Baselines and ER in policy gradient learning

Nikita Petrenko

November 7, 2017

Overview

Common methods of PG learning

Discounted state distribution: $\rho_\pi := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P(s_0 \to s|t)$

- A3C: $\nabla_\theta V(s) = E_{\rho_\pi,\pi}(\nabla \log \pi_\theta(a) * (r(s,a) + \gamma V(s') - V(s)))$ basic algorithm which exhibits high gradient variance and inability to learn on off-policy data (including Experience Replay)
- A3C with Importance Sampling (IS):

$$\nabla_\theta V(s) = E_{traj\sim\pi_{\theta'}}\left[\frac{P(traj)}{P'(traj)}\nabla \log \pi_\theta(a)(r(s,a) + \gamma V(s') - V(s))\right]$$

  Unbiased estimation of policy gradient with off-policy data. However, it suffers from possibly infinite variance of density ratios if behavioral policy (the one that collected samples) and agent policy are too different.
  In MDP setting,

$$\frac{P(traj)}{P'(traj)} = \frac{p(s_0)\pi(a_0|s_0)p(s_1|a_0,s_0)...}{p(s_0)\pi'(a_0|s_0)p(s_1|a_0,s_0)...} = \prod_{t=0}^{T-1}\frac{\pi(a_t|s_t)}{\pi'(a_t|s_t)} \tag{1}$$

Common methods of PG learning

- TRPO: derives lower bound on policy improvement thus allowing to make several gradient update steps on sampled trajectories. Improves sample efficiency significantly.

Q-prop

Main features:

- Unbiased, low variance gradient
- On-policy actor with control variate
- Off-policy critic
- Ability to use TRPO for policy updates

Limitations:

- Continuous control only

Q-prop estimator

Variance reduction through action-dependent control variates

$\overline{f}(x,a) = f(x,\overline{a}) + \nabla_a f(x,a)(a-\overline{a})$

$\mu_\theta(s) = E_{\pi_\theta(a|s)}(a)$

Theorem (Q-prop gradient)

$\forall f, \overline{a}, \eta$

$$\nabla_\theta J = E_{\rho_\pi,\pi}\left[\nabla_\theta log \pi_\theta(a_t|s_t)(Q(s_t,a_t) - \eta(s_t)\overline{f}(s_t,a_t))\right] + \\ + E_{\rho_\pi}\left[\eta(s_t)\nabla_a f(s_t,a)|_{a=\overline{a_t}}\nabla_\theta\mu_\theta(s_t)\right] \quad (2)$$

Suggested choice:

- $f = Q_w$ – off-policy critic, same as in DDPG
- $\overline{a} = \mu_\theta(s_t)$

Compare it to DDPG policy gradient:

$E_{data}\left[\nabla_a Q_w(s_t,a)|_{a=\mu_\theta(s_t)}\nabla_\theta\mu_\theta(s_t)\right] = E_{data}\left[\nabla_\theta Q_w(s_t,\mu_\theta(s_t))\right]$

Proof

$$E_{\rho_\pi,\pi}\left[\nabla_\theta log\pi(a_t|s_t)\overline{f}(s_t,a_t)\right] = \tag{3}$$

$$= E_{\rho_\pi,\pi}\left[\nabla_\theta log\pi(a_t|s_t)(f(s_t,\overline{a_t}) + \nabla_a f(s_t,a)|_{a=\overline{a_t}}(a_t - \overline{a_t}))\right] \tag{4}$$

$$= E_{\rho_\pi,\pi}\left[\nabla_\theta log\pi(a_t|s_t)(\nabla_a f(s_t,a)|_{a=\overline{a_t}} a_t)\right] \tag{5}$$

$$= E_{\rho_\pi}\left[\int_A \nabla_\theta \pi(a_t|s_t)(\nabla_a f(s_t,a)|_{a=\overline{a_t}} a_t) da_t\right] \tag{6}$$

$$= E_{\rho_\pi}\left[\nabla_a f(s_t,a)|_{a=\overline{a_t}} \int_A \nabla_\theta \pi(a_t|s_t) a_t da_t\right] \tag{7}$$

$$= E_{\rho_\pi}\left[\nabla_a f(s_t,a)|_{a=\overline{a_t}} \nabla_\theta E_{\pi(a_t|s_t)} a_t\right] = E_{\rho_\pi}\left[\nabla_a f(s_t,a)|_{a=\overline{a_t}} \nabla_\theta \mu_\theta(s_t)\right] \tag{8}$$

