Lecture 9
Advanced Policy Optimization

Policy gradient

objective: surrogated distribution over returns

$$J(\theta) = \int_{Tra} p_\theta(\tau) R(\tau) \, \mathrm{d}\tau \qquad J(\theta) = \int_S d^{\pi_\theta}(s) \int_A \pi_\theta(a|s) r(s,a) \, \mathrm{ds} \, \mathrm{da}$$

gradient:

$$\nabla_\theta J(\theta) = E[\nabla_\theta \log \pi_\theta(a|s) r(s,a)]$$

equivalent to $E[\sum_{i=1}^{T} \nabla_\theta \log \pi_\theta(a_i|s_i) R(s_i, a_i)]$

actor-critic:

$$\nabla_\theta J(\theta) = E[\nabla_\theta \log \pi_\theta(a|s) Q_w(s,a)]$$

subtract baseline

$$\nabla_\theta J(\theta) = E[\nabla_\theta \log \pi_\theta(a|s)(Q_w(s,a) - b(s))]$$

$$\nabla_\theta J(\theta) = E[\nabla_\theta \log \pi_\theta(a|s) A_w(s,a)]$$

Questions

1. on-policy or off-policy?

2. surrogate objective good or bad?

Going off-policy by importance sampling

use IS weight

$$J(\theta) = \int_{Tra} p_{\theta'}(\tau) \frac{p_\theta(\tau)}{p_{\theta'}(\tau)} R(\tau) \ \mathrm{d}\tau$$

gradient:

$$\nabla_\theta J(\theta) = E_{\pi_{\theta'}} [\sum_{t=0}^{\infty} \frac{p_\theta(\tau_t)}{p_{\theta'}(\tau_t)} \gamma^t \nabla_\theta \log \pi_\theta(a|s) A_w(s,a)]$$

$$\frac{p_\theta(\tau_t)}{p_{\theta'}(\tau_t)} = \prod_{i=0}^{t} \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta'}(a_t|s_t)}$$

Going off-policy by importance sampling

use IS weight:

$$J(\theta) = \int_S d^{\pi_{\theta'}}(s) \frac{d^{\pi_\theta}(s)}{d^{\pi_{\theta'}}(s)} \int_A \pi_{\theta'}(a|s) \frac{\pi_\theta(a|s)}{\pi_{\theta'}(a|s)} r(s,a) \text{ ds da}$$

$$d^\pi(s) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi)$$

gradient:

$$\nabla_\theta J(\theta) = E[\sum_{t=0}^{\infty} \frac{p_\theta(\tau_t)}{p_{\theta'}(\tau_t)} \gamma^t \nabla_\theta \log \pi_\theta(a|s) A_w(s,a)]$$

$$\frac{p_\theta(\tau_t)}{p_{\theta'}(\tau_t)} = \prod_{i=0}^{t} \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta'}(a_t|s_t)}$$

even larger variance ?

Policy relative performance

distribution shifted advantage

$$E_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^t A^{\pi}(s_t,a_t)\right]$$

$$= E_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^t\Big(r(s_t,a_t) + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t)\Big)\right]$$

$$J(\pi')$$

$$E_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\Big(\gamma^{t+1}V^{\pi}(s_{t+1}) - \gamma^t V^{\pi}(s_t)\Big)\right]$$

$$= -E_{\tau\sim\pi'}V^{\pi}(s_0) = -J(\pi)$$

relative performance: $$J(\pi') - J(\pi) = E_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^t A^{\pi}(s_t,a_t)\right]$$

Policy improvement objective

objective

$$\max_{\pi'} J(\pi')$$

$$\max_{\pi'} J(\pi') - J(\pi)$$

$$\max_{\pi'} E_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^t A^\pi(s_t,a_t)\right]$$

a good new policy is to maximize the advantage of the old policy

$$E_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^t A^\pi(s_t,a_t)\right] = \frac{1}{1-\gamma}E_{s\sim\pi'}E_{a\sim\pi'}\left[A^\pi(s_t,a_t)\right]$$

$$= \frac{1}{1-\gamma}E_{s\sim\pi'}E_{a\sim\pi}\left[\frac{\pi'(a_t|s_t)}{\pi(a_t|s_t)}A^\pi(s_t,a_t)\right]$$

what about $\frac{1}{1-\gamma}E_{s\sim\pi}E_{a\sim\pi}\left[\frac{\pi'(a_t|s_t)}{\pi(a_t|s_t)}A^\pi(s_t,a_t)\right]$

Distribution mismatch

if the policy is bounded $|\pi'(a|s) - \pi(a|s)| \leq \epsilon$

the state distribution is bounded

$$|d^{\pi'}(s) - d^{\pi}(s)| \leq 2\epsilon T$$

$$E_{s\sim\pi'} E_{a\sim\pi}\left[\frac{\pi'(a_t|s_t)}{\pi(a_t|s_t)} A^{\pi}(s_t, a_t)\right]$$

$$\geq E_{s\sim\pi} E_{a\sim\pi}\left[\frac{\pi'(a_t|s_t)}{\pi(a_t|s_t)} A^{\pi}(s_t, a_t)\right] - 2\epsilon T^2$$

