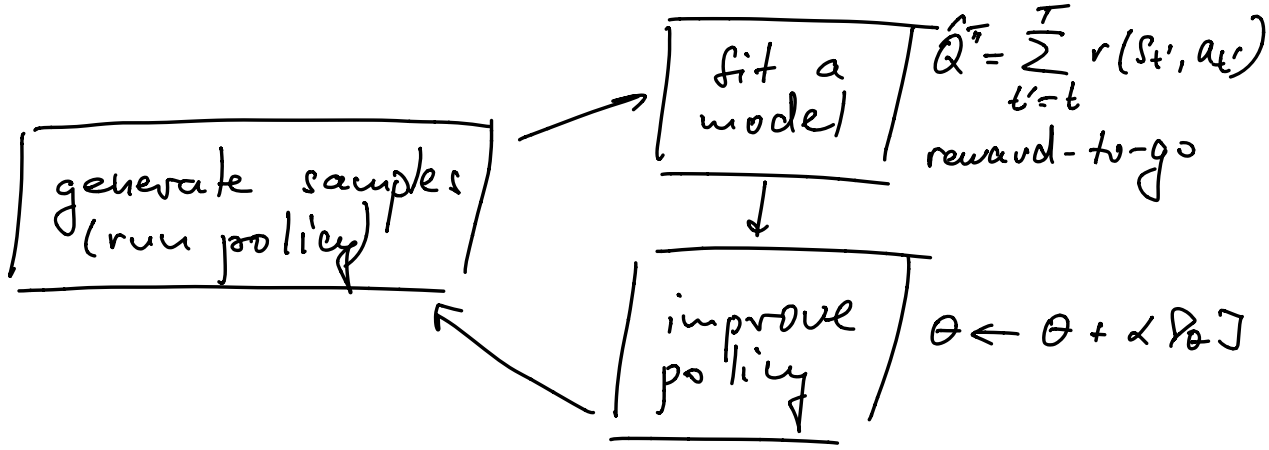
Actor Critic Methods

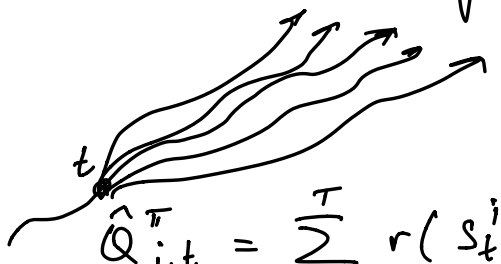
last time: policy gradient (PG)

REINFORCE

1. sample $\{\tau_j\}_j$ from π_θ
2. estimate $\nabla_\theta J(\theta)$
3. update: $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$



today: discuss better ways to estimate $\hat{Q}^\pi$
→ consider trajectory space



$\hat{Q}^\pi_{i,t} = \sum_{t'=t}^{T} r(s_t^i, a_t^i)$ → high variance in estimating $\nabla_\theta J$

want: $\hat{Q}^\pi_t \approx \sum_{t'=t}^{T} \mathbb{E}_{\pi_\theta}[r(s_{t'}, a_{t'})]$ expected reward-to-go

→ smaller variance
→ more consistent gradient estimates
→ learn faster

- <u>baseline</u> : does not change $\nabla_\theta J$
→ use state-dependent baseline

$b = V(s_t) = \mathbb{E}_{a \sim \pi_\theta}[Q(s_t, a_t)]$

leaves $\nabla_\theta J$ invariant (HW)

<u>but</u> : expectation over action further reduces variance!

<u>note</u> : action-dep. baselines change $\nabla_\theta J$

<u>recall</u> :

$$Q^\pi(s_t, a_t) = \sum_{t'=t}^{T} \mathbb{E}_{\pi_\theta}[r(s_{t'}, a_{t'}) \mid s_t, a_t]$$

expected return following policy π_θ given (s_t, a_t)

$$V^\pi(s_t) = \mathbb{E}_{a_t \sim \pi_\theta}[Q^\pi(s_t, a_t) \mid s_t]$$

expected return following policy π_θ given s_t

def: advantage function:

$$A^{\pi}(s_t, a_t) := Q^{\pi}(s_t, a_t) - V^{\pi}(s_t)$$

how much better action a_t is than the average action in state s_t under π_θ

- using $b = V^{\pi}(s_t)$, PG update reads

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t^i | s_t^i) \underbrace{\left[\left(\sum_{t'=t}^{T} r(s_{t'}^i, a_{t'}^i) \right) - b \right]}_{= A^{\pi_\theta}(s_t^i, a_t^i)}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t^i | s_t^i) \, A^{\pi_\theta}(s_t^i, a_t^i)$$

- idea: the better we can estimate A^{π_θ}, the lower the variance of $\nabla_\theta J$

- Value function fitting:

- which one of A^{π}, Q^{π}, V^{π} should we fit?