Variance analysis

Our final gradient estimator:

$$\nabla_\theta J = E_{\rho_\pi,\pi}\left[\nabla_\theta log\pi_\theta(a_t|s_t)(A(s_t,a_t)-\eta(s_t)\overline{A_w}(s_t,a_t))\right] + \\ + E_{\rho_\pi}\left[\eta(s_t)\nabla_a Q_w(s_t,a)|_{a=\mu_\theta(s_t)}\nabla_\theta\mu_\theta(s_t)\right]$$

Authors analyze

$$\begin{aligned} Var^* &= E_{\rho_\pi}\left[Var_{a_t}(A(s_t,a_t)-\eta(s_t)\overline{A_w}(s_t,a_t))\right] \\ &= Var + E_{\rho_\pi}\left[\eta(s_t)Cov_{a_t}(A(s_t,a_t),\overline{A}(s_t,a_t)) + \eta^2(s_t)Var(\overline{A}(s_t,a_t))\right] \end{aligned}$$

$$\begin{aligned} Cov_{a_t}(A(s_t,a_t),\overline{A}(s_t,a_t)) &= E_\pi(A(s_t,a_t)\overline{A}(s_t,a_t)) \\ Var_{a_t}(\overline{A}(s_t,a_t)) &= E_\pi(\overline{A}^2(s_t,a_t)) \\ &= \nabla_a Q_w(s_t,a)|^T_{a=\mu_\theta(s_t)}\Sigma_\theta(s_t)\nabla_a Q_w(s_t,a)|_{a=\mu_\theta(s_t)} \end{aligned}$$

where $\Sigma_\theta(s_t)$ is a covariance matrix of stochastic policy π_θ at state s_t
Therefore, optimal $\eta^*(s_t) = Cov(A,\overline{A})/Var(\overline{A})$ can be approximated with single sample

Choice of η

- $\eta(s_t) = Cov(A, \overline{A})/Var(\overline{A})$ leads to Adaptive Q-prop. However, its variance can be big itself if we're using single sample estimation of Cov.
- Conservative Q-prop:
  $\eta(s_t) = \begin{cases} 1 & Cov > 0 \\ 0 & \ldots \end{cases}$
- Aggressive Q-prop:
  $\eta(s_t) = sgn(Cov)$

Other form of control variate

Actually we don't have to restrict ourselves to using first order Taylor expansion of critic by observing that:

$$E_{\pi} \nabla_{\theta} log \pi Q_w(s, a) = \nabla_{\theta} E_{\pi} Q_w(s, a)$$

In discrete action spaces $\nabla_{\theta} E_{\pi} ...$ can be estimated in analytic form, in continuous one would have to use reparametrization trick
The gradient estimate then becomes:

$$\nabla_{\theta} J = E_{\rho_{\pi}, \pi} \left[\nabla_{\theta} log \pi_{\theta}(a_t|s_t)(A(s_t, a_t) - \eta(s_t) A_w(s_t, a_t)) \right] + \\ + E_{\rho_{\pi}} \left[\eta(s_t) \nabla_{\theta} E_{\pi} Q_w(s_t, a) \right]$$

Value function estimation

Q-function estimation for control variate:

$$T_\pi[Q](s,a) := r(s,a) + E_\pi[Q(s',a')|s,a]$$

$$w \leftarrow argmin_w \| T_\pi[Q_{w'}] - Q_w \|_2$$
$$w' \leftarrow \tau w' + (1-\tau) w, \tau = 0.999$$