Constrained objective

$$\arg\max_{\theta'} E_{s\sim\pi_\theta} E_{a\sim\pi_\theta}\left[\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} A^{\pi_\theta}(s_t, a_t)\right]$$

$$s.t. \quad |\pi_{\theta'}(a|s) - \pi_\theta(a|s)| \leq \epsilon$$

which (approximately) enquires monotonic increase of J

$$|\pi_{\theta'}(a|s) - \pi_\theta(a|s)| \leq \sqrt{\frac{1}{2} D_{KL}(\pi_{\theta'}(a|s)||\pi_\theta(a|s))}$$

$$D_{KL}(p(x)||q(x)) = E_{x\sim p(x)}\left[\log\frac{p(x)}{q(x)}\right] = E_{x\sim p(x)}\left[\log p(x) - \log q(x)\right]$$

Constrained objective

$$\bar{A}_\theta(\theta) = J(\theta)$$

$$\arg\max_{\theta'} E_{s\sim\pi_\theta} E_{a\sim\pi_\theta}\left[\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} A^{\pi_\theta}(s_t, a_t)\right] = \arg\max_{\theta'} \bar{A}_\theta(\theta')$$

$$s.t. \quad D_{KL}(\pi_{\theta'}(a|s)||\pi_\theta(a|s)) \leq \epsilon$$

approximation:

$$\arg\max_{\theta'} \nabla_\theta \bar{A}_\theta(\theta)(\theta' - \theta)$$

$$s.t. \quad D_{KL}(\pi_{\theta'}(a|s)||\pi_\theta(a|s)) \leq \epsilon$$

gradient?

Gradient and natural gradient

gradient decent $$-\frac{\nabla_\theta L(\theta)}{\|\nabla_\theta L(\theta)\|} = \lim_{\epsilon\to 0}\frac{1}{\epsilon}\arg\min_{\|\delta\theta\|\leq\epsilon} L(\theta+\delta\theta)$$

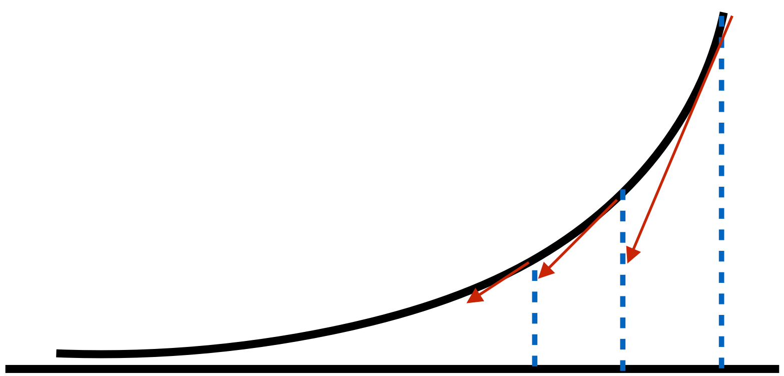
min with KL measure $$\min_{D_{KL}[p_\theta\|\theta+\delta\theta]\leq\epsilon} L(\theta+\delta\theta)$$

$$\min L(\theta+\delta\theta)+\lambda(D_{KL}[p_\theta\|p_{\theta+\delta\theta}]-\epsilon)$$

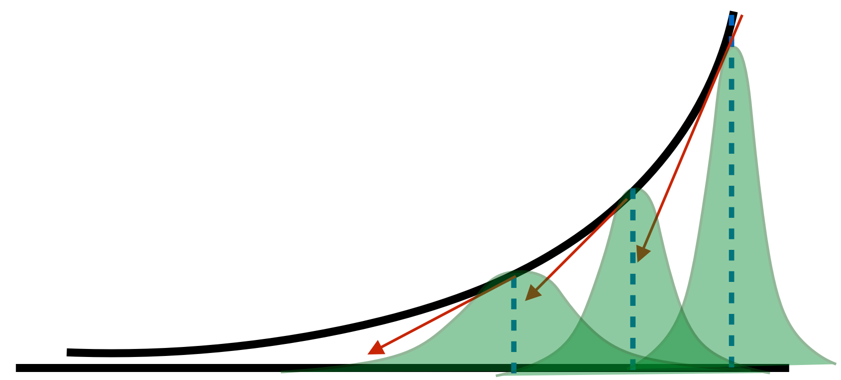
$$\approx \min L(\theta)+\nabla L(\theta)^\top\delta\theta+\lambda(\frac{1}{2}(\delta\theta)^\top F\delta\theta-\epsilon)$$

