State-of-the-art RL methods

Olivier Sigaud

Sorbonne Université
http://people.isir.upmc.fr/sigaud

Outline

- Start from algorithms close to PG: TRPO and ACKTR
- Three aspects distinguish TRPO:
  - Surrogate return objective
  - Natural policy gradient
  - Conjugate gradient approach
- Differences in ACKTR:
  - Approximate second order gradient descent (Hessian)
  - Using Kronecker Factored Approximated Curvature

Surrogate return objective

- The standard policy gradient algorithm for stochastic policies is:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_t[\nabla_{\boldsymbol{\theta}} \text{log} \pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)\hat{A}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}]$$

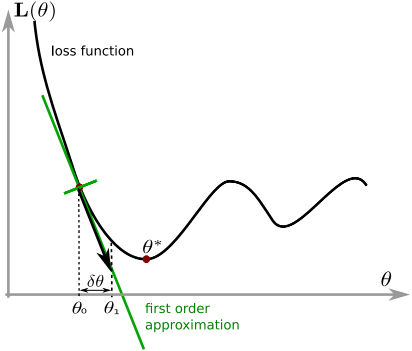
- This gradient is obtained from differentiating $Loss^{PG}(\boldsymbol{\theta}) = \mathbb{E}_t[\text{log}\pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)\hat{A}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}]$
- But we obtain the same gradient from differentiating

$$Loss^{IS}(\boldsymbol{\theta}) = \mathbb{E}_t[\frac{\pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{\boldsymbol{\theta}old}(\mathbf{a}_t|\mathbf{s}_t)}\hat{A}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}]$$

  where $\pi_{\boldsymbol{\theta}old}$ is the policy at the previous iteration
- Because $\nabla_{\boldsymbol{\theta}}\text{log}f(\boldsymbol{\theta})|_{\boldsymbol{\theta}old} = \frac{\nabla_{\boldsymbol{\theta}}f(\boldsymbol{\theta})|_{\boldsymbol{\theta}old}}{f(\boldsymbol{\theta}old)} = \nabla_{\boldsymbol{\theta}}(\frac{f(\boldsymbol{\theta})}{f(\boldsymbol{\theta}old)})|_{\boldsymbol{\theta}old}$

- Another view based on importance sampling
- See John Schulmann's Deep RL bootcamp lecture #5 https://www.youtube.com/watch?v=SQtOI9jsrJ0 (8')

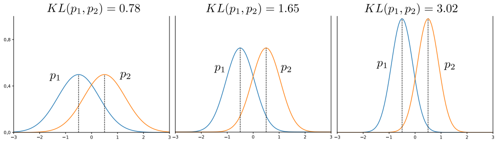
Trust region



- The gradient of a function is only accurate close to the point where it is calculated
- $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ is only accurate close to the current policy $\pi_{\boldsymbol{\theta}}$
- Thus, when updating, $\pi_{\boldsymbol{\theta}}$ must not move too far away from a "trust region" around $\pi_{\boldsymbol{\theta}old}$

Kakade, S. & Langford, J. (2002) Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pages 267–274

Natural Policy Gradient



- One way to constrain two stochastic policies to stay close is constraining their KL divergence
- The KL divergence is smaller when the variance is larger
- Under fixed KL constraint, it is easier to move the mean further away when the variance is large
- Thus the mean policy converges first, then the variance is reduced
- Ensures a large enough amount of exploration noise
- Other properties presented in the Pierrot et al. (2018) paper

Sham M. Kakade. A natural policy gradient. In *Advances in neural information processing systems*, pp. 1531–1538, 2002

Pierrot, T., Perrin, N., & Sigaud, O. (2018) First-order and second-order variants of the gradient descent: a unified framework. *arXiv preprint arXiv:1810.08102*

Trust Region Policy Optimization

- Theory: monotonous improvement towards the optimal policy (Assumptions do not hold in practice)
- To ensure small steps, TRPO uses a natural gradient update instead of standard gradient
- Minimize Kullback-Leibler divergence to previous policy
-

$$\max_{\boldsymbol{\theta}} \mathbb{E}_t[\frac{\pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{\boldsymbol{\theta}old}(\mathbf{a}_t|\mathbf{s}_t)} A_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}old}}(\mathbf{s}_t, \mathbf{a}_t)]$$

$$\text{subject to } \mathbb{E}_t[KL(\pi_{\boldsymbol{\theta}old}(.|\mathbf{s})||\pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t))] \leq \delta$$

- In TRPO, optimization performed using a conjugate gradient method to avoid approximating the Fisher Information matrix

Schulman, J., Levine, S., Moritz, P., Jordan, M. I., & Abbeel, P. (2015) Trust Region Policy Optimization. *CoRR, abs/1502.05477*

