xrightarrow{a_{t+1}} \boxed{s_{t+2}}}_{r_{t+1}} \to \cdots$$

$r_t + r_{t+1} + \cdots$

(no r_t in Go
r_t in Atari)

World Model
① Dynamics model:

Deterministic $s_{t+1} = F(s_t, a_t)$

Stochastic $P(s_{t+1} | s_t, a_t)$

② Reward model:

Deterministic: $r_t = r(s_t, a_t)$

Stochastic: $P(r_t | s_t, a_t, s_{t+1})$

Reward to go: $G_t = r_t + r_{t+1} + \cdots$

Goal: $\max E(G_t)$ max cumulative reward

policy: $p(a|s)$

value: $V(s)$, $Q(s, a)$

Model-based
known (1), (2), plan by e.g. MCTS
play out in imagination Go

Model-free
Do not know (1), (2)
play out in real environment, swimming, riding bicycle
practice in real world