$$\delta\theta^* = -\frac{1}{\lambda}F^{-1}\nabla_\theta L(\theta)$$

Property of natural gradient



gradient in Euclidean distance



gradient on manifold

Natural Policy Gradient

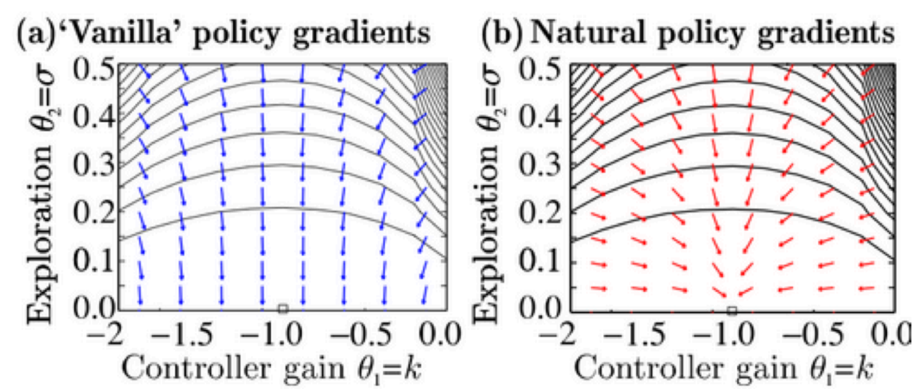
$$\arg\max_{\theta'} \nabla_\theta J(\theta)^T(\theta' - \theta)$$

$$\text{such that } D_{\mathrm{KL}}(\pi_{\theta'}(\mathbf{a}_t|\mathbf{s}_t)\|\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)) \leq \epsilon$$

$$D_{\mathrm{KL}}(\pi_{\theta'}\|\pi_\theta) \approx \frac{1}{2}(\theta' - \theta)^T\mathbf{F}(\theta' - \theta)$$

$$\theta' = \theta + \alpha\mathbf{F}^{-1}\nabla_\theta J(\theta) \qquad \alpha = \sqrt{\frac{2\epsilon}{\nabla_\theta J(\theta)^T\mathbf{F}\nabla_\theta J(\theta)}}$$

advantage: not effected by parameter space
disadvantage: unstable computation



(figure from Peters & Schaal 2008)

Natural Policy Gradient

1. collect sample by π
2. calculate policy gradient and Fisher matrix
3. update policy parameter by

$$\theta' = \theta + \alpha \mathbf{F}^{-1} \nabla_\theta J(\theta) \qquad \alpha = \sqrt{\frac{2\epsilon}{\nabla_\theta J(\theta)^T \mathbf{F} \nabla_\theta J(\theta)}}$$

Truncated Natural Policy Gradient, TNPG

use fixed step conjugate gradient

Trust Region Policy Optimization

we want to ensure

$$J(\theta') - J(\theta) \approx \bar{A}_\theta(\theta') \quad \text{to be positive}$$

$$\text{and} \quad D_{KL}(\pi_{\theta'}(a|s)||\pi_\theta(a|s)) \leq \epsilon$$

which NPG and TNPG may not

do a line search of parameter to ensure those

Trust Region Policy Optimization

monotonic improvement

$$\theta' \leftarrow \arg\max_{\theta'} \sum_t E_{\mathbf{s}_t \sim p_\theta(\mathbf{s}_t)} \left[E_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)} \left[\frac{\pi_{\theta'}(\mathbf{a}_t|\mathbf{s}_t)}{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)} \gamma^t A^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t) \right] \right]$$

$$\text{such that } D_{\mathrm{KL}}(\pi_{\theta'}(\mathbf{a}_t|\mathbf{s}_t) \| \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)) \leq \epsilon$$

$$\mathcal{L}(\theta', \lambda) = \sum_t E_{\mathbf{s}_t \sim p_\theta(\mathbf{s}_t)} \left[E_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)} \left[\frac{\pi_{\theta'}(\mathbf{a}_t|\mathbf{s}_t)}{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)} \gamma^t A^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t) \right] \right] - \lambda (D_{\mathrm{KL}}(\pi_{\theta'}(\mathbf{a}_t|\mathbf{s}_t) \| \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)) - \epsilon)$$

1. Maximize $\mathcal{L}(\theta', \lambda)$ with respect to θ' ← **can do this incompletely (for a few grad steps)**
2. $\lambda \leftarrow \lambda + \alpha (D_{\mathrm{KL}}(\pi_{\theta'}(\mathbf{a}_t|\mathbf{s}_t) \| \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)) - \epsilon)$

do a line search

Proximal Policy Optimization

more straightforward

$$\min_\theta \sum_{s,a} \frac{\pi_\theta(a|s)}{\pi_{old}(a|s)} A(s,a) - \beta D_{KL}[\pi_{old} \| \pi_\theta]$$

for t = 1, 2, …
run policy for T trajectories
estimate advantage function
do SGD to the policy optimization objective for N epochs
adjust β, KL too high => increase, KL too low=> decrease
next

$$\text{Compute } d = \hat{\mathbb{E}}_t[\mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]]$$