Advantage estimation

- To get $\hat{A}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}$, an empirical estimate of $V^{\pi_{\boldsymbol{\theta}}}(s)$ is needed
- TRPO uses a MC estimate approach through regression, but constrains it (as for the policy):

$$\min_{\boldsymbol{\phi}} \sum_{n=0}^{N} ||V^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}(s_n) - V^{\pi_{\boldsymbol{\theta}}}(s_n)||^2$$

$$\text{subject to } \frac{1}{N} \sum_{n=0}^{N} \frac{||V^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}(s_n) - V^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}_{old}}(s_n)||^2}{2\sigma^2} \leq \epsilon$$

- Equivalent to a mean KL divergence constraint between $V^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}$ and $V^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}_{old}}$

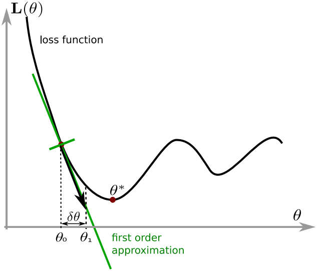
Properties

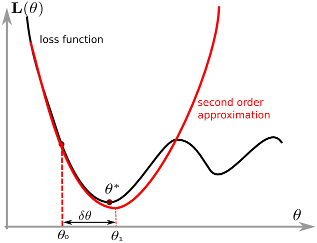
- Moves slowly away from current policy
- Key: use of line search to deal with the gradient step size
- More stable than DDPG, performs well in practice, but less sample efficient
- Conjugate gradient approach not provided in standard tensor gradient librairies, thus not much used
- Greater impact of PPO
- Related work: NAC, REPS

Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71 (7-9):1180–1190, 2008

Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *AAAI*, pp. 1607–1612. Atlanta, 2010

First order versus second order derivative





- In first order methods, need to define a step size
- Second order methods provide a more accurate approximation
- They also provide a true minimum, when the Hessian matrix is symmetric positive-definite (SPD)
- In both cases, the derivative is very local
- The trust region constraint applies too

ACKTR

- K-FAC: Kronecker Factored Approximated Curvature: efficient estimate of the gradient
- Using block diagonal estimations of the Hessian matrix, to do better than first order
- ACKTR: TRPO with K-FAC natural gradient calculation
- But closer to actor-critic updates (see PPO)
- The per-update cost of ACKTR is only 10% to 25% higher than SGD
- Improves sample efficiency
- Not much excitement: less robust gradient approximation?
- Next lesson: PPO

Yuhuai Wu, Elman Mansimov, Shun Liao, Roger Grosse, and Jimmy Ba (2017) Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation. *arXiv preprint arXiv:1708.05144*

Outline

- There are two PPO algorithms
- They are well covered on youtube videos
- So only a quick overview here
- Easy implementation, a lot used
- Key question: is it Actor-Critic?

Proximal Policy Optimization (Algorithm 1)

- The conjugate gradient method of TRPO is not available in tensor libraries
- Same idea as TRPO, but uses a soft constraint on trust region rather than a hard one
- Instead of:

$$\max_{\boldsymbol{\theta}} \mathbb{E}_t\left[\frac{\pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{\boldsymbol{\theta}old}(\mathbf{a}_t|\mathbf{s}_t)} A_{\pi_{\boldsymbol{\theta}old}}(\mathbf{s}_t,\mathbf{a}_t)\right]$$

$$\text{subject to } \mathbb{E}_t[KL(\pi_{\boldsymbol{\theta}old}(.|s)||\pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t))] \leq \delta$$

- Rather use:

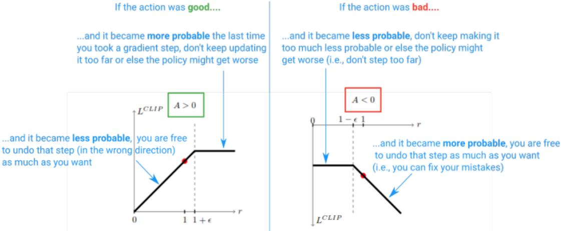
$$\max_{\boldsymbol{\theta}} \mathbb{E}_{s\sim\rho,a\sim\pi}\left[\frac{\pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{\boldsymbol{\theta}old}(\mathbf{a}_t|\mathbf{s}_t)} A_{\pi_{\boldsymbol{\theta}old}}(\mathbf{s}_t,\mathbf{a}_t)\right] - \beta\mathbb{E}_{s\sim\rho}[KL(\pi_{\boldsymbol{\theta}old}(.|\mathbf{s})||\pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t))]$$

- Makes it possible to use SGD instead of conjugate gradient

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal Policy Optimization Algorithms. arXiv preprint arXiv:1707.06347.

