ETH zürich

FoRL Spring 2024
Project #8

Extending SHAC with Variance Reduction Techniques

Kick Timon[1], Papadaki Kalliopi[1], Virgilio Strozzi[1]
[1]D-INFK, ETH Zurich

1 Introduction and Motivation

Actor-Critic algorithms have achieved notable success in RL, but challenges like sample inefficiency and gradient instability remain. In 2022, the SHAC (Short-Horizon Actor-Critic) algorithm[1] mitigates some issues through efficient parallel differentiable simulation and short optimization windows. We hypothesize further improvements are possible employing **variance reduction** such as td-λ, α-gradient and checkpoint gradients.

2 SHAC (Short Horizon Critic)

Main Idea:

- Split the task horizon H into sub-windows of smaller horizons h.
- Sample N multiple-different trajectories concurrently from a differentiable environment of length h at each step.

Result:

Less problems with local minimal and exploding/vanishing numerical gradient.

More efficient and faster training.

Actor Loss

$$L_\theta := -\frac{1}{Nh}\sum_{i=1}^{N}\left[\left(\sum_{t=t_0}^{t_0+h-1}\gamma^{t-t_0}r_t^i\right)+\gamma^h V_\phi(s_{t_0+h}^i)\right]$$

Critic Loss

$$\mathcal{L}_\phi = \mathop{\mathbb{E}}_{s\in\{\tau_i\}}\left[\|V_\phi(\mathbf{s})-\tilde{V}(\mathbf{s})\|^2\right]$$

td-λ

$$\tilde{V}(\mathbf{s}_t)=(1-\lambda)\left(\sum_{k=1}^{h-t-1}\lambda^{k-1}G_t^k\right)+\lambda^{h-t-1}G_t^{h-t}$$

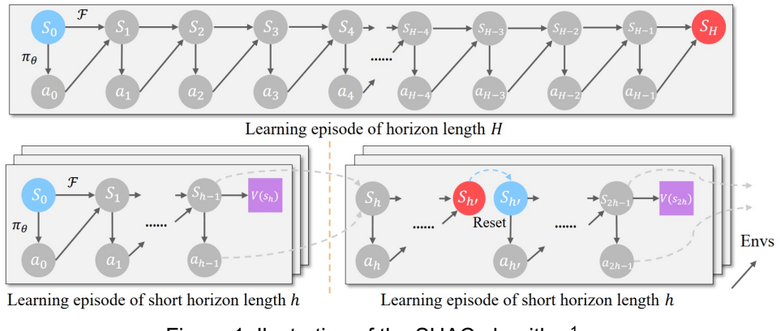


Figure 1: Illustration of the SHAC algorithm[1]

3a Method overview td-λ

Augments the Actor Loss L_θ with a td-λ formulation:

$$-\frac{1}{Nh}\sum_{i=1}^{N}\left[(1-\lambda)\left(\sum_{k=1}^{h-1}\lambda^{k-1}G_k^i\right)+\lambda^{h-1}G_h^i\right]$$

3b Method overview α-order gradient[2]

Policy with added Gaussian noise w_h:

$$V_h(\mathbf{s}_h,\mathbf{w}_{h:H},\boldsymbol{\theta})=\textstyle\sum_{h'=h}^{H}c_h(\mathbf{s}_{h'},\mathbf{u}_{h'}),$$
$$\text{s.t. } \mathbf{x}_{h'+1}=\phi(\mathbf{s}_{h'},\mathbf{u}_{h'}),\ \mathbf{u}_{h'}=\pi(\mathbf{s}_{h'},\boldsymbol{\theta})+\mathbf{w}_{h'},\ h'\geq h$$

Let the α-order gradient be a comb. of 0th and 1th grad:

$$\bar{\nabla}^{[\alpha]}F(\boldsymbol{\theta})=\alpha\bar{\nabla}^{[1]}F(\boldsymbol{\theta})+(1-\alpha)\bar{\nabla}^{[0]}F(\boldsymbol{\theta})$$

$$\hat{\nabla}^{[0]}F_i(\boldsymbol{\theta}):=\frac{1}{\sigma^2}\left[V_1(\mathbf{s}_1,\mathbf{w}_{1:H}^i,\boldsymbol{\theta})-b\right]\left[\sum_{h=1}^{H}\mathrm{D}_{\boldsymbol{\theta}}\pi(\mathbf{s}_h^i,\boldsymbol{\theta})^\intercal\mathbf{w}_h^i\right]$$