- If $d < d_{\text{targ}}/1.5$, $\beta \leftarrow \beta/2$
- If $d > d_{\text{targ}} \times 1.5$, $\beta \leftarrow \beta \times 2$

Proximal Policy Optimization 2

clip the IS ratio

$$\min_\theta \min\left(\frac{\pi_\theta(a|s)}{\pi_{old}(a|s)}A, \quad \textbf{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{old}(a|s)}, 1-\epsilon, 1+\epsilon\right)A\right)$$

algorithm	avg. normalized score
No clipping or penalty	-0.39
Clipping, $\epsilon = 0.1$	0.76
Clipping, $\epsilon = 0.2$	**0.82**
Clipping, $\epsilon = 0.3$	0.70
Adaptive KL $d_{\text{targ}} = 0.003$	0.68
Adaptive KL $d_{\text{targ}} = 0.01$	0.74
Adaptive KL $d_{\text{targ}} = 0.03$	0.71
Fixed KL, $\beta = 0.3$	0.62
Fixed KL, $\beta = 1.$	0.71
Fixed KL, $\beta = 3.$	0.72
Fixed KL, $\beta = 10.$	0.69

Benchmarking Deep Reinforcement Learning for Continuous Control

Yan Duan[†] ROCKYDUAN@EECS.BERKELEY.EDU
Xi Chen[†] C.XI@EECS.BERKELEY.EDU
Rein Houthooft[†‡] REIN.HOUTHOOFT@UGENT.BE
John Schulman[†§] JOSCHU@EECS.BERKELEY.EDU
Pieter Abbeel[†] PABBEEL@CS.BERKELEY.EDU
[†] University of California, Berkeley, Department of Electrical Engineering and Computer Sciences
[‡] Ghent University - iMinds, Department of Information Technology
§ OpenAI