$$Q^{\pi}(s_t, a_t) = r(s_t, a_t) + \sum_{t'=t+1}^{T} \mathbb{E}_{\pi_\theta}[r(s_{t'}, a_{t'}) | s_t, a_t]$$

$$= r(s_t, a_t) + \mathbb{E}_{s_{t+1} \sim p(s_{t+1} | s_t, a_t)}[V^{\pi}(s_{t+1})]$$

use single sample estimate for p

$$\approx r(s_t, a_t) + V^\pi(s_{t+1})$$

$$\Rightarrow$$

$$A^\pi(s_t, a_t) \approx r(s_t, a_t) + V^\pi(s_{t+1}) - V^\pi(s_t)$$

→ only V^π occurs! → fit V^π

use a DNN with parameters φ

$$s \longrightarrow DNN_\varphi \longrightarrow V^\pi_\varphi(s)$$

– policy evaluation: what do we fit V^π_φ to?

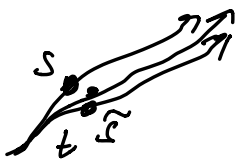
$$V^\pi(s_t) = \sum_{t'=t}^{T} \mathbb{E}_{\pi_\theta}[r(s_{t'}, a_{t'}) \mid s_t]$$

$$J(\theta) = \mathbb{E}_{s_0 \sim p(s_0)}[V^\pi(s_0)]$$

1) MC policy evaluation (same as PG, last time)

$$V^\pi(s_t) \approx \sum_{t'=t}^{T} r(s_{t'}, a_{t'})$$

single sample estimate (unlikely to repeat trajectory twice)

why meaningful? 

different trials visit different states but NN can interpolate / generalize

$$V^\pi_\varphi(s) \approx V^\pi_\varphi(\tilde{s})$$

training data set for V_φ^π:

$$\left\{ s_t^j,\ y_t^j = \sum_{t'=t}^{T} r(s_{t'}^j, a_{t'}^j) \right\}_{j=1}^{N}$$

↑ data points ↑ continuous "labels"

→ use mean-square loss (similar to regression)

loss: $\mathcal{L}(\varphi) = \frac{1}{2} \sum_{j=1}^{N} \| V_\varphi^\pi(s_j) - y_j \|^2$

→ unbiased estimator ✓

→ single sample estimates → large variance

2) bootstrap estimate (TD learning estimate)

$$V^\pi(s_t^j) \approx r(s_t^j, a_t^j) + \underbrace{\sum_{t'=t+1}^{T} \mathbb{E}_{\pi_\theta}\left[r(s_{t'}, a_{t'}) \mid s_{t+1}^j \right]}$$

$$\approx V^\pi(s_{t+1}^j) \approx V_\varphi^\pi(s_{t+1})$$

≈ neglect $\mathbb{E}_{p(s_{t+1}|s_t,a_t)}$

$V^\pi(s_{t+1}^j)$: don't know this value yet at step t

$V_\varphi^\pi(s_{t+1})$: plug in previous iteration estimate

⇓ biased estimator but lower variance

training data:

$$\left\{ s_t^j,\ y_t^j = r(s_t^j, a_t^j) + V_\varphi^\pi(s_{t+1}^j) \right\}_{j=1}^{N}$$

→ same loss as before

π_θ V_φ^π

— Algorithm: Offline Actor Critic method

(PG with value function estimation)

1. sample $\{s^j, a^j\}$ from π_θ → go until end of episode!
2. fit $V_\varphi^\pi(s)$ to the sampled data:

$$\mathcal{L}(\varphi) = \frac{1}{2} \sum_j \| V_\varphi^\pi(s_j) - y_j \|^2$$

a) MC estimate: $y_t^j = \sum_{t=t'}^{T} r(s_{t'}^j, a_{t'}^j)$

OR

b) bootstrap estimate: $y_t^j = r(s_t^j, a_t^j) + \gamma V_\varphi^\pi(s_{t+1}^j)$

3. evaluate advantage function:

$$A^\pi(s_t^j, a_t^j) \approx r(s_t^j, a_t^j) + \gamma V_\varphi^\pi(s_{t+1}^j) - V_\varphi^\pi(s_t^j)$$

4. compute the $\nabla_\theta J$:

$$\nabla_\theta J \approx \frac{1}{N} \sum_{j=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t^j | s_t^j) \cdot A^\pi(s_t^j, a_t^j)$$

5. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

- problem: keep adding values to $V^\pi(s)$
  → $V^\pi(s)$ could grow indefinitely

→ use a discount factor $\gamma \in [0, 1]$

$$y_t^i \approx r(s_t^i, a_t^i) + \gamma V_\varphi^\pi(s_{t+1}^i)$$

$$A^\pi(s_t^i, a_t^i) = r(s_t^i, a_t^i) + \gamma V_\varphi^\pi(s_{t+1}^i) - V_\varphi^\pi(s_t^i)$$

γ decreases reward-to-go
⇒ reduces variance

NB: in AC methods we need $\gamma < 1$
($\gamma \approx 0.99$) even in episodic tasks!

intuition: rewards in the immediate future count more than rewards in distant future

observation: bootstrap estimate does not require to finish episode in order to be computed

- Algorithm: Online Actor-Critic method

1. take action $a \sim \pi_\theta(a|s)$, get (s, a, r, s')

2. update V_φ^π using bootstrap/TD target

$y(s) = r + \gamma V_\varphi^\pi(s')$: $L(\varphi) = \frac{1}{2} \| V_\varphi^\pi(s) - y(s) \|$

3. estimate advantage function

$A^\pi(s,a) \approx r(s,a) + \gamma V_\varphi^\pi(s') - V_\varphi^\pi(s)$

4. $\nabla_\theta J \approx \nabla_\theta \log \pi_\theta(a|s) A^\pi(s,a)$ no sums over trajectories or time steps!

