GAE

High-Dimensional Continuous Control Using
Generalized Advantage Estimation
2016

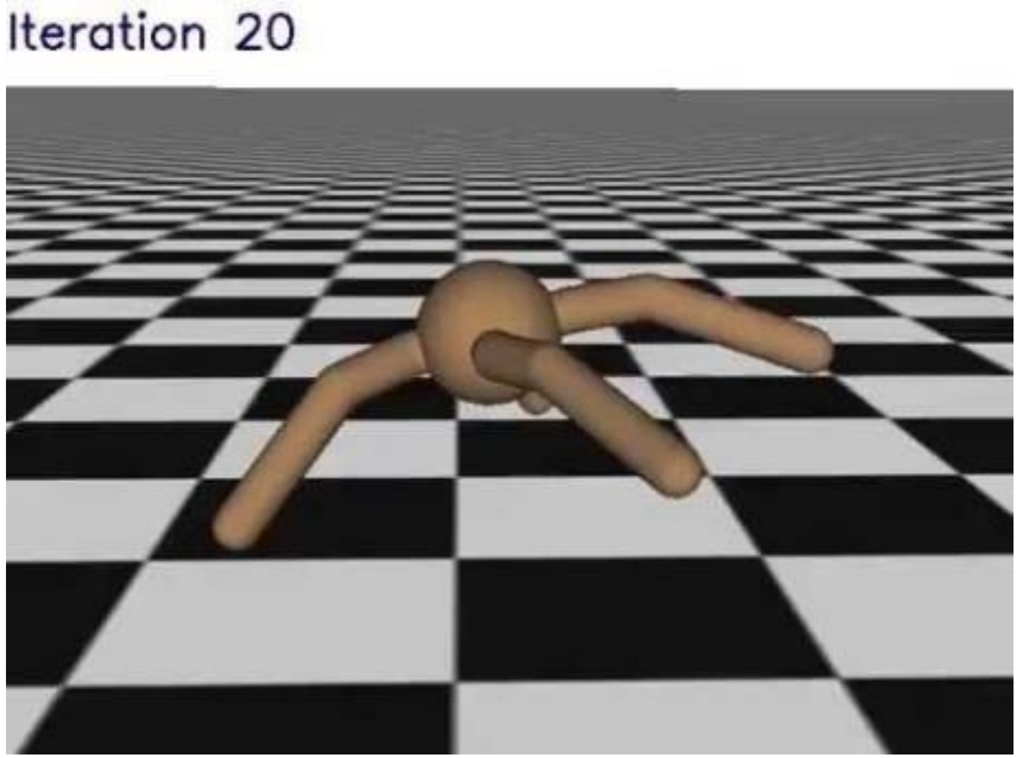
John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan and Pieter Abbeel
University of California, Berkeley

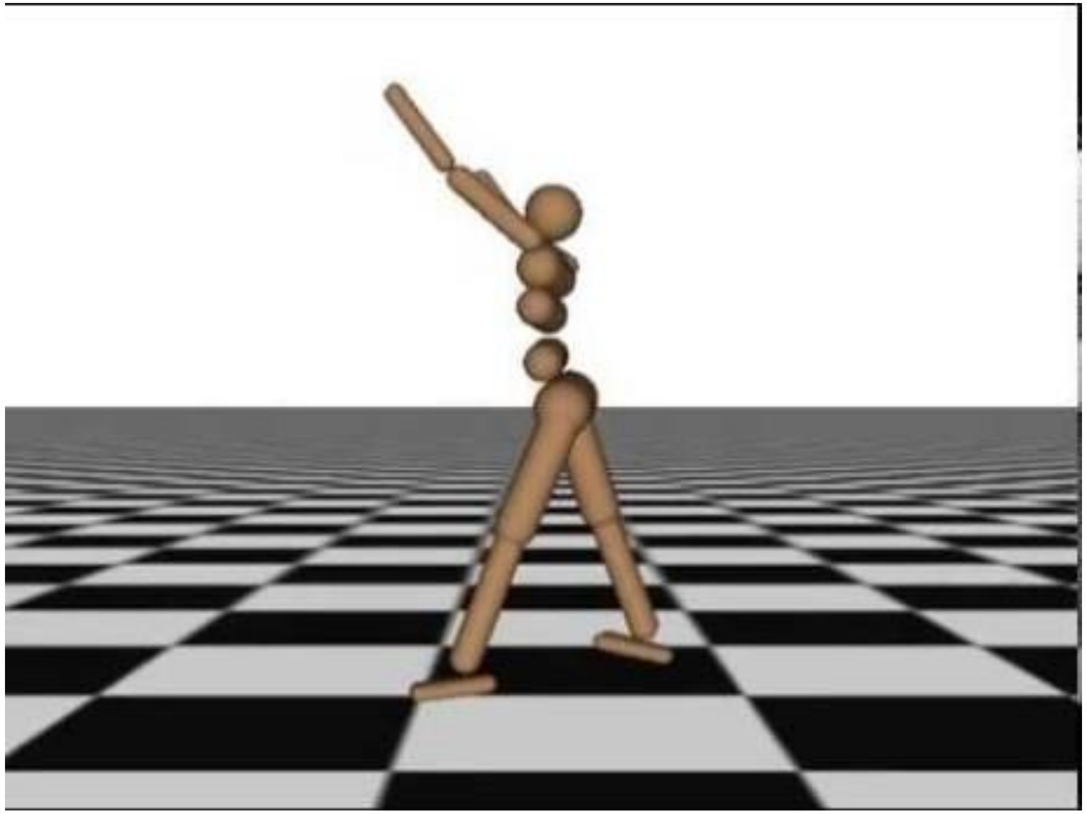
Paper Summary

1. Policy gradient를 사용할 때 나타나는 문제점 지적과 해결
2. 3D locomotion task에 적용한 연구
3. 기존 GAE 연구보다 trust region을 포함한 일반적인 알고리즘 집합을 적용할 수 있게 함.