Task	Random	REINFORCE	TNPG	RWR	REPS	TRPO	CEM	CMA-ES	DDPG
Cart-Pole Balancing	77.1 ± 0.0	4693.7 ± 14.0	**3986.4 ± 748.9**	**4861.5 ± 12.3**	565.6 ± 137.6	**4869.8 ± 37.6**	4815.4 ± 4.8	2440.4 ± 568.3	4634.4 ± 87.8
Inverted Pendulum*	−153.4 ± 0.2	13.4 ± 18.0	**209.7 ± 55.5**	84.7 ± 13.8	−113.3 ± 4.6	**247.2 ± 76.1**	38.2 ± 25.7	−40.1 ± 5.7	40.0 ± 244.6
Mountain Car	−415.4 ± 0.0	−67.1 ± 1.0	**-66.5 ± 4.5**	−79.4 ± 1.1	−275.6 ± 166.3	**-61.7 ± 0.9**	−66.0 ± 2.4	−85.0 ± 7.7	−288.4 ± 170.3
Acrobot	−1904.5 ± 1.0	−508.1 ± 91.0	−395.8 ± 121.2	−352.7 ± 35.9	−1001.5 ± 10.8	−326.0 ± 24.4	−436.8 ± 14.7	−785.6 ± 13.1	**-223.6 ± 5.8**
Double Inverted Pendulum*	149.7 ± 0.1	4116.5 ± 65.2	**4455.4 ± 37.6**	3614.8 ± 368.1	446.7 ± 114.8	**4412.4 ± 50.4**	2566.2 ± 178.9	1576.1 ± 51.3	2863.4 ± 154.0
Swimmer*	−1.7 ± 0.1	92.3 ± 0.1	**96.0 ± 0.2**	60.7 ± 5.5	3.8 ± 3.3	**96.0 ± 0.2**	68.8 ± 2.4	64.9 ± 1.4	85.8 ± 1.8
Hopper	8.4 ± 0.0	714.0 ± 29.3	**1155.1 ± 57.9**	553.2 ± 71.0	86.7 ± 17.6	**1183.3 ± 150.0**	63.1 ± 7.8	20.3 ± 14.3	267.1 ± 43.5
2D Walker	−1.7 ± 0.0	506.5 ± 78.8	**1382.6 ± 108.2**	136.0 ± 15.9	−37.0 ± 38.1	**1353.8 ± 85.0**	84.5 ± 19.2	77.1 ± 24.3	318.4 ± 181.6
Half-Cheetah	−90.8 ± 0.3	1183.1 ± 69.2	**1729.5 ± 184.6**	376.1 ± 28.2	34.5 ± 38.0	**1914.0 ± 120.1**	330.4 ± 274.8	441.3 ± 107.6	**2148.6 ± 702.7**
Ant*	13.4 ± 0.7	548.3 ± 55.5	**706.0 ± 127.7**	37.6 ± 3.1	39.0 ± 9.8	**730.2 ± 61.3**	49.2 ± 5.9	17.8 ± 15.5	326.2 ± 20.8
Simple Humanoid	41.5 ± 0.2	128.1 ± 34.0	**255.0 ± 24.5**	93.3 ± 17.4	28.3 ± 4.7	**269.7 ± 40.3**	60.6 ± 12.9	28.7 ± 3.9	99.4 ± 28.1
Full Humanoid	13.2 ± 0.1	262.2 ± 10.5	**288.4 ± 25.2**	46.7 ± 5.6	41.7 ± 6.1	**287.0 ± 23.4**	36.9 ± 2.9	N/A ± N/A	119.0 ± 31.2
Cart-Pole Balancing (LS)*	77.1 ± 0.0	420.9 ± 265.5	**945.1 ± 27.8**	68.9 ± 1.5	898.1 ± 22.1	**960.2 ± 46.0**	227.0 ± 223.0	68.0 ± 1.6	
Inverted Pendulum (LS)	−122.1 ± 0.1	−13.4 ± 3.2	**0.7 ± 6.1**	−107.4 ± 0.2	−87.2 ± 8.0	**4.5 ± 4.1**	−81.2 ± 33.2	−62.4 ± 3.4	
Mountain Car (LS)	−83.0 ± 0.0	−81.2 ± 0.6	**-65.7 ± 9.0**	−81.7 ± 0.1	−82.6 ± 0.4	**-64.2 ± 9.5**	**-68.9 ± 1.3**	**-73.2 ± 0.6**	
Acrobot (LS)*	−393.2 ± 0.0	−128.9 ± 11.6	**-84.6 ± 2.9**	−235.9 ± 5.3	−379.5 ± 1.4	**-83.3 ± 9.9**	−149.5 ± 15.3	−159.9 ± 7.5	
Cart-Pole Balancing (NO)*	101.4 ± 0.1	616.0 ± 210.8	**916.3 ± 23.0**	93.8 ± 1.2	99.6 ± 7.2	606.2 ± 122.2	181.4 ± 32.1	104.4 ± 16.0	
Inverted Pendulum (NO)	−122.2 ± 0.1	6.5 ± 1.1	**11.5 ± 0.5**	−110.0 ± 1.4	−119.3 ± 4.2	**10.4 ± 2.2**	−55.6 ± 16.7	−80.3 ± 2.8	
Mountain Car (NO)	−83.0 ± 0.0	−74.7 ± 7.8	**-64.5 ± 8.6**	−81.7 ± 0.1	−82.9 ± 0.1	**-60.2 ± 2.0**	−67.4 ± 1.4	−73.5 ± 0.5	
Acrobot (NO)*	−393.5 ± 0.0	**-186.7 ± 31.3**	**-164.5 ± 13.4**	−233.1 ± 0.4	−258.5 ± 14.0	**-149.6 ± 8.6**	−213.4 ± 6.3	−236.6 ± 6.2	
Cart-Pole Balancing (SI)*	76.3 ± 0.1	431.7 ± 274.1	**980.5 ± 7.3**	69.0 ± 2.8	702.4 ± 196.4	**980.3 ± 5.1**	746.6 ± 93.2	71.6 ± 2.9	
Inverted Pendulum (SI)	−121.8 ± 0.2	−5.3 ± 5.6	**14.8 ± 1.7**	−108.7 ± 4.7	−92.8 ± 23.9	**14.1 ± 0.9**	−51.8 ± 10.6	−63.1 ± 4.8	
Mountain Car (SI)	−82.7 ± 0.0	−63.9 ± 0.2	**-61.8 ± 0.4**	−81.4 ± 0.1	−80.7 ± 2.3	**-61.6 ± 0.4**	−63.9 ± 1.0	−66.9 ± 0.6	
Acrobot (SI)*	−387.8 ± 1.0	**-169.1 ± 32.3**	**-156.6 ± 38.9**	−233.2 ± 2.6	−216.1 ± 7.7	**-170.9 ± 40.3**	−250.2 ± 13.7	−245.0 ± 5.5	
Swimmer + Gathering	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Ant + Gathering	−5.8 ± 5.0	−0.1 ± 0.1	−0.4 ± 0.1	−5.5 ± 0.5	−6.7 ± 0.7	−0.4 ± 0.0	−4.7 ± 0.7	N/A ± N/A	−0.3 ± 0.3
Swimmer + Maze	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Ant + Maze	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	N/A ± N/A	0.0 ± 0.0

IMPLEMENTATION MATTERS IN DEEP POLICY GRADIENTS: A CASE STUDY ON PPO AND TRPO

Logan Engstrom[*], Andrew Ilyas[*], Shibani Santurkar[1], Dimitris Tsipras[1],
Firdaus Janoos[2], Larry Rudolph[1,2], and Aleksander Mądry[1]

[1]MIT [2]Two Sigma
{engstrom,ailyas,shibani,tsipras,madry}@mit.edu
rudolph@csail.mit.edu, firdaus.janoos@twosigma.com

https://openreview.net/forum?id=r1etN1rtPB

What Matters In On-Policy Reinforcement Learning? A Large-Scale Empirical Study

**Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini,
Sertan Girgin, Raphael Marinier, Léonard Hussenot, Matthieu Geist,
Olivier Pietquin, Marcin Michalski, Sylvain Gelly, Olivier Bachem**