Heess, N., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, A., Riedmiller, M., et al. (2017) Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*

Proximal Policy Optimization (Algorithm 2)



If the action was **good....**

...and it became **more probable** the last time you took a gradient step, don't keep updating it too far or else the policy might get worse

...and it became **less probable**, you are free to undo that step (in the wrong direction) as much as you want

If the action was **bad....**

...and it became **less probable**, don't keep making it too much less probable or else the policy might get worse (i.e., don't step too far)

...and it became **more probable**, you are free to undo that step as much as you want (i.e., you can fix your mistakes)

Figure 1: Plots showing one term (i.e., a single timestep) of the surrogate function L^{CLIP} as a function of the probability ratio r, for positive advantages (left) and negative advantages (right). The red circle on each plot shows the starting point for the optimization, i.e., $r = 1$. Note that L^{CLIP} sums many of these terms.

- Image taken from stackoverflow.com
- $\frac{\pi_{\boldsymbol{\theta}}(a|s)}{\pi_{\boldsymbol{\theta}old}(a|s)}$ may get huge if $\pi_{\boldsymbol{\theta}old}$ is very small
- Clipped importance sampling loss (clipping the surrogate objective)

$$r_t(\boldsymbol{\theta}) = \frac{\pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{\boldsymbol{\theta}old}(\mathbf{a}_t|\mathbf{s}_t)}$$

$$L^{CLIP}(\boldsymbol{\theta}) = \mathbb{E}_t[min(r_t(\boldsymbol{\theta})\hat{A}_t, clip(r_t(\boldsymbol{\theta}), 1-\epsilon, 1+\epsilon)\hat{A}_t)]$$

- Back-propagate $L^{CLIP}(\boldsymbol{\theta})$ through a policy network

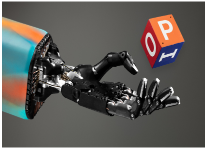
Is PPO actor-critic?

- Improvement over TRPO, thus REINFORCE-like policy update
- But:
  - Algorithm: "PPO, actor-critic style"
  - In the Dota-2 paper: "PPO, a variant of advantage actor-critic, ..."
- What matters is the critic (or baseline) update method
- Uses N-step Generalized Advantage Estimate instead of Monte Carlo
- Thus somewhere between MC and TD (same for ACKTR)
- Other properties:
  - Simpler implementation, better performance than TRPO
  - Does not use a replay buffer → more stable, less sample efficient
  - Still on-policy, $\pi_{\boldsymbol{\theta}}$ and $\pi_{\boldsymbol{\theta}old}$ cannot differ much

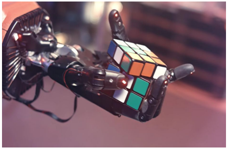
Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019

PPO applications

1536 GPU at peak, 10 months for training, 40.000 years



a pool of 384 worker machines, each with 16 CPU cores



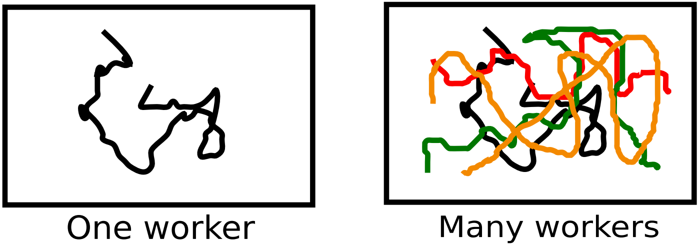
64 V100 GPU + 900 workers, with 32 CPU cores, several months, 13.000 years

- Massive parallel versions of PPO, with dedicated architectures
- Very few teams can afford such engineering and computing effort

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019

OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020

Massive parallel updates

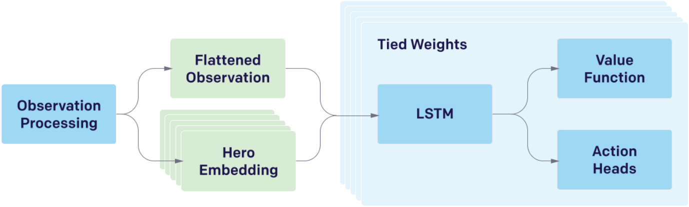


- Several workers in parallel: more i.i.d and faster exploration
- The acceleration is better than linear in the number of workers
- No need for a replay buffer (as in A3C), but loss of sample efficiency

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al. (2018) Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. *arXiv preprint arXiv:1802.01561*

Adamski, I., Adamski, R., Grel, T., Jedrych, A., Kaczmarek, K., & Michalewski, H. (2018) Distributed deep reinforcement learning: Learn how to play atari games in 21 minutes. *arXiv preprint arXiv:1801.02852*