0th order grad, , τ_i

$$\hat{\nabla}^{[1]}F_i(\boldsymbol{\theta}):=\nabla_{\boldsymbol{\theta}}V_1(\mathbf{s}_1,\mathbf{w}_{1:H}^i,\boldsymbol{\theta})$$

1th order grad, τ_i

α minimizing variance

$$\min_{\alpha\in[0,1]}\ \alpha^2\hat{\sigma}_1^2+(1-\alpha)^2\hat{\sigma}_0^2$$
$$\text{s.t. }\ \epsilon+\alpha\underbrace{\|\bar{\nabla}^{[1]}F-\bar{\nabla}^{[0]}F\|}_{B}\leq\gamma$$

Express the Actor Loss L_θ as objective F_θ, use α-gradient:

$$-\frac{1}{Nh}\sum_{i=1}^{N}\left[\left(\sum_{t=t_0}^{t_0+h-1}\gamma^{t-t_0}R(s_t^i,a_t^i)\right)+\gamma^h V_\phi(s_{t_0+h}^i)\right]\cong\mathbb{E}_{\mathbf{s}_{t_0}\sim\rho}\mathbb{E}_{\mathbf{w}_h\overset{\text{iid}}{\sim}p}V_{t_0}(\mathbf{s}_{t_0},\mathbf{w}_{t_0:t_0+h},\boldsymbol{\theta})$$

$(Def)\ F_\theta$

3c Method overview checkpoint gradients

Checkpoint gradients reduce variance in gradient estimation by periodically storing full-batch gradient computations and using them to update gradients efficiently.

General Framework:

$$x_{t+1}=x_t-\alpha h_t,\qquad t=1,\ldots,T,$$

where

$$h_t=\begin{cases}\frac{1}{D}\sum_{i=1}^{D}\nabla f(x_t;\xi_t^i), & \text{if } t\equiv 0 \pmod{Q};\\ (1-\eta)\left(h_{t-1}-\frac{1}{S}\sum_{i=1}^{S}\nabla f(x_{t-1};\xi_t^i)\right)+\frac{1}{S}\sum_{i=1}^{S}\nabla f(x_t;\xi_t^i), & \text{otherwise.}\end{cases}$$

4 Experiments

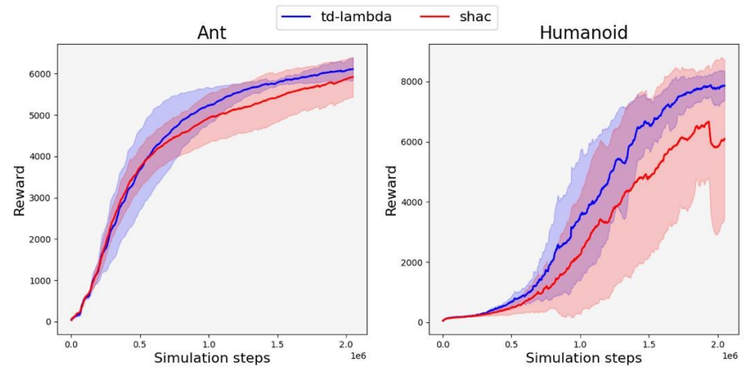
Experimental Environment

- Differentiable simulator: Dflex simulator[4]
- Environments: Ant, Humanoid, SNUHumanoid (RL Games)[3]

Parameters

- Fixed Stochastic Model, #Actors = 64, #Simulations Steps = 1000, Window Length = 32, γ = 0.99.

5 Results



6 Further Extensions

- Same number of Actors between all methods
- Verify scaling on number of environments
- Combine the methods
- 0th order method only

References

1. Jie Xu, Viktor Makoviychuk, Yashraj Narang, Fabio Ramos, Wojciech Matusik, Animesh Garg, and Miles Macklin. Accelerated policy learning with parallel differentiable simulation, 2021.
2. H. J. Terry Suh, Max Simchowitz, Kaiqing Zhang, and Russ Tedrake. Do differentiable simulators give better policy gradients? 2022.
3. GitHub - Denys88/rlgames: RL implementations — github.com. https://github.com/Denys88/rl_games, . [Accessed 28-03-2024].
4. GitHub - NVlabs/DiffRL: [ICLR 2022] Accelerated Policy Learning with Parallel DifferentiableSimulation — github.com. https://github.com/NVlabs/DiffRL, . [Accessed 28-03-2024]
5. Proximal Policy Optimization Algorithms — arxiv.org. https://arxiv.org/abs/1707.06347, . [Accessed 28-03-2024]