Google Research, Brain Team

https://arxiv.org/pdf/2006.05990.pdf

Another surrogate objective

previous (stochastic) objective and gradient

$$J(\theta) = \int_S d^{\pi_\theta}(s) \int_A \pi_\theta(a|s) Q^\pi(s,a) \text{ ds da}$$

$$\nabla_\theta J(\theta) = E[\nabla_\theta \log \pi_\theta(a|s) Q_w(s,a)]$$

where we treat *Q* the same as *R* as a black box

Can we treat Q_w as a differentiable model?

Deterministic policy gradient: DPG

$$J(\theta) = \int_S d^{\pi_\theta}(s) Q^{\pi}(s, \pi_\theta(s)) \text{ ds da}$$

$$\nabla_\theta J(\theta) = E_{d^{\pi_\theta}}[\nabla_\theta \pi_\theta(s) \nabla_a Q_w(s,a)|_{a=\pi(s)}]$$

on-policy update

$$\delta_t = r_t + \gamma Q^w(s_{t+1}, a_{t+1}) - Q^w(s_t, a_t)$$
$$w_{t+1} = w_t + \alpha_w \delta_t \nabla_w Q^w(s_t, a_t)$$
$$\theta_{t+1} = \theta_t + \alpha_\theta \nabla_\theta \mu_\theta(s_t) \; \nabla_a Q^w(s_t, a_t)|_{a=\mu_\theta(s)}$$

off-policy update

$$\delta_t = r_t + \gamma Q^w(s_{t+1}, \mu_\theta(s_{t+1})) - Q^w(s_t, a_t)$$
$$w_{t+1} = w_t + \alpha_w \delta_t \nabla_w Q^w(s_t, a_t)$$
$$\theta_{t+1} = \theta_t + \alpha_\theta \nabla_\theta \mu_\theta(s_t) \; \nabla_a Q^w(s_t, a_t)|_{a=\mu_\theta(s)}$$

[David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, Martin A. Riedmiller. Deterministic Policy Gradient Algorithms. ICML 2014: 387-395]

Deep deterministic policy gradient: DDPG

Algorithm 1 DDPG algorithm

Randomly initialize critic network $Q(s,a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ.
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R
for episode = 1, M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for t = 1, T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
    Execute action a_t and observe reward r_t and observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in R
    Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
    Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$$

    Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

  end for
end for

[Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, Daan Wierstra. Continuous control with deep reinforcement learning. ICLR 2016]

Twin delayed DDPG: TD3

Algorithm 1 TD3

Initialize critic networks Q_{θ_1}, Q_{θ_2}, and actor network π_ϕ with random parameters θ_1, θ_2, ϕ
Initialize target networks $\theta'_1 \leftarrow \theta_1$, $\theta'_2 \leftarrow \theta_2$, $\phi' \leftarrow \phi$
Initialize replay buffer $\mathcal{B}$
for $t = 1$ **to** T **do**
  Select action with exploration noise $a \sim \pi_\phi(s) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma)$ and observe reward r and new state s'
  Store transition tuple (s, a, r, s') in $\mathcal{B}$

  Sample mini-batch of N transitions (s, a, r, s') from $\mathcal{B}$
  $\tilde{a} \leftarrow \pi_{\phi'}(s') + \epsilon, \quad \epsilon \sim \text{clip}(\mathcal{N}(0, \tilde{\sigma}), -c, c)$
  $y \leftarrow r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \tilde{a})$
  Update critics $\theta_i \leftarrow \text{argmin}_{\theta_i} N^{-1} \sum (y - Q_{\theta_i}(s, a))^2$
  if t mod d **then**
    Update ϕ by the deterministic policy gradient:
    $\nabla_\phi J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$
    Update target networks:
    $\theta'_i \leftarrow \tau\theta_i + (1 - \tau)\theta'_i$
    $\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$
  end if
end for

Twin:
consider Q overestimation

delayed update +
smoothed update

[Fujimoto, Scott, Herke van Hoof, and Dave Meger. Addressing Function Approximation Error in Actor-Critic Methods. arXiv:1802.09477 (2018)]