5. $\theta \leftarrow \theta + \alpha \nabla_\theta J$

- comparison AC vs. PG:

• AC:

$$\nabla_\theta J \approx \frac{1}{N} \sum_i \sum_t \nabla_\theta \log \pi_\theta(a_t^i | s_t^i) \times$$

$$\times \left[r(s_t^i, a_t^i) + \gamma V_\varphi^\pi(s_{t+1}^i) - V_\varphi^\pi(s_t^i) \right]$$

→ lower variance (due to critic V_φ^π)

→ biased (if critic is not perfect/exact)

· PG

$$\nabla_\theta J \approx \frac{1}{N} \sum_i \sum_t \nabla_\theta \log \pi_\theta(a_t^i | s_t^i) \cdot \left[\left\{ \sum_{t'=t}^{T} \gamma^{t'-t} r(s_{t'}^i, a_{t'}^i) \right\} - b \right]$$

→ no bias (MC target)

→ high variance (single-sample estimates)

• $b = V_\varphi^\pi(s)$: state-dep. estimate

→ no bias

→ lower variance (not as much as AC)

– practical implementation:

• need to parametrize both π_θ & V_φ^π

1) network architecture

i) two independent deep neural nets

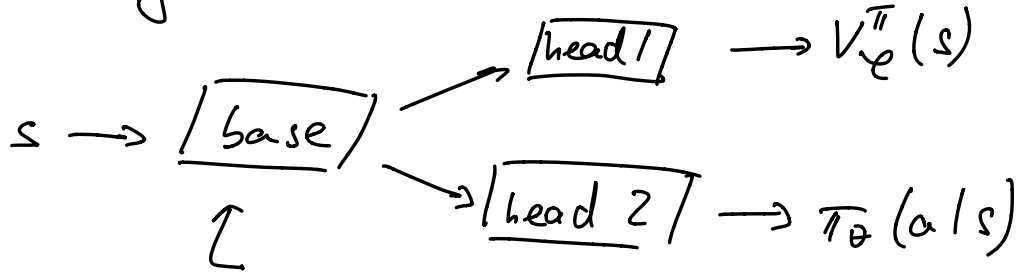
critic : $s \longrightarrow DNN_\varphi \longrightarrow V_\varphi^\pi(s)$

actor : $s \longrightarrow DNN_\theta \longrightarrow \pi_\theta(a|s)$

• stable training ✓

• no shared features ×

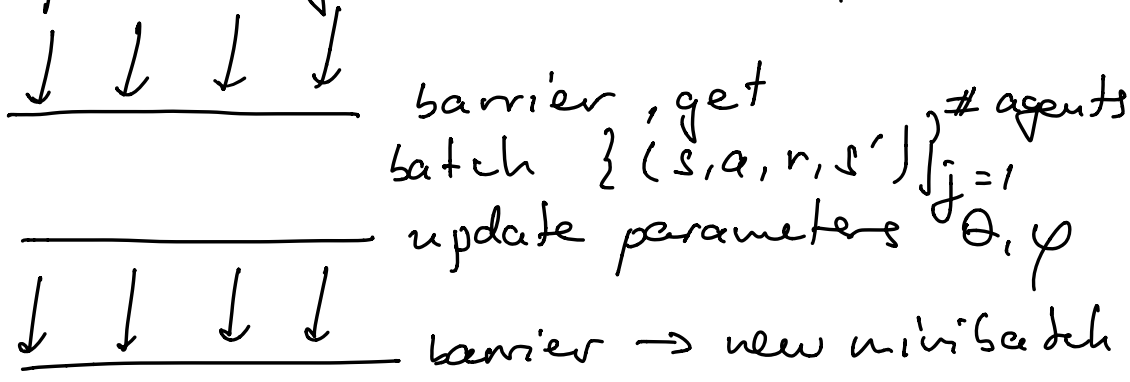
ii) single base + two heads



- shared features are possible ✓
- shared gradients in base layer can be of different scale for V_φ^π, π_θ
  → fine-tune learning rates

2) issue with online AC:
uses a minibatch size of 1!
→ use parallelization (JAX → pmap, vmap)

a) synchronized:
parallel agents, share π_θ, V_φ^π



5) asynchronous parallel AC
   -> Berkeley video lectures