Value function estimation:

$$\phi \leftarrow argmin \sum \| V_\phi(s_n) - (r + V_\phi(s'_n)) \|_2;$$

$$\text{subj. to } \frac{1}{N}\sum_{n=1}^{N} \frac{\| V_\phi(s_n) - V_{\phi_{old}}(s_n) \|_2}{2\sigma^2} < \epsilon$$
$$(s_n, s'_n) \sim MDP_\pi$$

Continuous control of bias-variance tradeoff (GAE(λ)):

$$Q^\lambda(s_t, a_t) =$$
$$(1-\lambda)\sum_{k=0}^{\infty}\lambda^k\left[\sum_{m=0}^{k-1}\gamma^m r(s_{t+m}, a_{t+m}) + \gamma^k V_\phi(s_{t+k}, a_{t+k})\right]$$
$$\lambda \to 1 \Rightarrow Q^\lambda \to R$$
$$\lambda \to 0 \Rightarrow Q^\lambda \to V_\phi$$

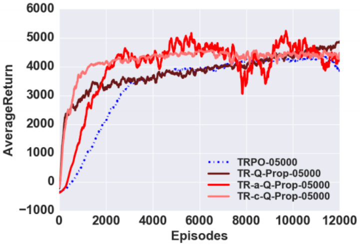
Advantage for CV:

$$\overline{A_w}(s_t, a_t) = (a - \mu(\theta))\nabla_a Q(s_t, a)|_{a=\mu(\theta)}$$

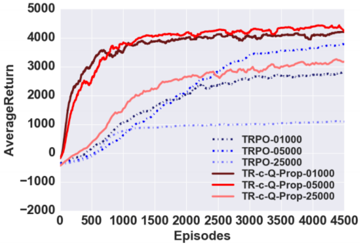
Algorithm

Algorithm 1 Adaptive Q-Prop

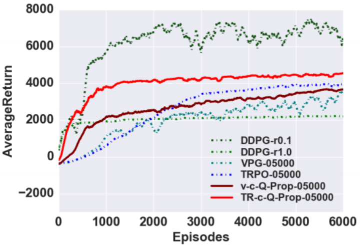
Initialize w for critic Q_w, θ for stochastic policy π_θ, and replay buffer $\mathscr{R} \leftarrow \emptyset$.
repeat
  for $e = 1, \dots, E$ **do** ▷ Collect E episodes of on-policy experience using π_θ
    $\boldsymbol{s}_{0,e} \sim p(\boldsymbol{s}_0)$
    for $t = 0, \dots, T-1$ **do**
      $\boldsymbol{a}_{t,e} \sim \pi_\theta(\cdot|\boldsymbol{s}_{t,e})$, $\boldsymbol{s}_{t+1,e} \sim p(\cdot|\boldsymbol{s}_{t,e}, \boldsymbol{a}_{t,e})$, $r_{t,e} = r(\boldsymbol{s}_{t,e}, \boldsymbol{a}_{t,e})$
  Add batch data $\mathscr{B} = \{\boldsymbol{s}_{0:T,1:E}, \boldsymbol{a}_{0:T-1,1:E}, r_{0:T-1,1:E}\}$ to replay buffer $\mathscr{R}$
  Take $E \cdot T$ gradient steps on Q_w using $\mathscr{R}$ and π_θ
  Fit $V_\phi(\boldsymbol{s}_t)$ using $\mathscr{B}$
  Compute $\hat{A}_{t,e}$ using GAE(λ) and $\bar{A}_{t,e}$ using Eq. 7
  Set $\eta_{t,e}$ based on Section 3.2
  Compute and center the learning signals $l_{t,e} = \hat{A}_{t,e} - \eta_{t,e}\bar{A}_{t,e}$
  Compute $\nabla_\theta J(\theta) \approx \frac{1}{ET} \sum_e \sum_t \nabla_\theta \log \pi_\theta(\boldsymbol{a}_{t,e}|\boldsymbol{s}_{t,e}) l_{t,e} + \eta_{t,e} \nabla_{\boldsymbol{a}} Q_w(\boldsymbol{s}_{t,e}, \boldsymbol{a})|_{\boldsymbol{a}=\boldsymbol{\mu}_\theta(\boldsymbol{s}_{t,e})} \nabla_\theta \boldsymbol{\mu}_\theta(\boldsymbol{s}_{t,e})$
  Take a gradient step on π_θ using $\nabla_\theta J(\theta)$, optionally with a trust-region constraint using $\mathscr{B}$
until π_θ converges.



