Deep Policy Search

3. TRPO, ACKTR and PPO

Olivier Sigaud

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

Outline

- This class builds on my previous Policy Gradient class
- Three algorithms are presented: TRPO, ACKTR, PPO
- Two aspects distinguish TRPO:
  - Surrogate return objective
  - Natural policy gradient
- A small difference with ACKTR:
  - Using Kronecker Factored Approximated Curvature to estimate the natural gradient
- About PPO:
  - There are two PPO algorithms
  - They are well covered on youtube videos
  - So only a quick overview here
  - Easy implementation, a lot used

Surrogate return objective

- The standard policy gradient algorithm for stochastic policies is:

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

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

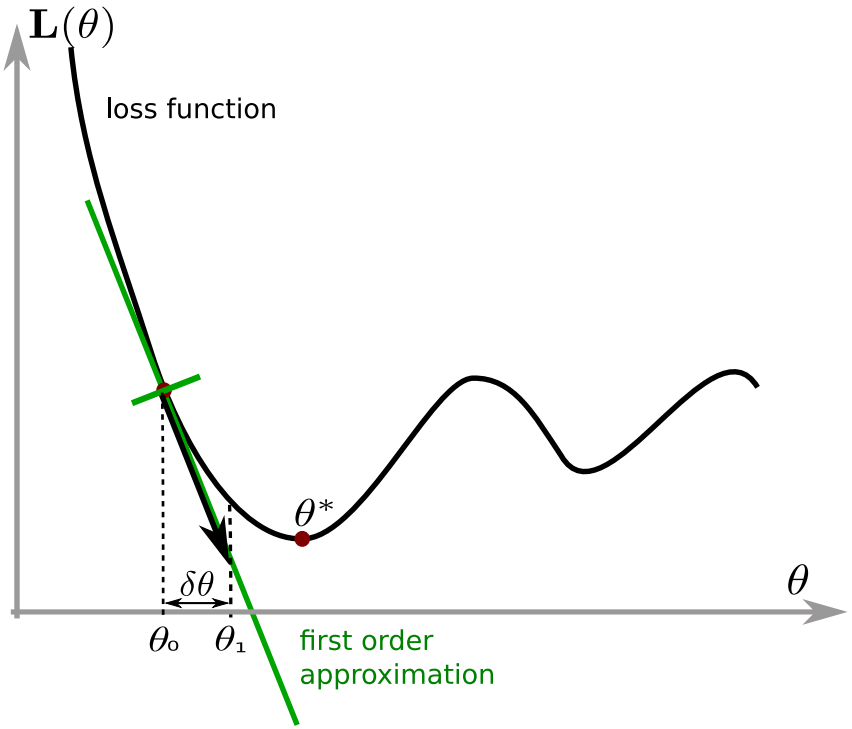
$$Loss^{IS}(\theta) = \mathbb{E}_t[\frac{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{\theta_{old}}(\mathbf{a}_t|\mathbf{s}_t)}\hat{A}_\phi]$$

  where $\pi_{\theta_{old}}$ is the policy at the previous iteration

- Because $\nabla_\theta \log f(\theta)|_{\theta_{old}} = \frac{\nabla_\theta f(\theta)|_{\theta_{old}}}{f(\theta_{old})} = \nabla_\theta(\frac{f(\theta)}{f(\theta_{old})})|_{\theta_{old}}$

- 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 that function
- The gradient of the surrogate objective is only accurate close to the current policy π_θ
- Thus, when updated, the new policy must not move too far away from a "trust region" around the current policy