GAE Summary

1. γ ,λ 를 사용한 estimation 스키마로서 효과적으로 variance를 줄이는 방법
2. Value function을 위해서 trust region optimization method를 사용함
3. 위 두 방법을 결합해서 control task를 풀기 위해 neural network policies를 학습시키는데 능한 알고리즘 얻음


Iteration 20



Preliminaries

s_0 : ρ_0 분포로부터 샘플링 된 초기 state

$a_t \sim \pi(a_t|s_t)$ 에 따라 action 샘플링

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

하나의 trajectory (s0,a0,s1,a1,⋯) 생성

$r_t = r(s_t, a_t\ s_{t+1})$ 매 스텝마다 받는 보상

모든 정책들에 대해서 유한하다고 가정함

Discount factor concept을 사용하지 않는다.

Discounted problem을 undiscounted problem으로 표현할 수 있으며 이것은 시간에 의존적이게 된다.

Preliminaries

Policy gradient는 보상 총합의 기댓값을 최대화하는 쪽으로 반복해서 gradient를 계산한다.

$$g := \nabla_\theta \mathbb{E}\left[\sum_{t=0}^{\infty} r_t\right]$$

우리는 policy gradient 표현을 아래와 같이 할 수 있고

ψ에 여러 형태가 들어갈 수 있음.

$$g = \mathbb{E}\left[\sum_{t=0}^{\infty} \Psi_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\right]$$

Preliminaries

$$g = \mathbb{E}\left[\sum_{t=0}^{\infty} \Psi_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\right]$$

ψ에 들어갈 수 있는 식

1. $\sum_{t=0}^{\infty} r_t$: total reward of the trajectory.
2. $\sum_{t'=t}^{\infty} r_{t'}$: reward following action a_t.
3. $\sum_{t'=t}^{\infty} r_{t'} - b(s_t)$: baselined version of previous formula.
4. $Q^\pi(s_t, a_t)$: state-action value function.
5. $A^\pi(s_t, a_t)$: advantage function.
6. $r_t + V^\pi(s_{t+1}) - V^\pi(s_t)$: TD residual.

Preliminaries

$$g = \mathbb{E}\left[\sum_{t=0}^{\infty} \Psi_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\right]$$

ψ에 들어갈 수 있는 식

4. $Q^\pi(s_t, a_t)$: state-action value function.
5. $A^\pi(s_t, a_t)$: advantage function.
6. $r_t + V^\pi(s_{t+1}) - V^\pi(s_t)$: TD residual.

$$V^{\pi,\gamma}(s_t) := \mathbb{E}_{s_{t+1:\infty}, a_{t:\infty}}\left[\textstyle\sum_{l=0}^{\infty} \gamma^l r_{t+l}\right]$$

$$Q^{\pi}(s_t, a_t) := \mathbb{E}_{s_{t+1:\infty}, a_{t+1:\infty}}\left[\textstyle\sum_{l=0}^{\infty} \gamma^l r_{t+l}\right]$$

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

Preliminaries

Estimating the Advantage Function (1)

- The advantage function can significantly reduce variance of policy gradient
- So the critic should really estimate the advantage function
- For example, by estimating *both* $V^{\pi_\theta}(s)$ *and* $Q^{\pi_\theta}(s, a)$
- Using two function approximators and two parameter vectors,

$$\begin{aligned} V_v(s) &\approx V^{\pi_\theta}(s) \\ Q_w(s, a) &\approx Q^{\pi_\theta}(s, a) \\ A(s, a) &= Q_w(s, a) - V_v(s) \end{aligned}$$

- And updating *both* value functions by e.g. TD learning

Preliminaries

$$V^{\pi,\gamma}(s_t) := \mathbb{E}_{s_{t+1:\infty},a_{t:\infty}}\left[\sum_{l=0}^{\infty}\gamma^l r_{t+l}\right]$$

$$Q^{\pi}(s_t, a_t) := \mathbb{E}_{s_{t+1:\infty},a_{t+1:\infty}}\left[\sum_{l=0}^{\infty}\gamma^l r_{t+l}\right]$$

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

$$g^{\gamma} := \mathbb{E}_{\substack{s_{0:\infty}\\ a_{0:\infty}}}\left[\sum_{t=0}^{\infty} A^{\pi,\gamma}(s_t, a_t)\nabla_\theta \log \pi_\theta(a_t \mid s_t)\right]$$

γ 사용하면 bias가 생기는데 g에 대해서는 unbiased estimate를 얻고 싶음
어떻게 γ 를 사용하면서 unbiased estimate를 얻을 수 있지..?