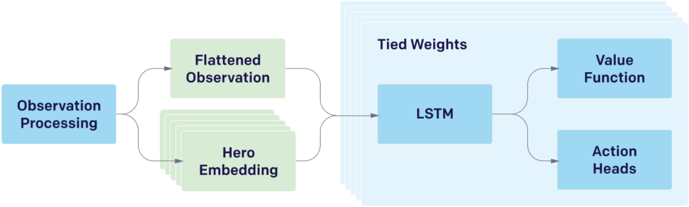
OpenIA five



- The LSTM deals with non-Markov data
- The vision layers are problem specific

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019

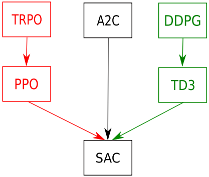
OpenIA five



- The LSTM deals with non-Markov data
- The vision layers are problem specific

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019

Soft Actor Critic: The best of two worlds



- TRPO and PPO: π_θ stochastic, on-policy, low sample efficiency, stable
- DDPG and TD3: π_θ deterministic, replay buffer, better sample efficiency, unstable
- SAC: "Soft" means "entropy regularized", π_θ stochastic, replay buffer
- Adds entropy regularization to favor exploration (follow-up of several papers)
- Attempt to be stable and sample efficient
- Three successive versions

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A. Abbeel, P. et al. (2018) Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018) Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*

Haarnoja, T. Tang, H., Abbeel, P. and Levine, S. (2017) Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165*

Soft Actor-Critic

SAC learns a **stochastic** policy π^* maximizing both rewards and entropy:

$$\pi^* = \arg\max_{\pi_{\boldsymbol{\theta}}} \sum_t \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\rho_{\pi_{\boldsymbol{\theta}}}} \left[r(\mathbf{s}_t,\mathbf{a}_t) + \alpha\mathcal{H}(\pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t))\right]$$

- The entropy is defined as: $\mathcal{H}(\pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t)) = \mathbb{E}_{\mathbf{a}_t\sim\pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t)}\left[-\log\pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)\right]$
- SAC changes the traditional MDP objective
- Thus, it converges toward different solutions
- Consequently, it introduces a new value function, the soft value function
- As usual, we consider a policy $\pi_{\boldsymbol{\theta}}$ and a soft action-value function $\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}$

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. (2016) Asynchronous methods for deep reinforcement learning. *arXiv preprint arXiv:1602.01783*

Soft policy evaluation

- Usually, we define $\hat{V}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}(\mathbf{s}_t) = \mathbb{E}_{\mathbf{a}_t \sim \pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t)}\left[\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t)\right]$
- In soft updates, we rather use:

$$\begin{aligned}\hat{V}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}(\mathbf{s}_t) &= \mathbb{E}_{\mathbf{a}_t \sim \pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t)}\left[\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t)\right] + \alpha \mathcal{H}(\pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t)) \\ &= \mathbb{E}_{\mathbf{a}_t \sim \pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t)}\left[\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t)\right] + \alpha \mathbb{E}_{\mathbf{a}_t \sim \pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t)}\left[-\log \pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)\right] \\ &= \mathbb{E}_{\mathbf{a}_t \sim \pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t)}\left[\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t) - \alpha \log \pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)\right]\end{aligned}$$

Critic updates

- We define a standard Bellman operator:

$$\begin{aligned}\mathcal{T}^{\pi}\hat{Q}_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}(\mathbf{s}_t,\mathbf{a}_t) &= r(\mathbf{s}_t,\mathbf{a}_t)+\gamma V_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}(\mathbf{s}_{t+1})\\ &= r(\mathbf{s}_t,\mathbf{a}_t)+\gamma\mathbb{E}_{\mathbf{a}_t\sim\pi_{\boldsymbol{\theta}}(.|\mathbf{s}_{t+1})}\left[\hat{Q}_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}(\mathbf{s}_{t+1},\mathbf{a}_t)-\alpha\log\pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_{t+1})\right]\end{aligned}$$

Critic parameters can be learned by minimizing the loss associated to $J_Q(vth)$:

$$loss_Q(\boldsymbol{\theta})=\mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_{t+1})\sim\mathcal{D}}\left[\left(r(\mathbf{s}_t,\mathbf{a}_t)+\gamma\hat{V}_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}(\mathbf{s}_{t+1})-\hat{Q}_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}(\mathbf{s}_t,\mathbf{a}_t)\right)^2\right]$$

where $V_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}(\mathbf{s}_{t+1})=\mathbb{E}_{\mathbf{a}\sim\pi_{\boldsymbol{\theta}}(.|\mathbf{s}_{t+1})}\left[\hat{Q}_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}(\mathbf{s}_{t+1},\mathbf{a})-\alpha\log\pi_{\boldsymbol{\theta}}(\mathbf{a}|\mathbf{s}_{t+1})\right]$