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

Trust Region Policy Optimization

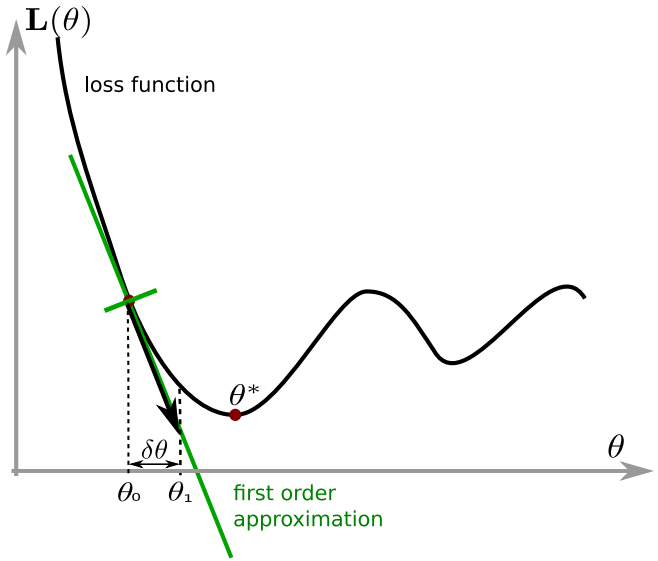
- Theory: monotonous improvement w.r.t. the cost function (Assumptions do not hold in practice)
- To ensure small steps, TRPO uses a natural gradient update instead of standard gradient
- Minimize Kullback-Leibler divergence to previous policy
-

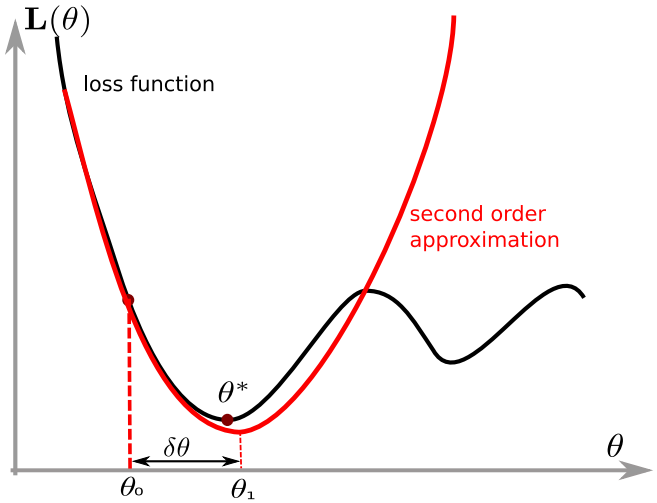
$$\max_{\theta} \mathbb{E}_t\left[\frac{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{\theta_{old}}(\mathbf{a}_t|\mathbf{s}_t)} A_{\pi_{\theta_{old}}}(\mathbf{s}_t, \mathbf{a}_t)\right]$$

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

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

First order versus second order derivative

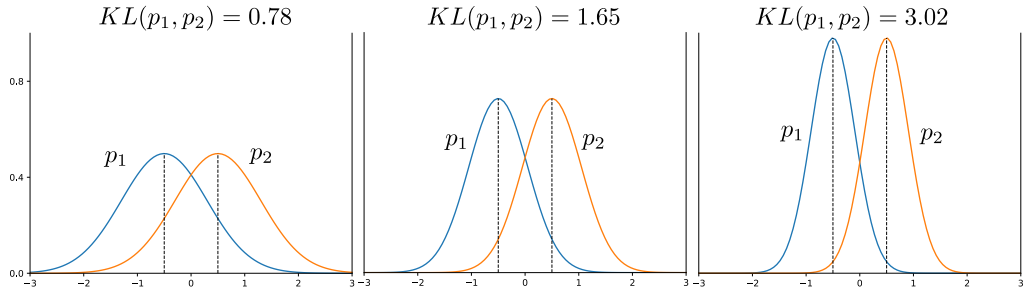




- In first order methods, need to define a step size
- Second order methods provide a more accurate approximation
- They also provide a true minimum, when the Hessian matrix is symmetric positive-definite matrix (SPD)
- In both cases, the derivative is very local
- The gradient should not be applied too far away from the current point