→ unbiased estimate를 얻을 수 있는 **γ -just estimator**에 대해 소개함

Preliminaries

Definition 1. *The estimator* $\hat{A}_t$ *is* γ*-just if*

$$\mathbb{E}_{\substack{s_{0:\infty}\\a_{0:\infty}}}\left[\hat{A}_t(s_{0:\infty}, a_{0:\infty})\nabla_\theta \log \pi_\theta(a_t \mid s_t)\right] = \mathbb{E}_{\substack{s_{0:\infty}\\a_{0:\infty}}}\left[A^{\pi,\gamma}(s_t, a_t)\nabla_\theta \log \pi_\theta(a_t \mid s_t)\right]$$

It follows immediately that if $\hat{A}_t$ is γ-just for all t, then

$$\mathbb{E}_{\substack{s_{0:\infty}\\a_{0:\infty}}}\left[\sum_{t=0}^{\infty}\hat{A}_t(s_{0:\infty}, a_{0:\infty})\nabla_\theta \log \pi_\theta(a_t \mid s_t)\right] = g^\gamma$$

γ -just인 $\hat{A}_t$ 에 대한 조건 : function Q_t와 b_t로 나뉠 수 있다.

Q_t : γ -discounted Q-function의 unbiased estimator
B_t : action a_t전에 샘플링 된 states와 actions의 arbitrary function(임의함수)

Preliminaries

Proposition 1.

모든 (s_t, a_t)에 대해,

$$\mathbb{E}_{s_{t+1:\infty}, a_{t+1:\infty} | s_t, a_t}[Q_t(s_{t:\infty}, a_{t:\infty})] = Q^{\pi,\gamma}(s_t, a_t)$$

로 인하여 $\hat{A}_t$이

$$\hat{A}_{s_{0:\infty}, a_{0:\infty}} = Q_t(s_{0:\infty}, a_{0:\infty}) - b_t(s_{0:t}, a_{0:t-1})$$

형태라고 가정합시다. (가정을 바탕으로 이루어지는 명제라는 점을 주목합시다.)

그 때, $\hat{A}_t$은 γ-just입니다.

γ -just advantage estimator

- $\sum_{l=0}^{\infty} \gamma^l r_{t+1}$
- $A^{\pi,\gamma}(s_t, a_t)$
- $Q^{\pi,\gamma}(s_t, a_t)$
- $r_t + \gamma V^{\pi,\gamma}(s_{t+1}) - V^{\pi,\gamma}(s_t)$

Proof

Proof of Proposition 1: First we can split the expectation into terms involving Q and b,

$$
\begin{aligned}
&\mathbb{E}_{s_{0:\infty},a_{0:\infty}}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t)(Q_t(s_{0:\infty},a_{0:\infty}) - b_t(s_{0:t},a_{0:t-1}))\right] \\
&\quad= \mathbb{E}_{s_{0:\infty},a_{0:\infty}}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t)(Q_t(s_{0:\infty},a_{0:\infty}))\right] \\
&\quad\;- \mathbb{E}_{s_{0:\infty},a_{0:\infty}}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t)(b_t(s_{0:t},a_{0:t-1}))\right]
\end{aligned}
$$

We'll consider the terms with Q and b in turn.

$$
\begin{aligned}
&\mathbb{E}_{s_{0:\infty},a_{0:\infty}}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) Q_t(s_{0:\infty},a_{0:\infty})\right] \\
&\quad= \mathbb{E}_{s_{0:t},a_{0:t}}\left[\mathbb{E}_{s_{t+1:\infty},a_{t+1:\infty}}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) Q_t(s_{0:\infty},a_{0:\infty})\right]\right] \\
&\quad= \mathbb{E}_{s_{0:t},a_{0:t}}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \boxed{\mathbb{E}_{s_{t+1:\infty},a_{t+1:\infty}}\left[Q_t(s_{0:\infty},a_{0:\infty})\right]}\right] \\
&\quad= \mathbb{E}_{s_{0:t},a_{0:t-1}}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \boxed{A^\pi(s_t,a_t)}\right]
\end{aligned}
$$

Next,

$$
\begin{aligned}
&\mathbb{E}_{s_{0:\infty},a_{0:\infty}}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) b_t(s_{0:t},a_{0:t-1})\right] \\
&\quad= \mathbb{E}_{s_{0:t},a_{0:t-1}}\left[\mathbb{E}_{s_{t+1:\infty},a_{t:\infty}}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) b_t(s_{0:t},a_{0:t-1})\right]\right] \\
&\quad= \mathbb{E}_{s_{0:t},a_{0:t-1}}\left[\boxed{\mathbb{E}_{s_{t+1:\infty},a_{t:\infty}}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t)\right]} b_t(s_{0:t},a_{0:t-1})\right] \\
&\quad= \mathbb{E}_{s_{0:t},a_{0:t-1}}\left[\boxed{0} \cdot b_t(s_{0:t},a_{0:t-1})\right] \\
&\quad= 0.
\end{aligned}
$$

Advantage Function Estimation

$$\mathbb{E}_{\substack{s_{0:\infty} \\ a_{0:\infty}}} \left[\sum_{t=0}^{\infty} \hat{A}_t(s_{0:\infty}, a_{0:\infty}) \nabla_\theta \log \pi_\theta(a_t \mid s_t) \right] = g^\gamma \tag{8}$$

$$\hat{g} = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=0}^{\infty} \hat{A}_t^n \nabla_\theta \log \pi_\theta(a_t^n \mid s_t^n) \tag{9}$$