- Similar to DDPG update, but with entropy

Actor updates

- Update policy such as to become greedy w.r.t to the soft Q-value
- Choice: update the policy towards the exponential of the soft Q-value

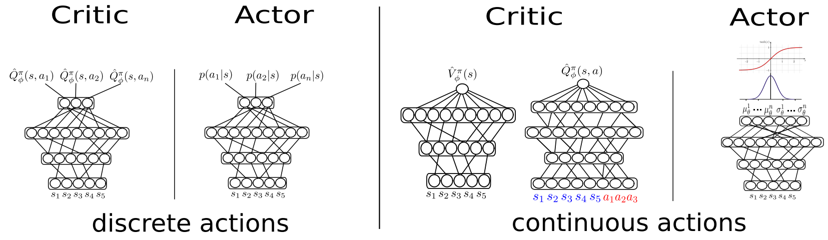
$$J_\pi(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}}[KL(\pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t))||\frac{\exp(\frac{1}{\alpha}\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}(\mathbf{s}_t, .))}{Z_{\boldsymbol{\theta}}(\mathbf{s}_t)}].$$

- $Z_{\boldsymbol{\theta}}(\mathbf{s}_t)$ is just a normalizing term to have a distribution
- SAC does not minimize directly this expression but a surrogate one that has the same gradient w.r.t $\boldsymbol{\theta}$

The policy parameters can be learned by minimizing:

$$J_\pi(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}}\left[\mathbb{E}_{\mathbf{a}_t \sim \pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t)}\left[\alpha \log \pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) - \hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t)\right]\right]$$

Continuous vs discrete actions setting


Critic
Actor
$\hat{Q}^{\pi}_{\phi}(s,a_1)$ $\hat{Q}^{\pi}_{\phi}(s,a_2)$ $\hat{Q}^{\pi}_{\phi}(s,a_n)$
$p(a_1|s)$ $p(a_2|s)$ $p(a_n|s)$
$s_1\ s_2\ s_3\ s_4\ s_5$
discrete actions
Critic
Actor
$\hat{V}^{\pi}_{\phi}(s)$
$\hat{Q}^{\pi}_{\phi}(s,a)$
$\mu^1_\theta \cdots \mu^n_\theta\ \sigma^1_\theta \cdots \sigma^n_\theta$
$s_1\ s_2\ s_3\ s_4\ s_5\ a_1 a_2 a_3$
continuous actions

- SAC works in both the discrete action and the continuous action setting
- Discrete action setting:
  - The critic takes a state and returns a Q-value per action
  - The actor takes a state and returns probabilities over actions
- Continuous action setting:
  - The critic takes a state and an action vector and returns a scalar Q-value
  - Need to choose a distribution function for the actor
  - SAC uses a squashed Gaussian: $\mathbf{a} = \tanh(n)$ where $n \sim \mathcal{N}(\mu_{\boldsymbol{\phi}}, \sigma_{\boldsymbol{\phi}})$

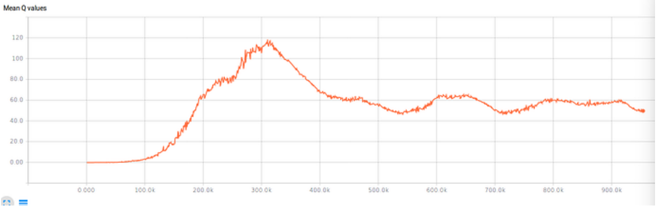
Continuous vs discrete actions setting

- In $J_\pi(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}}\left[\mathbb{E}_{\mathbf{a}_t \sim \pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t)}\left[\alpha \log \pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) - \hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t)\right]\right]$
- SAC updates require to estimate an expectation over actions sampled from the actor,
- That is $\mathbb{E}_{\mathbf{a}_t \sim \pi_{\boldsymbol{\theta}}(.|s)}\left[F(\mathbf{s}_t, \mathbf{a}_t)\right]$ where F is a scalar function.