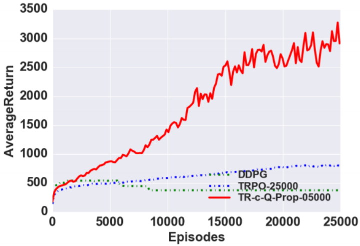
(a) Standard Q-Prop vs adaptive variants.



(b) Conservative Q-Prop vs TRPO across batch sizes.



(a) Comparing algorithms on HalfCheetah-v1.



(b) Comparing algorithms on Humanoid-v1.

		TR-c-Q-Prop		TRPO		DDPG	
Domain	Threshold	MaxReturn.	Episodes	MaxReturn	Epsisodes	MaxReturn	Episodes
Ant	3500	3534	**4975**	**4239**	13825	957	N/A
HalfCheetah	4700	4811	20785	4734	26370	**7490**	**600**
Hopper	2000	**2957**	5945	2486	5715	2604	**965**
Humanoid	2500	>**3492**	**14750**	918	>30000	552	N/A
Reacher	-7	**-6.0**	**2060**	-6.7	2840	-6.6	**1800**
Swimmer	90	103	2045	110	3025	**150**	**500**
Walker	3000	**4030**	3685	3567	18875	3626	**2125**

Table 1: Q-Prop, TRPO and DDPG results showing the max average rewards attained in the first 30k episodes and the episodes to cross specific reward thresholds. Q-Prop often learns more sample efficiently than TRPO and can solve difficult domains such as Humanoid better than DDPG.

Unifying Policy Gradient And Actor-Critic

Proposed extention:

$$\nabla_\theta J \simeq \alpha E_{\rho_\pi,\pi}\left[\nabla_\theta log \pi_\theta(a_t|s_t)(A(s_t,a_t) - \eta\overline{A_w}(s_t,a_t))\right] + \\ + \eta E_{\rho_{CR}}\left[\nabla_a Q_w(s_t,a)|_{a=\mu_\theta(s_t)} \nabla_\theta \mu_\theta(s_t)\right]$$

Parameter	Implementation options	Introduce bias?
Q_w	off-policy TD; on-policy TD(λ); model-based; etc.	No
V_ϕ	on-policy Monte Carlo fitting; $\mathbb{E}_{\pi_\theta}[Q_w(\boldsymbol{s}_t, \boldsymbol{a}_t)]$; etc	No
λ	$0 \leq \lambda \leq 1$	Yes, except $\lambda = 1$
α	$\alpha \geq 0$	Yes, except $\alpha = 1$
η	any η	No
ρ_{CR}	ρ of any policy	Yes, except $\rho_{CR} = \rho_\pi$

Retrace

Consider Bellman operator:
$T_\pi Q(s,a) = r(s,a) + \gamma E_\pi Q(s',a')$

Projection operator:
$P_{\Omega,\alpha} Q = argmin_{x \in \Omega} \|x - Q\|_\alpha$

- $\forall \alpha \in [1, \infty], \|TQ - TQ'\|_\alpha \leq \gamma \|Q - Q'\|_\alpha, T_\pi Q_\pi = Q_\pi$
- Then $P_{\Omega,2} T$ is also a γ-contraction mapping with supposedly appropriate stationary point

So why bother?

Let:

- F - γ-contraction with fixed point Q^*
- $Q : \|FQ - Q\| \leq \epsilon$

Then $\|Q^* - Q\| \leq \frac{\epsilon}{1-\gamma}$ (use triangle inequality, Luke!)