Advantage Function Estimation

$$\mathbb{E}_{s_{t+1}}\left[\delta_t^{V^{\pi,\gamma}}\right] = \mathbb{E}_{s_{t+1}}\left[r_t + \gamma V^{\pi,\gamma}(s_{t+1}) - V^{\pi,\gamma}(s_t)\right] = \mathbb{E}_{s_{t+1}}\left[Q^{\pi,\gamma}(s_t, a_t) - V^{\pi,\gamma}(s_t)\right] = A^{\pi,\gamma}(s_t, a_t). \quad (10)$$

$$\hat{A}_t^{(1)} := \delta_t^V = -V(s_t) + r_t + \gamma V(s_{t+1}) \quad (11)$$

$$\hat{A}_t^{(2)} := \delta_t^V + \gamma\delta_{t+1}^V = -V(s_t) + r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2}) \quad (12)$$

$$\hat{A}_t^{(3)} := \delta_t^V + \gamma\delta_{t+1}^V + \gamma^2\delta_{t+2}^V = -V(s_t) + r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 V(s_{t+3}) \quad (13)$$

$$\hat{A}_t^{(k)} := \sum_{l=0}^{k-1} \gamma^l \delta_{t+l}^V = -V(s_t) + r_t + \gamma r_{t+1} + \cdots + \gamma^{k-1} r_{t+k-1} + \gamma^k V(s_{t+k}) \quad (14)$$

$$\hat{A}_t^{(\infty)} = \sum_{l=0}^{\infty} \gamma^l \delta_{t+l}^V = -V(s_t) + \sum_{l=0}^{\infty} \gamma^l r_{t+l},$$

TD(λ)

n-Step Return

- Consider the following *n*-step returns for $n = 1, 2, \infty$:

$$
\begin{array}{lll}
n = 1 & (TD) & G_t^{(1)} = R_{t+1} + \gamma V(S_{t+1}) \\
n = 2 & & G_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2}) \\
\vdots & & \vdots \\
n = \infty & (MC) & G_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{T-1} R_T
\end{array}
$$

- Define the *n*-step return

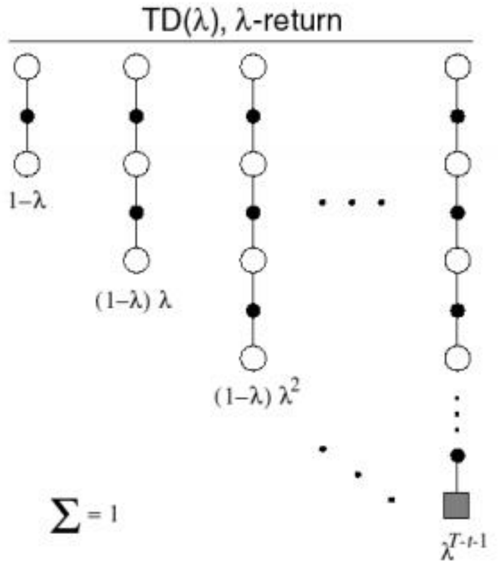
$$G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{n-1} R_{t+n} + \gamma^n V(S_{t+n})$$

- *n*-step temporal-difference learning

$$V(S_t) \leftarrow V(S_t) + \alpha \left(G_t^{(n)} - V(S_t) \right)$$

TD(λ)

λ-return



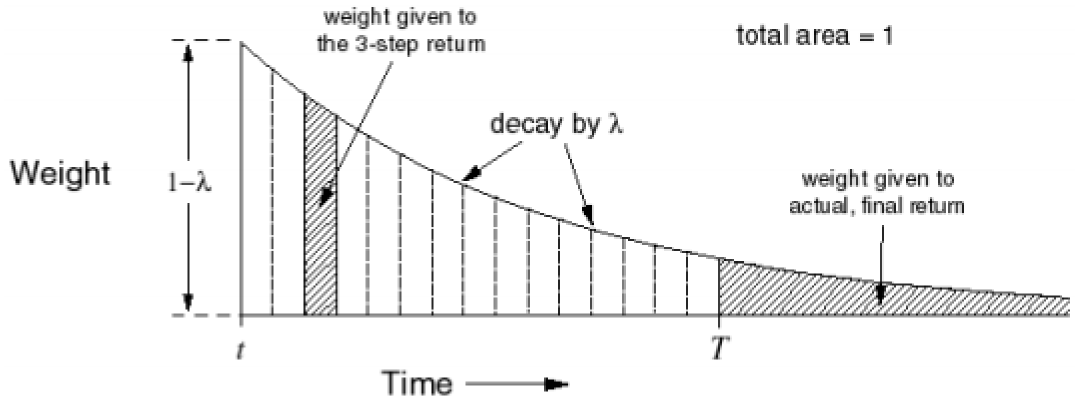
- The λ-*return* G_t^λ combines all n-step returns $G_t^{(n)}$
- Using weight $(1-\lambda)\lambda^{n-1}$

$$G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$$

- Forward-view TD(λ)

$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t^\lambda - V(S_t)\right)$$