- In the discrete action setting, $\pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t)$ is a vector of probabilities
    - $\mathbb{E}_{\mathbf{a}_t \sim \pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t)}\left[F(\mathbf{s}_t, \mathbf{a}_t)\right] = \pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t)^T F(\mathbf{s}_t, .)$

- In the continuous action setting:
    - The actor returns $\mu_{\boldsymbol{\theta}}$ and $\sigma_{\boldsymbol{\theta}}$
    - Re-parameterization trick: $\mathbf{a}_t = \tanh(\mu_{\boldsymbol{\theta}} + \epsilon.\sigma_{\boldsymbol{\theta}})$ where $\epsilon \sim \mathcal{N}(0, 1)$
    - Thus, $\mathbb{E}_{\mathbf{a}_t \sim \pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t)}\left[F(\mathbf{s}_t, \mathbf{a}_t)\right] = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1)}\left[F(\mathbf{s}_t, \tanh(\mu_{\boldsymbol{\theta}} + \epsilon\sigma_{\boldsymbol{\theta}}))\right]$
    - This trick reduces the variance of the expectation estimate
    - And allows to backprop through the expectation w.r.t $\boldsymbol{\theta}$

Twin Delayed Deep Deterministic PG



- All descendants of Q-learning suffer from over-estimation bias
- Clipping the critic from the knowledge of R_{max} helps
- TD3: Introduce two critics $\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}_1}$ and $\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}_2}$
- Compute the TD-target as the minimum to reduce the over-estimation bias
- Less problem knowledge than critic clipping
- Next lesson: Soft Actor Critic

Fujimoto, S., van Hoof, H., & Meger, D. (2018) Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*

Critic update improvements (from TD3)

- As in TD3, SAC uses two critics $\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}_1}$ and $\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}_2}$
- The TD-target becomes:

$$y_t = r + \gamma \mathbb{E}_{\mathbf{a}_{t+1}\sim\pi_{\boldsymbol{\theta}}(.|\mathbf{s}_{t+1})}\left[\min_{i=1,2}\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\bar{\boldsymbol{\phi}}_i}(\mathbf{s}_{t+1},\mathbf{a}_{t+1}) - \alpha\log\pi_{\boldsymbol{\theta}}(\mathbf{a}_{t+1}|\mathbf{s}_{t+1})\right]$$

And the losses:

$$\begin{cases} J(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_{t+1})\sim\mathcal{D}}\left[\left(\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}_1}(\mathbf{s}_t,\mathbf{a}_t) - y_t\right)^2 + \left(\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}_2}(\mathbf{s}_t,\mathbf{a}_t) - y_t\right)^2\right] \\ J(\boldsymbol{\theta}) = \mathbb{E}_{s\sim\mathcal{D}}\left[\mathbb{E}_{\mathbf{a}_t\sim\pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t)}\left[\alpha\log\pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) - \min_{i=1,2}\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\bar{\boldsymbol{\phi}}_i}(\mathbf{s}_t,\mathbf{a}_t)\right]\right] \end{cases}$$

Fujimoto, S., van Hoof, H., & Meger, D. (2018) Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*

Automatic Entropy Adjustment

- The temperature α needs to be tuned for each task
- Finding a good α is non trivial
- Instead of tuning α, tune a lower bound $\mathcal{H}_0$ for the policy entropy
- And change the optimization problem into a constrained one

$$\begin{cases} \pi^* = \underset{\pi}{\operatorname{argmax}} \sum_t \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\rho_{\pi_{\boldsymbol{\theta}}}} \left[r(\mathbf{s}_t,\mathbf{a}_t)\right] \\ \text{s.t. } \forall t \ \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\rho_{\pi_{\boldsymbol{\theta}}}} \left[-\log \pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)\right] \geq \mathcal{H}_0, \end{cases}$$

- Use heuristic to compute $\mathcal{H}_0$ from the action space size

α can be learned to satisfy this constraint by minimizing:

$$J(\alpha) = \mathbb{E}_{\mathbf{s}_t\sim\mathcal{D}} \left[\mathbb{E}_{\mathbf{a}_t\sim\pi_{\boldsymbol{\theta}}(.|\mathbf{s}_t)} \left[-\alpha \log \pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) - \alpha\mathcal{H}_0\right]\right]$$

Practical algorithm

- Initialize neural networks $\pi_{\boldsymbol{\theta}}$ and $\hat{Q}_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}$ weights
- Play k steps in the environment by sampling actions with $\pi_{\boldsymbol{\theta}}$
- Store the collected transitions in a replay buffer
- Sample k batches of transitions in the replay buffer
- Update the temperature α, the actor and the critic using SGD
- Repeat this cycle until convergence

Truncated Quantile Critics

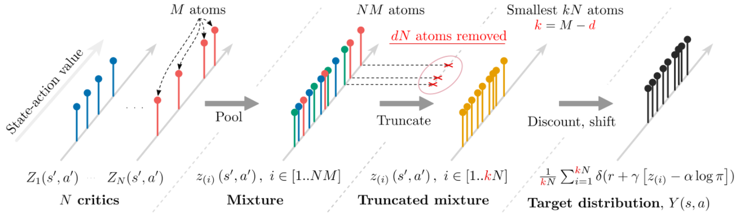
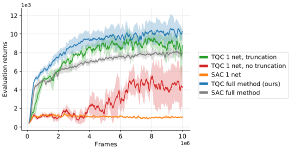