Twin delayed DDPG: TD3

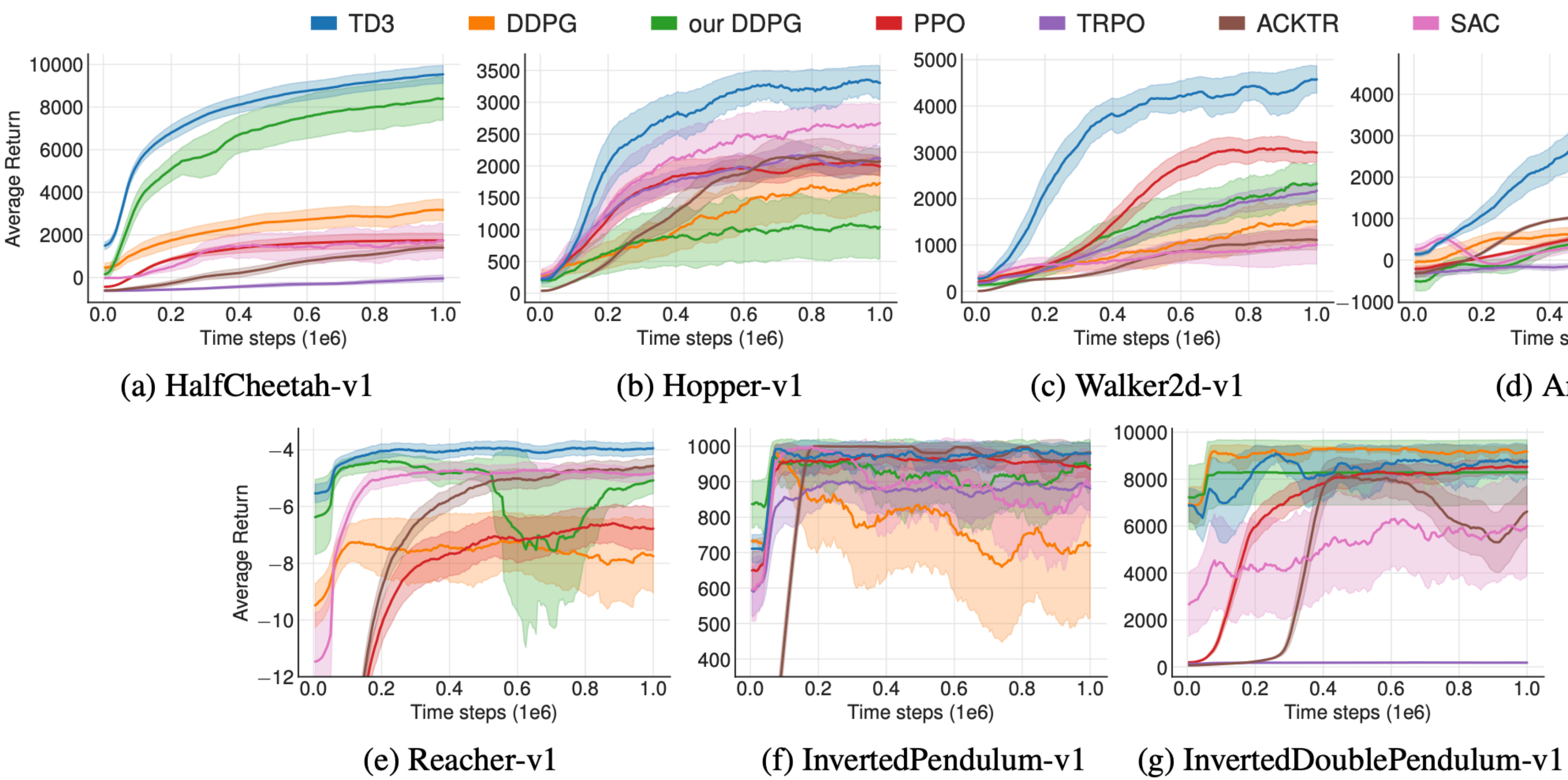


Figure 5. Learning curves for the OpenAI gym continuous control tasks. The shaded region represents half a standard deviation of the average evaluation over 10 trials. Curves are smoothed uniformly for visual clarity.

[Fujimoto, Scott, Herke van Hoof, and Dave Meger. Addressing Function Approximation Error in Actor-Critic Methods. arXiv:1802.09477 (2018)]

Learning with MaxEntropy

$$J(\theta) = E_{(s,a)\sim d^{\pi_\theta}}[Q^{\pi_\theta}(s,a) + \alpha H(\pi_\theta(\cdot|s))]$$

better robustness + natural exploration

Maximum entropy learning

consider a distribution *p* of *x*, and a cost function

$$\sum_x p(x) = 1 \qquad\qquad c(x)$$

we want to find a distribution *p* that maximizes the cost

$$\arg\max_p \sum_x p(x)c(x)$$

would overfit the largest cost. Thus introduce a regularization

$$\arg\max_p \sum_x p(x)c(x) - \lambda \sum_x p(x) \log p(x)$$

Maximum entropy learning

$$L(p,\lambda) = \sum_x p(x)c(x) - \lambda \sum_x p(x) \log p(x)$$

find the solution

$$\partial_p L = \sum_x [c(x) - \lambda \log p(x) - \lambda] = 0$$

$$p(x) \sim e^{c(x)}$$

exponential family

maximizing the entropy of the distribution implies maximizing the likelihood of exponential family distribution