TD(λ) Weighting Function



$$G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$$

Advantage Function Estimation

$$\mathbb{E}_{s_{t+1}}\left[\delta_t^{V^{\pi,\gamma}}\right] = \mathbb{E}_{s_{t+1}}\left[r_t + \gamma V^{\pi,\gamma}(s_{t+1}) - V^{\pi,\gamma}(s_t)\right]$$
$$= \mathbb{E}_{s_{t+1}}\left[Q^{\pi,\gamma}(s_t, a_t) - V^{\pi,\gamma}(s_t)\right] = A^{\pi,\gamma}(s_t, a_t). \quad (10)$$

$$\hat{A}_t^{(1)} := \boxed{\delta_t^V} = \boxed{-V(s_t) + r_t + \gamma V(s_{t+1})} \quad (11)$$
$$\hat{A}_t^{(2)} := \delta_t^V + \gamma\delta_{t+1}^V = -V(s_t) + r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2}) \quad (12)$$
$$\hat{A}_t^{(3)} := \delta_t^V + \gamma\delta_{t+1}^V + \gamma^2\delta_{t+2}^V = -V(s_t) + r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 V(s_{t+3}) \quad (13)$$

$$\hat{A}_t^{(k)} := \sum_{l=0}^{k-1} \gamma^l \delta_{t+l}^V = -V(s_t) + r_t + \gamma r_{t+1} + \cdots + \gamma^{k-1} r_{t+k-1} + \gamma^k V(s_{t+k}) \quad (14)$$

$$\hat{A}_t^{(\infty)} = \sum_{l=0}^{\infty} \gamma^l \delta_{t+l}^V = -V(s_t) + \sum_{l=0}^{\infty} \gamma^l r_{t+l},$$

Advantage Function Estimation

$V = V^{\pi,\gamma}$ 우리가 이러한 correct value function을 가지고 있다면,
이것은 γ -just advantage estimator이다.
$A^{\pi,\gamma}(s_t, a_t)$ 의 unbiased estimator은 아래 식으로 표현할 수 있다.

$$\begin{aligned}\mathbb{E}_{s_{t+1}}\left[\delta_t^{V^{\pi,\gamma}}\right] &= \mathbb{E}_{s_{t+1}}\left[r_t + \gamma V^{\pi,\gamma}(s_{t+1}) - V^{\pi,\gamma}(s_t)\right] \\ &= \mathbb{E}_{s_{t+1}}\left[Q^{\pi,\gamma}(s_t, a_t) - V^{\pi,\gamma}(s_t)\right] = A^{\pi,\gamma}(s_t, a_t). \qquad (10)\end{aligned}$$

Advantage Function Estimation

$$\hat{A}_t^{(1)} := \delta_t^V = -V(s_t) + r_t + \gamma V(s_{t+1}) \tag{11}$$

$$\hat{A}_t^{(2)} := \delta_t^V + \gamma \delta_{t+1}^V = -V(s_t) + r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2}) \tag{12}$$

$$\hat{A}_t^{(3)} := \delta_t^V + \gamma \delta_{t+1}^V + \gamma^2 \delta_{t+2}^V = -V(s_t) + r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 V(s_{t+3}) \tag{13}$$

$$\hat{A}_t^{(k)} := \sum_{l=0}^{k-1} \gamma^l \delta_{t+l}^V = -V(s_t) + r_t + \gamma r_{t+1} + \cdots + \gamma^{k-1} r_{t+k-1} + \gamma^k V(s_{t+k}) \tag{14}$$

$$\hat{A}_t^{(\infty)} = \sum_{l=0}^{\infty} \gamma^l \delta_{t+l}^V = -V(s_t) + \sum_{l=0}^{\infty} \gamma^l r_{t+l},$$

Advantage Function Estimation

$$
\begin{aligned}
\hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)} &:= (1-\lambda)\Big(\hat{A}_t^{(1)} + \lambda\hat{A}_t^{(2)} + \lambda^2\hat{A}_t^{(3)} + \dots\Big) \\
&= (1-\lambda)\big(\delta_t^V + \lambda(\delta_t^V + \gamma\delta_{t+1}^V) + \lambda^2(\delta_t^V + \gamma\delta_{t+1}^V + \gamma^2\delta_{t+2}^V) + \dots\big) \\
&= (1-\lambda)(\delta_t^V(1+\lambda+\lambda^2+\dots) + \gamma\delta_{t+1}^V(\lambda+\lambda^2+\lambda^3+\dots) \\
&\qquad + \gamma^2\delta_{t+2}^V(\lambda^2+\lambda^3+\lambda^4+\dots) + \dots) \\
&= (1-\lambda)\left(\delta_t^V\left(\frac{1}{1-\lambda}\right) + \gamma\delta_{t+1}^V\left(\frac{\lambda}{1-\lambda}\right) + \gamma^2\delta_{t+2}^V\left(\frac{\lambda^2}{1-\lambda}\right) + \dots\right) \\
&= \sum_{l=0}^{\infty}(\gamma\lambda)^l\delta_{t+l}^V \qquad (16)
\end{aligned}
$$

Advantage Function Estimation

$$\hat{A}_t^{GAE(\gamma,\lambda)} := (1-\lambda)\left(\hat{A}_t^{(1)} + \lambda \hat{A}_t^{(2)} + \lambda^{(2)} \hat{A}_t^{(3)} + \cdots\right)$$

$$\hat{A}_t^{(\infty)} = \sum_{l=0}^{\infty} \gamma^{l} \delta_{t+l}^{V}$$

$$= (1-\lambda)\left(\delta_t^V + \lambda(\delta_t^V + \gamma\delta_{t+1}^V) + \lambda^2(\delta_t^V + \gamma\delta_{t+1}^V + \gamma^2\delta_{t+2}^V) + \cdots\right)$$