State-action value
M atoms
NM atoms
dN atoms removed
Smallest kN atoms
$k = M - d$
Pool
Truncate
Discount, shift
$Z_1(s', a')$ ⋯ $Z_N(s', a')$
N critics
$z_{(i)}(s', a'),\ i \in [1..NM]$
Mixture
$z_{(i)}(s', a'),\ i \in [1..kN]$
Truncated mixture
$\frac{1}{kN}\sum_{i=1}^{kN}\delta(r + \gamma\,[z_{(i)} - \alpha \log \pi])$
Target distribution, $Y(s, a)$

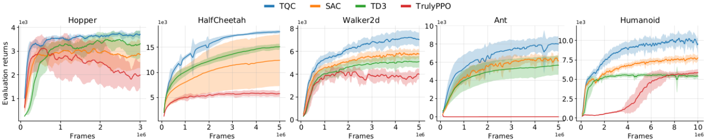
Figure 2. Step-by-step construction of the temporal difference target distribution $Y(s, a)$. First, we compute approximations of the return distribution conditioned on s' and a' by evaluating N separate target critics. Second, we make a mixture out of the N distributions from the previous step. Third, we truncate the right tail of this mixture to obtain atoms $z_{(i)}(s', a')$ from equation 11. Fourthly, we add entropy term, discount and add reward as in soft Bellman equation.

- To fight overestimation bias, TD3 and SAC take the min over two critics
- Using a distribution of estimates is more stable than a single estimate
- TQC uses stochastic critics and truncates the higher quantiles

Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *International Conference on Machine Learning*, pp. 5556–5566. PMLR, 2020

Performance



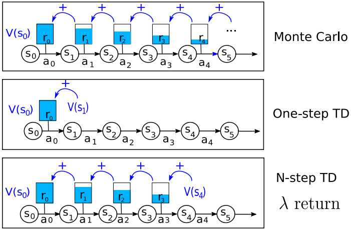


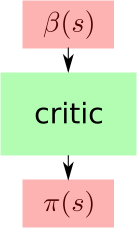
- ▶ From 5 to a single critic
- ▶ Outperforms SAC, easier to use

Key Policy Gradient Steps

- 1. Splitting the trajectory into steps: Markov Hypothesis required
- Key difference to Direct Policy Search methods
- Makes it possible to optimize trajectories using a gradient over policy params
- 2. Introducing the Q function
- Makes it possible to perform policy updates from a single step
- Opens the way to the replay buffer, critic networks, partly off-policy methods
- 3. Using baselines
- Makes it possible to reduce variance
- When learning critics from bootstrap, becomes actor-critic

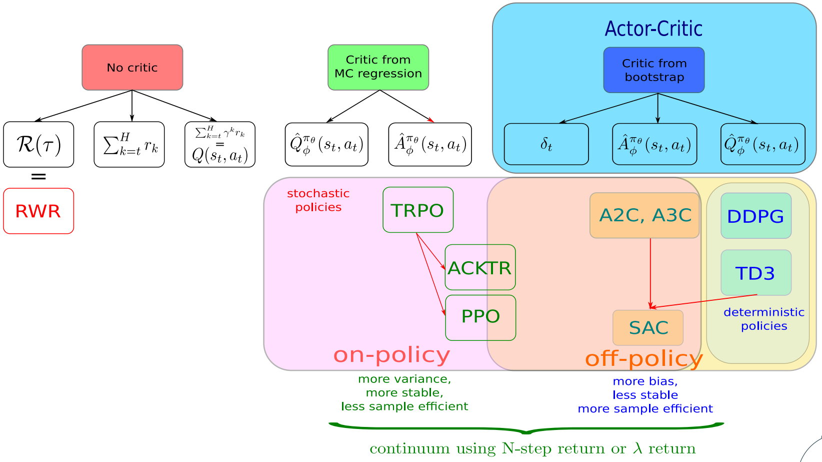
Bias-variance, Being Off-policy


V(s_0)
r_0
r_1
r_2
r_3
r_4
...
s_0 a_0 s_1 a_1 s_2 a_2 s_3 a_3 s_4 a_4 s_5
Monte Carlo
V(s_0)
r_0
V(s_1)
s_0 a_0 s_1 a_1 s_2 a_2 s_3 a_3 s_4 a_4 s_5
One-step TD
V(s_0)
r_0
r_1
r_2
r_3
V(s_4)
s_0 a_0 s_1 a_1 s_2 a_2 s_3 a_3 s_4 a_4 s_5
N-step TD
λ return



- Continuum between Monte Carlo methods and bootstrap methods
- Playing on the continuum helps finding the right bias-variance trade-off
- Being off-policy requires bootstrap
- No deep RL algorithm is truly off-policy, it's a matter of degree