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*

Natural Policy Gradient



- One way to constrain to stochastic policies to stay close to each other is constraining the KL divergence
- Under KL constraint, it is easier to move 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 listed in the Pierrot et al. (2018) paper
- In TRPO, optimization performed using a conjugate gradient method to avoid approximating the Fisher Information matrix

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

Advantage estimation

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

$$\min_\phi \sum_{n=0}^{N} ||V_\phi(s_n) - V(s_n)||^2$$

$$\text{subject to } \frac{1}{N} \sum_{n=0}^{N} \frac{||V_\phi(s_n) - V_{\phi_{old}}(s_n)||^2}{2\sigma^2} \leq \epsilon$$

- Equivalent to a mean KL divergence constraint between V_ϕ and $V_{\phi_{old}}$

Properties

- Does not move far away from current policy, thus on-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

ACKTR

- K-FAC: Kronecker Factored Approximated Curvature: efficient estimate of natural 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
- The per-update cost of ACKTR is only 10% to 25% higher than SGD
- Improves sample efficiency
- Not much excitement: less robust gradient approximation?

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*

Proximal Policy Optimization (Algorithm 1)

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

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

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

- Rather use:

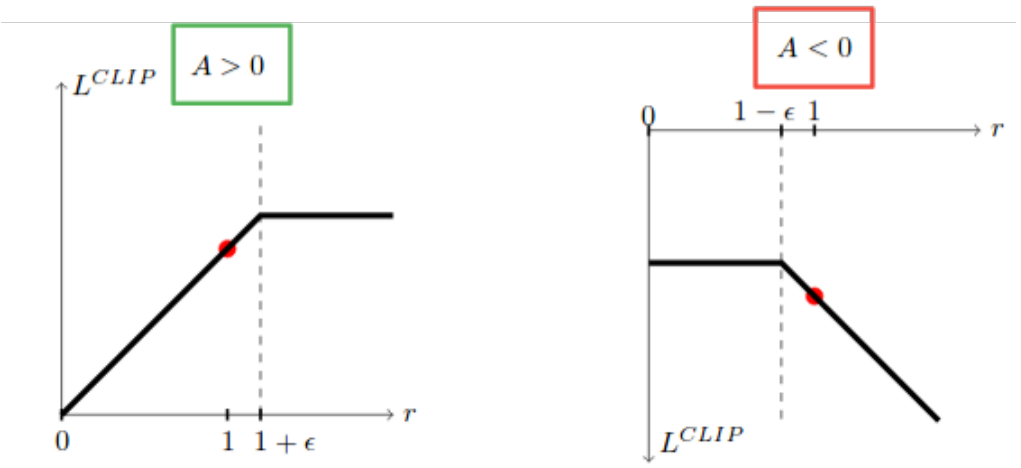
$$\max_{\theta} \mathbb{E}_{s\sim\rho, a\sim\pi}\left[\frac{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{\theta_{old}}(\mathbf{a}_t|\mathbf{s}_t)} A_{\pi_{\theta_{old}}}(\mathbf{s}_t, \mathbf{a}_t)\right] - \beta \mathbb{E}_{s\sim\rho}[KL(\pi_{\theta_{old}}(.|\mathbf{s})||\pi_\theta(\mathbf{a}_t|\mathbf{s}_t))]$$

- 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)