δ 로 묶음.

$$= (1-\lambda)\left(\delta_t^V \underbrace{(1+\lambda+\lambda^2+\lambda^3+\cdots)} + \gamma\delta_{t+1}^V \underbrace{(\lambda+\lambda^2+\lambda^3+\cdots)}_{\frac{\lambda}{1-\lambda}} + \gamma^2\delta_{t+2}^V \underbrace{(\lambda^2+\lambda^3+\cdots)}_{\frac{\lambda^2}{1-\lambda}} + \cdots\right)$$

$$\sum_{k=0}^{\infty} ar^k = \lim_{n\to\infty} \sum_{k=0}^{n-1} ar^k = \lim_{n\to\infty} \frac{a(1-r^n)}{1-r} = \frac{a}{1-r}$$

무한등비급수.

$$\therefore \frac{1}{1-\lambda}$$

$$= \cancel{(1-\lambda)}\left(\delta_t^V \left(\frac{1}{\cancel{1-\lambda}}\right) + \gamma\delta_{t+1}^V \left(\frac{\lambda}{\cancel{1-\lambda}}\right) + \cdots\right)$$

$$= \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V$$

Advantage Function Estimation

λ =0, λ =1 case

$$\begin{aligned}
\text{GAE}(\gamma, 0): \quad & \hat{A}_t := \delta_t && = r_t + \gamma V(s_{t+1}) - V(s_t) \\
\text{GAE}(\gamma, 1): \quad & \hat{A}_t := \sum_{l=0}^{\infty} \gamma^l \delta_{t+l} && = \sum_{l=0}^{\infty} \gamma^l r_{t+l} - V(s_t)
\end{aligned}$$

$$g^\gamma \approx \mathbb{E}\left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t \mid s_t) \hat{A}_t^{\text{GAE}(\gamma,\lambda)}\right] = \mathbb{E}\left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t \mid s_t) \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V\right], \quad (19)$$

where equality holds when $\lambda = 1$.

Advantage Function Estimation

γ 와 λ 를 사용하여 advantage estimator를 표현함.
γ 는 가장 중요하게 $V^{\pi,\gamma}$ 의 scale을 결정한다.
또한 λ 에 의존하지 않는 파라미터

γ <1로 설정 시, value function 정확도와 상관없이 policy gradient estimate에서 bias하게 함.
λ <1은 value function이 부정확할 때에만 bias를 만든다.

λ 의 best value는 γ 의 best value보다 훨씬 낮다.
왜냐하면 γ 보다 λ 가 정확한 value function에 대해 훨씬 덜 bias하기 때문이다.

?

$$g^{\gamma} \approx \mathbb{E}\left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t \mid s_t) \hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)}\right] = \mathbb{E}\left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t \mid s_t) \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V\right], \quad (19)$$

where equality holds when $\lambda = 1$.

Value function estimation - TRPO

Value function 최적화를 위해 Trust Region method 사용
Trust region은 최근 데이터에 대해서 overfitting 되는 것을 막는다.

First, 가장 쉽게 estimation하는 법

$$\underset{\phi}{\text{minimize}} \sum_{n=1}^{N} \|V_\phi(s_n) - \hat{V}_n\|^2$$

$$\hat{V}_t = \sum_{l=0}^{\infty} \gamma^l r_{t+l}$$

Question

$$\underset{\phi}{\text{minimize}} \sum_{n=1}^{N} \|V_\phi(s_n) - \hat{V}_n\|^2$$

where $\hat{V}_t = \sum_{l=0}^{\infty} \gamma^l r_{t+l}$ is the discounted sum of rewards, and n indexes over all timesteps in a batch of trajectories. This is sometimes called the Monte Carlo or TD(1) approach for estimating the value function (Sutton & Barto, 1998).[2]

Value function estimation - TRPO

Trust region을 적용하자.
TRPO는 KL divergence를 사용해서 constraint policy update 구현
하나의 policy와 다른 policy를 비교해 차이를 구하면서 업데이트

1. π 따라 sample batch 모음
2. Sample batch로부터 training batch 구성해서 π' 최적화
    1. $\Delta\eta$ 이라는 extra return 구하고
    2. KL divergence 줄임. (max 구하기 어려워서 mean으로 대체)
3. Set $\pi = \pi'$

Objective Function with Importance Sampling

$$\underset{\theta}{\text{maximize}}\, L_{\theta_{\text{old}}}(\theta) \quad \text{subject to } \overline{D}_{\text{KL}}^{\rho_{\theta_{\text{old}}}}(\theta_{\text{old}}, \theta) \leq \delta.$$

단순히 L을 풀어 쓴 것일 뿐

$$\underset{\theta}{\text{maximize}} \sum_s \rho_{\theta_{\text{old}}}(s) \boxed{\sum_a \pi_\theta(a|s) A_{\theta_{\text{old}}}(s,a)} \text{ subject to } \overline{D}_{\text{KL}}^{\rho_{\theta_{\text{old}}}}(\theta_{\text{old}}, \theta) \leq \delta.$$

$$\sum_a \pi_\theta(a|s_n) A_{\theta_{\text{old}}}(s_n, a) = \mathbb{E}_{a\sim q}\left[\frac{\pi_\theta(a|s_n)}{q(a|s_n)} \boxed{A_{\theta_{\text{old}}}(s_n, a)}\right]$$