Take $F = PT_\pi$;
$\epsilon = \|Q_\pi - PQ_\pi\| = \|Q_\pi - PT_\pi Q_\pi\|$ - projection accuracy
$\Rightarrow$ for PT_π's stationary point Q^{PT_π} :

$$\|Q^{PT_\pi} - Q_\pi\| \leq \frac{\epsilon}{1-\gamma}$$

The upper bound is strict (least upper bound), so for weak contractions the stationary point of PT_π can be very far from Q_π

Consider general form of operator:

$$RQ(x,a) = Q(x,a) + E_\mu \left[\sum_{t=0}^{\infty} \gamma^t \left(\prod_{i=1}^{t} c_t \right) (r_t + \gamma E_\pi Q(x_{t+1}, .) - Q(x_t, a_t)) \right]$$

- **IS:** $c_t = \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}$
  Importance Sampling for policy estimation with baseline Q.
  $\forall Q, RQ = Q_\pi$
  Yields Q_π immediately, though has high variance due to importance sampling
- **Q(λ):** $c_t = \lambda$
  Let $\epsilon := max_x \parallel \pi(.|x) - \mu(.|x) \parallel_1$,
  then $\forall \lambda < \frac{1-\gamma}{\epsilon\gamma}$, R is a contraction with fixed point Q_π
  Impractical because it's hard to estimate ϵ, low λ yields Bellman operator

$$RQ(x,a) = Q(x,a) + E_\mu\left[\sum_{t=0}^{\infty}\gamma^t\left(\prod_{i=1}^{t} c_t\right)(r_t + \gamma E_\pi Q(x_{t+1},.) - Q(x_t,a_t))\right]$$

Retrace(λ): $c_t = \lambda min\left(1, \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}\right)$, $\lambda \in [0,1]$

- γ-contraction around Q_π.
  Though we will see a much stronger result in a second
- $\lambda \to 0 \Rightarrow R \to T_\pi$
  Becomes deterministic as $\lambda \to 0$ - bias-variance tradeoff possibility, though usually unexploited ($\lambda = 1$)
- "Safe", unlike Q(λ) - defines contraction mapping for any pair of policies, independently of hyperparameters

Also, recursive form:
$Q^{ret}(s_t,a_t) = r_t + \gamma c_{t+1}(Q^{ret}(s_{t+1},a_{t+1}) - Q(s_{t+1},a_{t+1})) + \gamma V(s_{t+1})$

Theorem

Let $\forall t, 0 \leq c_t \leq \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}$, *then:*

$$\forall \alpha \in [1,\infty], \quad |RQ(x,a) - Q_\pi(x,a)| \leq \eta(x,a) \parallel Q - Q_\pi \parallel_\alpha$$

where $\eta(x,a) := 1 - (1-\gamma) E_\mu \left[\sum_{t \geq 0} \gamma^t \left(\prod_{i=1}^t c_t \right) \right]$

$\forall t, c_t \simeq 1 \Rightarrow \eta \simeq 0 \Rightarrow RQ \simeq Q_\pi$ - R yields Q_π almost immediately if μ and π are close

ACER

A combination of:

- RETRACE
- Stochastic Dueling Networks: simultaneous Q-V off-policy estimation in continuous domain
- "Trust Region" policy updates for lowering gradient dispersion
- Importance Sampling with weight truncation and correction

Truncation with bias correction

$$\nabla J = E_{traj \sim \mu}\left[\left(\prod_{i=0}^{t} \rho_i\right) \nabla \log \pi(a_t|s_t) Q(s_t, a_t) | s_0 = s\right];\ \rho_i = \frac{\pi(a_i|s_i)}{\mu(a_i|s_i)}$$

First, replace full-trajectory importance sampling, involving product of many potentially unbounded weights, with last importance weight ρ_t:

$$\nabla J \simeq g^{marg} = E_{traj \sim \mu}\left[\rho_t \nabla \log \pi(a_t|s_t) Q(s_t, a_t)\right]$$

Let $\overline{\rho} = min(c, \rho)$, then:

$$\begin{aligned} g^{marg} =& E_{traj \sim \mu}\Big[\overline{\rho_t} \nabla \log \pi(a_t|s_t) Q(s_t, a_t) + \\ & + E_{a \sim \pi}\left(\left[\frac{\rho_t(a) - c}{\rho_t(a)}\right]_+ \nabla \log \pi(a|s_t) Q(s_t, a)\right)\Big] \end{aligned}$$