Final view



No critic

$\mathcal{R}(\tau)$

$\sum_{k=t}^{H} r_k$

$\sum_{k=t}^{H} \gamma^k r_k = Q(s_t, a_t)$

=

RWR

Critic from MC regression

$\hat{Q}_{\phi}^{\pi_\theta}(s_t, a_t)$

$\hat{A}_{\phi}^{\pi_\theta}(s_t, a_t)$

Actor-Critic

Critic from bootstrap

δ_t

$\hat{A}_{\phi}^{\pi_\theta}(s_t, a_t)$

$\hat{Q}_{\phi}^{\pi_\theta}(s_t, a_t)$

stochastic policies

TRPO

ACKTR

PPO

A2C, A3C

SAC

DDPG

TD3

deterministic policies

on-policy

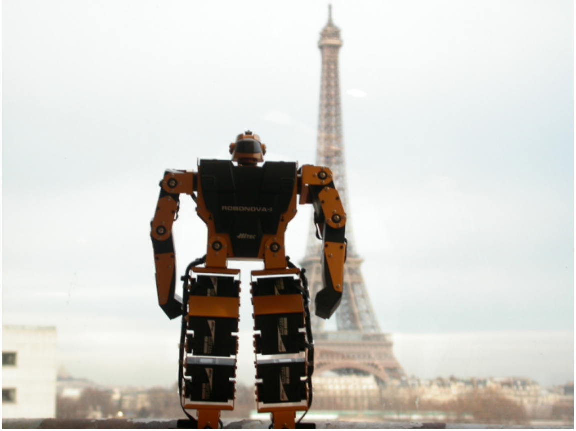
off-policy

more variance,
more stable,
less sample efficient

more bias,
less stable
more sample efficient

continuum using N-step return or λ return

Any question?



Send mail to: `Olivier.Sigaud@upmc.fr`

Adamski, I., Adamski, R., Grel, T., Jedrych, A., Kaczmarek, K., and Michalewski, H. (2018).
Distributed deep reinforcement learning: Learn how to play atari games in 21 minutes.
arXiv preprint arXiv:1801.02852.

Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., et al. (2019).
Solving rubik's cube with a robot hand.
arXiv preprint arXiv:1910.07113.

Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al. (2019).
Dota 2 with large scale deep reinforcement learning.
arXiv preprint arXiv:1912.06680.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., Legg, S., and Kavukcuoglu, K. (2018).
IMPALA: scalable distributed deep-rl with importance weighted actor-learner architectures.
In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1406–1415. PMLR.

Fujimoto, S., van Hoof, H., and Meger, D. (2018).
Addressing function approximation error in actor-critic methods.
In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1582–1591. PMLR.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018a).
Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor.
In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. (2018b).

Soft actor-critic algorithms and applications.
arXiv preprint arXiv:1812.05905.

Heess, N., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, A., Riedmiller, M., et al. (2017).
Emergence of locomotion behaviours in rich environments.
arXiv preprint arXiv:1707.02286.

Kakade, S. and Langford, J. (2002).
Approximately optimal approximate reinforcement learning.
In *ICML*, volume 2, pages 267–274.

Kakade, S. M. (2001).
A natural policy gradient.
In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 1531–1538. MIT Press.

Kuznetsov, A., Shvechikov, P., Grishin, A., and Vetrov, D. P. (2020).
Controlling overestimation bias with truncated mixture of continuous distributional quantile critics.
In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5556–5566. PMLR.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. (2016).
Asynchronous methods for deep reinforcement learning.
In Balcan, M. and Weinberger, K. Q., editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1928–1937. JMLR.org.

OpenAI, Andrychowicz, M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., et al. (2020).
Learning dexterous in-hand manipulation.
The International Journal of Robotics Research, 39(1):3–20.

Peters, J., Mülling, K., and Altun, Y. (2010).

Relative entropy policy search.
In *AAAI*, pages 1607–1612. Atlanta.

Peters, J. and Schaal, S. (2008).
Natural actor-critic.
Neurocomputing, 71(7-9):1180–1190.

Pierrot, T., Perrin, N., and Sigaud, O. (2018).
First-order and second-order variants of the gradient descent: a unified framework.
arXiv preprint arXiv:1810.08102.

Schulman, J., Levine, S., Abbeel, P., Jordan, M. I., and Moritz, P. (2015).
Trust region policy optimization.
In Bach, F. R. and Blei, D. M., editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1889–1897. JMLR.org.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017).
Proximal policy optimization algorithms.
arXiv preprint arXiv:1707.06347.

Wu, Y., Mansimov, E., Grosse, R. B., Liao, S., and Ba, J. (2017).
Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation.
In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5279–5288.