A를 최대화하는 것과 Q를 최대화 하는 것은 같다.

$$\underset{\theta}{\text{maximize}}\, \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}, a\sim q}\left[\frac{\pi_\theta(a|s)}{\boxed{q(a|s)}} \boxed{Q_{\theta_{\text{old}}}(s,a)}\right] \tag{14}$$

여기서 q는 $\pi_{\theta_{old}}$를 의미

$$\text{subject to } \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}}\left[D_{\text{KL}}(\pi_{\theta_{\text{old}}}(\cdot|s) \,\|\, \pi_\theta(\cdot|s))\right] \leq \delta.$$

https://github.com/minyoungjun/Pang-yo/blob/master/TRPO.pdf
Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

Value function estimation - TRPO

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}\|V_{\phi_{\text{old}}}(s_n) - \hat{V}_n\|^2$$

$$\underset{\phi}{\text{minimize}} \quad \sum_{n=1}^{N}\|V_\phi(s_n) - \hat{V}_n\|^2$$

$$\boxed{\text{subject to} \quad \frac{1}{N}\sum_{n=1}^{N}\frac{\|V_\phi(s_n) - V_{\phi_{\text{old}}}(s_n)\|^2}{2\sigma^2} \le \epsilon.}$$

이 식은 old value function과 new value function 사이의
KL Divergence를 ε 보다 작게 하는 것과 같다.
Value function은 평균 $V_\phi(s)$ 과 분산 σ^2로 조건부 가우스 분포를 매개변수화한 것.

Value function estimation - TRPO

Trust region 문제는 conjugate gradient algorithm 사용해서 풀 수 있다.
구체적으로 quadratic program을 풀게 된다.

$$\begin{aligned} \underset{\phi}{\text{minimize}} \quad & g^T(\phi - \phi_{\text{old}}) \\ \text{subject to} \quad & \frac{1}{N}\sum_{n=1}^{N}(\phi - \phi_{\text{old}})^T H(\phi - \phi_{\text{old}}) \le \epsilon. \end{aligned}$$

g = objective의 gradient
H = object의 hessian에 대해 gaussian newton method로 근사
Value function을 조건부확률로 보게 되면 H는 fisher information matrix이다.

Experiments

1. GAE 사용시 episodic total reward 최적화 할 때 λ ,γ 의 변화에 대한 경험적인 효과가 무엇인지

2. GAE를 trust region algorithm과 사용하여 policy, value function optimization할 때 어려운 문제 풀기 위해서 large neural network policy들을 최적화 할 수 있는지

Policy Optimization

$$\underset{\theta}{\text{minimize}}\, L_{\theta_{old}}(\theta)$$

$$\text{subject to } \overline{D}_{\mathrm{KL}}^{\theta_{old}}(\pi_{\theta_{old}}, \pi_\theta) \le \epsilon$$

$$\text{where } L_{\theta_{old}}(\theta) = \frac{1}{N}\sum_{n=1}^{N} \frac{\pi_\theta(a_n \mid s_n)}{\pi_{\theta_{old}}(a_n \mid s_n)} \hat{A}_n$$

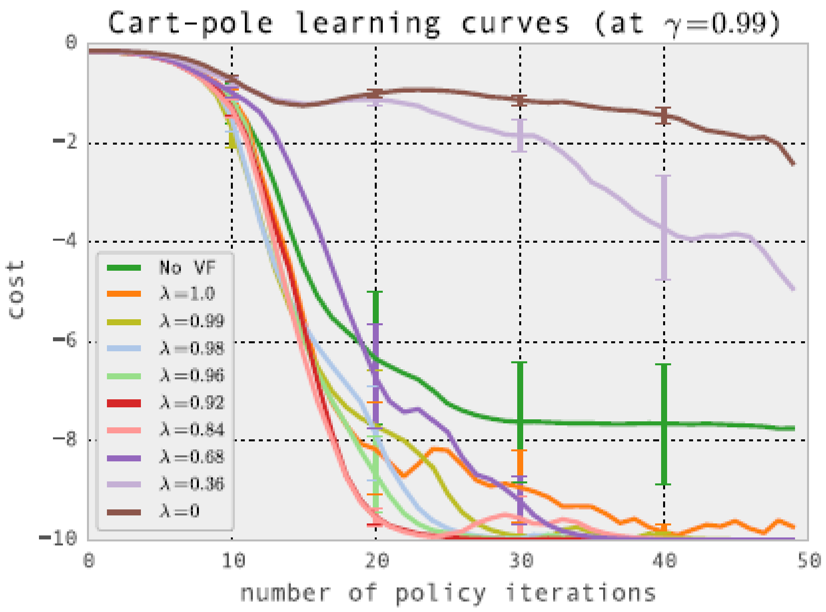
$$\overline{D}_{\mathrm{KL}}^{\theta_{old}}(\pi_{\theta_{old}}, \pi_\theta) = \frac{1}{N}\sum_{n=1}^{N} D_{KL}(\pi_{\theta_{old}}(\cdot \mid s_n) \parallel \pi_\theta(\cdot \mid s_n))$$