$\overline{\rho_t} \leq c$; $\left[\frac{\rho_t(a)-c}{\rho_t(a)}\right]_+ \leq 1$

Stochastic Dueling networks

Only for estimation in continuous domains

- Deterministic V estimation
- Stochastic Q and A estimation

Two "heads": A, V

$$\tilde{Q}_\omega(s_t, a_t) \sim V_\omega(s_t) + A_\omega(s_t, a_t) - \frac{1}{n}\sum_{i=1}^{n} A_\omega(s_t, u_i),$$

$$u_i \sim \pi(.|s_t)$$

- Estimate $V_\omega \simeq V$ is consistent with Q: $E_\pi E_u \tilde{Q}_\omega(s_t, a_t) = V_\omega(s_t)$
- Provides error signal for updating V_ω: $E_u \tilde{Q}_\omega = Q_\pi \Rightarrow V_\omega = E_a E_u \tilde{Q}_\omega = V_\pi$

Trust Region Updates

ACER gradient with respect to actor's statistics $\phi_\theta(s_t)$:

$$\begin{aligned} g^{acer} =& \overline{\rho_t} \nabla_\phi log(\pi(a_t|s_t)) \left[Q^{ret}(s_t, a_t) - V_\theta(s_t)\right] + \\ &+ E_{a\sim\pi} \left[\frac{\rho_t(a) - c}{\rho_t(a)}\right]_+ \nabla_\phi log(\pi(a|s_t))[Q_\theta(s_t, a) - V_\theta(s_t)] \end{aligned}$$

Trust Region:

$$\begin{aligned} \text{minimize } & \|g^{acer} - z\|_2 \\ \text{s.t. } & k^T z \leq \delta \end{aligned}$$

Where:

- $k = \nabla_{\phi_\theta(s_t)} D_{KL}\left[\pi_{\phi_{\theta_a}}(s_t) \| \pi_{\phi_\theta}(s_t)\right]$
- θ_a - average policy network

$z^* = g^{acer} - max\left(0, \frac{k^T g^{acer} - \delta}{\|k\|_2^2}\right) k$

z^* is then used to calculate gradients with respect to θ in backpropagation

Algorithm 2 ACER for discrete actions

Reset gradients $d\theta \leftarrow 0$ and $d\theta_v \leftarrow 0$.
Initialize parameters $\theta' \leftarrow \theta$ and $\theta'_v \leftarrow \theta_v$.
if not On-Policy **then**
  Sample the trajectory $\{x_0, a_0, r_0, \mu(\cdot|x_0), \cdots, x_k, a_k, r_k, \mu(\cdot|x_k)\}$ from the replay memory.
else
  Get state x_0
end if
for $i \in \{0, \cdots, k\}$ **do**
  Compute $f(\cdot|\phi_{\theta'}(x_i))$, $Q_{\theta'_v}(x_i, \cdot)$ and $f(\cdot|\phi_{\theta_a}(x_i))$.
  if On-Policy **then**
    Perform a_i according to $f(\cdot|\phi_{\theta'}(x_i))$
    Receive reward r_i and new state x_{i+1}
    $\mu(\cdot|x_i) \leftarrow f(\cdot|\phi_{\theta'}(x_i))$
  end if
  $\bar{\rho}_i \leftarrow \min\left\{1, \frac{f(a_i|\phi_{\theta'}(x_i))}{\mu(a_i|x_i)}\right\}$.
end for

$$Q^{ret} \leftarrow \begin{cases} 0 & \text{for terminal } x_k \\ \sum_a Q_{\theta'_v}(x_k, a) f(a|\phi_{\theta'}(x_k)) & \text{otherwise} \end{cases}$$

for $i \in \{k-1, \cdots, 0\}$ **do**
  $Q^{ret} \leftarrow r_i + \gamma Q^{ret}$
  $V_i \leftarrow \sum_a Q_{\theta'_v}(x_i, a) f(a|\phi_{\theta'}(x_i))$
  Computing quantities needed for trust region updating:

$$\begin{aligned} g &\leftarrow \min\{c, \rho_i(a_i)\} \nabla_{\phi_{\theta'}(x_i)} \log f(a_i|\phi_{\theta'}(x_i))(Q^{ret} - V_i) \\ &\quad + \sum_a \left[1 - \frac{c}{\rho_i(a)}\right]_+ f(a|\phi_{\theta'}(x_i)) \nabla_{\phi_{\theta'}(x_i)} \log f(a|\phi_{\theta'}(x_i))(Q_{\theta'_v}(x_i, a_i) - V_i) \\ k &\leftarrow \nabla_{\phi_{\theta'}(x_i)} D_{KL}\left[f(\cdot|\phi_{\theta_a}(x_i)) \| f(\cdot|\phi_{\theta'}(x_i))\right] \end{aligned}$$

  Accumulate gradients wrt θ': $d\theta' \leftarrow d\theta' + \frac{\partial \phi_{\theta'}(x_i)}{\partial \theta'}\left(g - \max\left\{0, \frac{k^T g - \delta}{\|k\|_2^2}\right\} k\right)$
  Accumulate gradients wrt θ'_v: $d\theta_v \leftarrow d\theta_v + \nabla_{\theta'_v}(Q^{ret} - Q_{\theta'_v}(x_i, a))^2$
  Update Retrace target: $Q^{ret} \leftarrow \bar{\rho}_i \left(Q^{ret} - Q_{\theta'_v}(x_i, a_i)\right) + V_i$
end for
Perform asynchronous update of θ using $d\theta$ and of θ_v using $d\theta_v$.
Updating the average policy network: $\theta_a \leftarrow \alpha\theta_a + (1-\alpha)\theta$

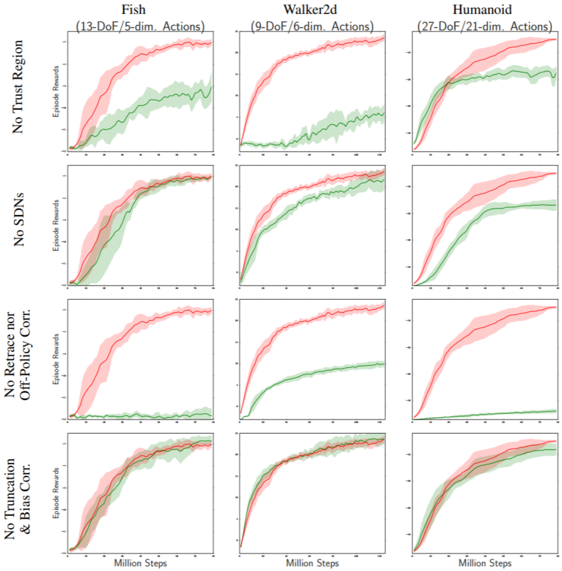
Algorithm 3 ACER for Continuous Actions

Reset gradients $d\theta \leftarrow 0$ and $d\theta_v \leftarrow 0$.
Initialize parameters $\theta' \leftarrow \theta$ and $\theta'_v \leftarrow \theta_v$.
Sample the trajectory $\{x_0, a_0, r_0, \mu(\cdot|x_0), \cdots, x_k, a_k, r_k, \mu(\cdot|x_k)\}$ from the replay memory.
for $i \in \{0, \cdots, k\}$ **do**
  Compute $f(\cdot|\phi_{\theta'}(x_i))$, $V_{\theta'_v}(x_i)$, $\widetilde{Q}_{\theta'_v}(x_i, a_i)$, and $f(\cdot|\phi_{\theta_a}(x_i))$.
  Sample $a'_i \sim f(\cdot|\phi_{\theta'}(x_i))$
  $\rho_i \leftarrow \frac{f(a_i|\phi_{\theta'}(x_i))}{\mu(a_i|x_i)}$ and $\rho'_i \leftarrow \frac{f(a'_i|\phi_{\theta'}(x_i))}{\mu(a'_i|x_i)}$
  $c_i \leftarrow \min\left\{1, (\rho_i)^{\frac{1}{d}}\right\}$.
end for
$Q^{ret} \leftarrow \begin{cases} 0 & \text{for terminal } x_k \\ V_{\theta'_v}(x_k) & \text{otherwise} \end{cases}$
$Q^{opc} \leftarrow Q^{ret}$
for $i \in \{k-1, \cdots, 0\}$ **do**
  $Q^{ret} \leftarrow r_i + \gamma Q^{ret}$
  $Q^{opc} \leftarrow r_i + \gamma Q^{opc}$
  Computing quantities needed for trust region updating:

$$
\begin{aligned}
g &\leftarrow \min\{c, \rho_i\} \nabla_{\phi_{\theta'}(x_i)} \log f(a_i|\phi_{\theta'}(x_i)) \left(Q^{opc}(x_i, a_i) - V_{\theta'_v}(x_i)\right) \\
&+ \left[1 - \frac{c}{\rho'_i}\right]_+ (\widetilde{Q}_{\theta'_v}(x_i, a'_i) - V_{\theta'_v}(x_i)) \nabla_{\phi_{\theta'}(x_i)} \log f(a'_i|\phi_{\theta'}(x_i)) \\
k &\leftarrow \nabla_{\phi_{\theta'}(x_i)} D_{KL}\left[f(\cdot|\phi_{\theta_a}(x_i)) \| f(\cdot|\phi_{\theta'}(x_i))\right]
\end{aligned}
$$

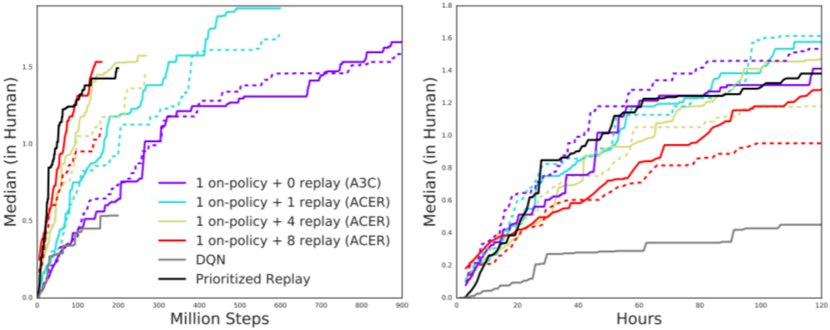
  Accumulate gradients wrt θ: $d\theta \leftarrow d\theta + \frac{\partial \phi_{\theta'}(x_i)}{\partial \theta'} \left(g - \max\left\{0, \frac{k^T g - \delta}{\|k\|_2^2}\right\} k\right)$
  Accumulate gradients wrt θ'_v: $d\theta_v \leftarrow d\theta_v + (Q^{ret} - \widetilde{Q}_{\theta'_v}(x_i, a_i)) \nabla_{\theta'_v} \widetilde{Q}_{\theta'_v}(x_i, a_i)$
  $d\theta_v \leftarrow d\theta_v + \min\{1, \rho_i\} \left(Q^{ret}(x_t, a_i) - \widetilde{Q}_{\theta'_v}(x_t, a_i)\right) \nabla_{\theta'_v} V_{\theta'_v}(x_i)$
  Update Retrace target: $Q^{ret} \leftarrow c_i \left(Q^{ret} - \widetilde{Q}_{\theta'_v}(x_i, a_i)\right) + V_{\theta'_v}(x_i)$
  Update Retrace target: $Q^{opc} \leftarrow \left(Q^{opc} - \widetilde{Q}_{\theta'_v}(x_i, a_i)\right) + V_{\theta'_v}(x_i)$
end for
Perform asynchronous update of θ using $d\theta$ and of θ_v using $d\theta_v$.
Updating the average policy network: $\theta_a \leftarrow \alpha\theta_a + (1 - \alpha)\theta$

Ablation analysis



Discrete actions

Median score across all ATARI games
$1 = \text{human}$
$0 = \text{random}$



Continuous actions