MaxEntropy policy improvement

The policy should be in the form of

$$\pi \sim \frac{e^{Q_t(s,\cdot)}}{Z_t(s)}$$

Set the policy update objective

$$\theta_{t+1} = \arg\min_{\theta} D_{KL}\left(\pi_\theta(\cdot|s) \middle\| \frac{e^{Q_t(s,\cdot)}}{Z_t(s)}\right)$$

Can the policy be improved?

$$\pi \sim \frac{e^{Q_t(s,\cdot)}}{Z_t(s)} = \mathrm{softmax}\ Q_t(s,\cdot)$$

About soft Q learning

1. define new reward $r_\pi(\mathbf{s}_t, \mathbf{a}_t) \triangleq r(\mathbf{s}_t, \mathbf{a}_t) + \mathbb{E}_{\mathbf{s}_{t+1}\sim p}\left[\mathcal{H}\left(\pi(\,\cdot\,|\mathbf{s}_{t+1})\right)\right]$

$$Q(\mathbf{s}_t, \mathbf{a}_t) \leftarrow r_\pi(\mathbf{s}_t, \mathbf{a}_t) + \gamma\, \mathbb{E}_{\mathbf{s}_{t+1}\sim p, \mathbf{a}_{t+1}\sim\pi}\left[Q(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})\right]$$

2. the policy objective

$$\begin{aligned}\pi_{\text{new}}(\,\cdot\,|\mathbf{s}_t) &= \arg\min_{\pi'\in\Pi} \mathrm{D}_{\text{KL}}\left(\pi'(\,\cdot\,|\mathbf{s}_t)\;\middle\|\;\exp\left(Q^{\pi_{\text{old}}}(\mathbf{s}_t,\,\cdot\,) - \log Z^{\pi_{\text{old}}}(\mathbf{s}_t)\right)\right)\\ &= \arg\min_{\pi'\in\Pi} J_{\pi_{\text{old}}}(\pi'(\,\cdot\,|\mathbf{s}_t)).\end{aligned}$$

ensures that $J_{\pi_{\text{old}}}(\pi_{\text{new}}(\,\cdot\,|\mathbf{s}_t)) \le J_{\pi_{\text{old}}}(\pi_{\text{old}}(\,\cdot\,|\mathbf{s}_t))$

which is

$$\mathbb{E}_{\mathbf{a}_t\sim\pi_{\text{new}}}\left[\log\pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t) - Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t) + \log Z^{\pi_{\text{old}}}(\mathbf{s}_t)\right] \le \mathbb{E}_{\mathbf{a}_t\sim\pi_{\text{old}}}\left[\log\pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t) - Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t) + \log Z^{\pi_{\text{old}}}(\mathbf{s}_t)\right]$$

and also is $\mathbb{E}_{\mathbf{a}_t\sim\pi_{\text{new}}}\left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t) - \log\pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t)\right] \ge V^{\pi_{\text{old}}}(\mathbf{s}_t)$

3. we obtain the value improvement by expanding and replacing as

$$\begin{aligned}Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t) &= r(\mathbf{s}_t, \mathbf{a}_t) + \gamma\, \mathbb{E}_{\mathbf{s}_{t+1}\sim p}\left[V^{\pi_{\text{old}}}(\mathbf{s}_{t+1})\right]\\ &\le r(\mathbf{s}_t, \mathbf{a}_t) + \gamma\, \mathbb{E}_{\mathbf{s}_{t+1}\sim p}\left[\mathbb{E}_{\mathbf{a}_{t+1}\sim\pi_{\text{new}}}\left[Q^{\pi_{\text{old}}}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) - \log\pi_{\text{new}}(\mathbf{a}_{t+1}|\mathbf{s}_{t+1})\right]\right]\\ &\;\;\vdots\\ &\le Q^{\pi_{\text{new}}}(\mathbf{s}_t, \mathbf{a}_t),\end{aligned}$$

SAC

Algorithm 1 Soft Actor-Critic

Initialize parameter vectors ψ, $\bar{\psi}$, θ, ϕ.
for each iteration **do**
  for each environment step **do**
    $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$
  end for
  for each gradient step **do**
    $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
    $\bar{\psi} \leftarrow \tau\psi + (1 - \tau)\bar{\psi}$
  end for
end for

$$J_V(\psi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}}\left[\tfrac{1}{2}\left(V_\psi(\mathbf{s}_t) - \mathbb{E}_{\mathbf{a}_t \sim \pi_\phi}\left[Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \log \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)\right]\right)^2\right]$$

$$J_Q(\theta) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \mathcal{D}}\left[\frac{1}{2}\left(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \hat{Q}(\mathbf{s}_t, \mathbf{a}_t)\right)^2\right]$$

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}}\left[\mathrm{D}_{\mathrm{KL}}\left(\pi_\phi(\,\cdot\,|\mathbf{s}_t)\,\middle\|\,\frac{\exp\left(Q_\theta(\mathbf{s}_t, \,\cdot\,)\right)}{Z_\theta(\mathbf{s}_t)}\right)\right]$$