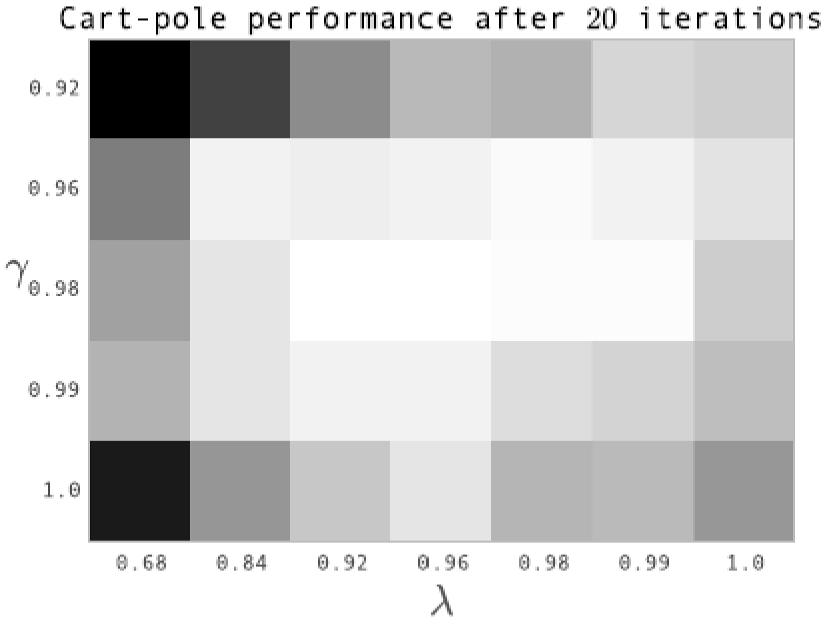
$r_t + \gamma V^{\pi,\gamma}(s_{t+1}) - V^{\pi,\gamma}(s_t)$

Initialize policy parameter θ_0 and value function parameter ϕ_0.
for $i = 0, 1, 2, \ldots$ **do**
  Simulate current policy π_{θ_i} until N timesteps are obtained.
  Compute δ_t^V at all timesteps $t \in \{1, 2, \ldots, N\}$, using $V = V_{\phi_i}$.
  Compute $\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V$ at all timesteps.
  Compute θ_{i+1} with TRPO update, Equation (31).
  Compute ϕ_{i+1} with Equation (30).
end for

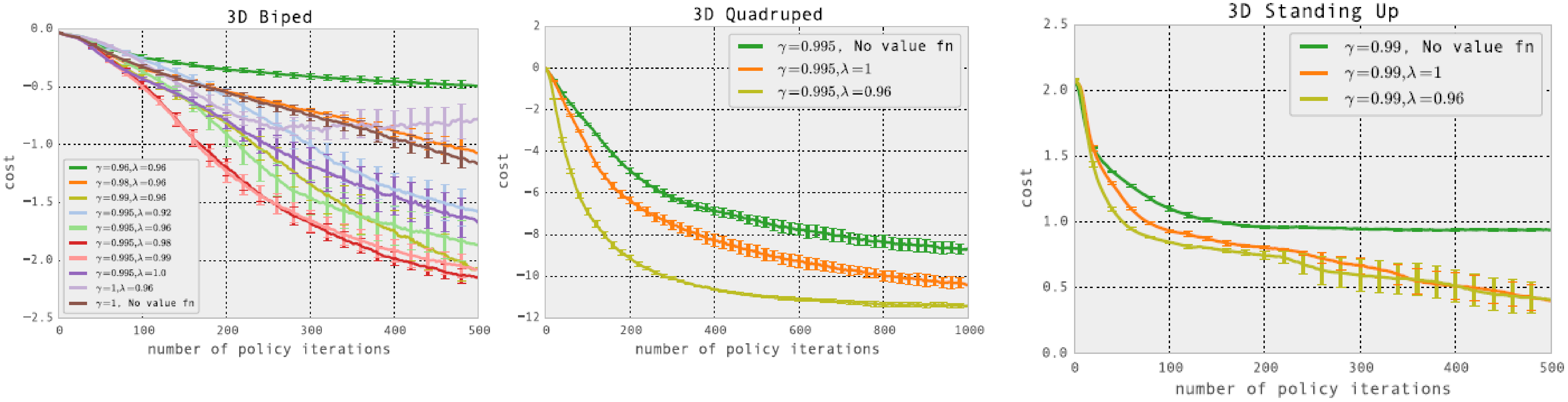
Experiments

$\gamma \in [0.96, 0.99]$ and $\lambda \in [0.92, 0.99]$.





Experiments



Discussion

PG는 bias한 gradient estimation을 제공함으로써 RL을 SGD로 줄이는 방법을 제공함.

그러나 control 문제같이 어려운 문제들은 sample의 복잡성때문에 푸는 것이 제한적이었음

이 논문에서 advantage function의 good estimate를 얻어서 variance를 줄이는 방법을 소개함

γ 와 λ 를 사용하여 GAE를 정의함.

GAE와 함께 trust region 방법으로 value function optimization을 하고 TRPO 사용해서 복잡했던 control 문제를 풀 수 있었음.

Value function estimation error와 policy gradient estimation error 간 관계를 알면 policy gradient estimation의 정확도인 quantity of interest와 잘 일치하는 value function fitting 에 대한 error metric을 잘 고를 수 있다.

Policy와 value function을 위한 function approximation 구조를 공유하는데 사용할 수 있다.

이 구조가 공유되면 더 빠른 학습이 가능해진다.