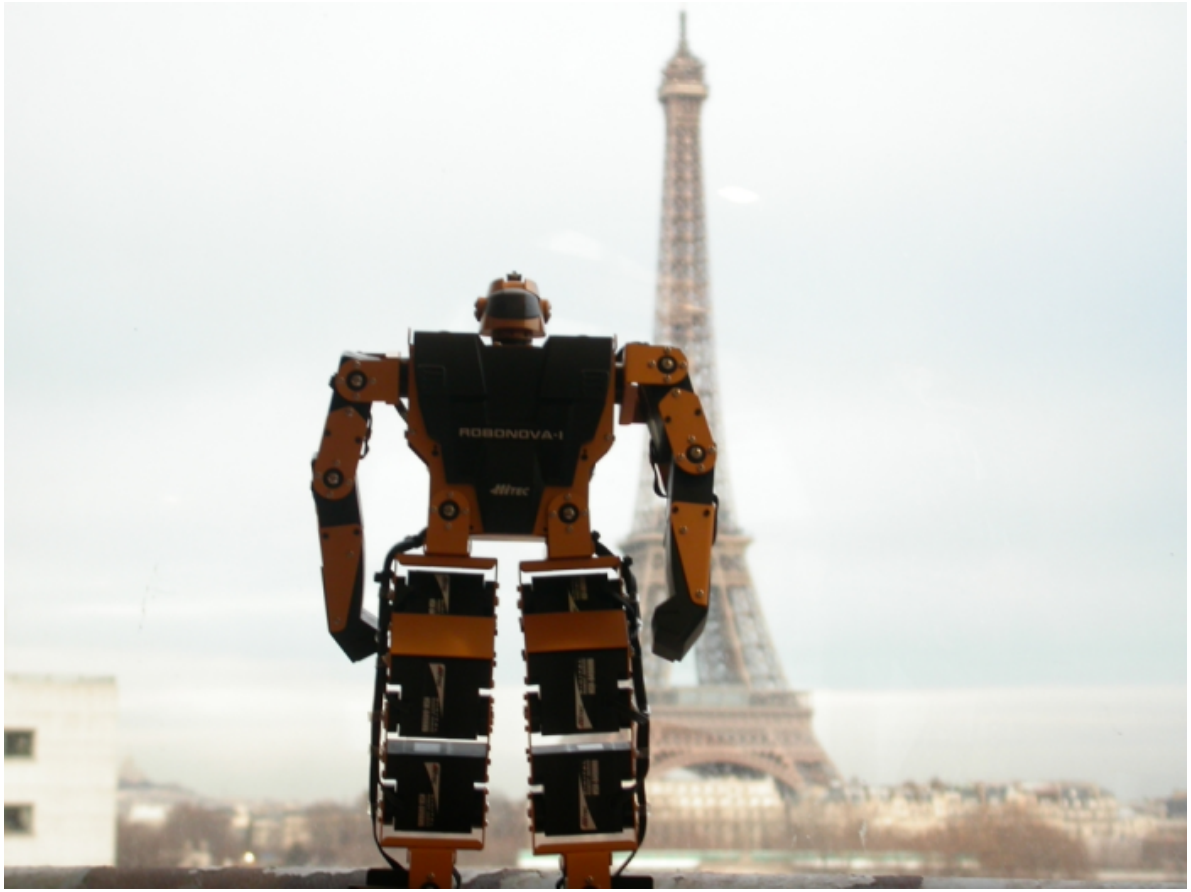
- $\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}$ may get huge if $\pi_{\theta_{old}}$ is very small
- Clipped importance sampling loss (clipping the surrogate objective)
- $$r_t(\theta) = \frac{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{\theta_{old}}(\mathbf{a}_t|\mathbf{s}_t)}$$
- $$L^{CLIP}(\theta) = \mathbb{E}_t[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]$$
- Back-propagate $L^{CLIP}(\theta)$ through a policy network

PPO properties

- Uses N-step return instead of Monte Carlo: more step-based than TRPO
- Simpler implementation, better performance
- But still on-policy because π_θ and $\pi_{\theta_{old}}$ cannot differ much
- Massive parallel versions exist (a lot used by OpenAI)

Frans, K. & Hafner, D. (2016) Parallel trust region policy optimization with multiple actors.

Any question?



Send mail to: Olivier.Sigaud@upmc.fr


Kevin Frans and Danijar Hafner.
Parallel trust region policy optimization with multiple actors.
2016.


Nicolas Heess, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, Ali Eslami, Martin Riedmiller, et al.
Emergence of locomotion behaviours in rich environments.
arXiv preprint arXiv:1707.02286, 2017.


Sham Kakade and John Langford.
Approximately optimal approximate reinforcement learning.
In *ICML*, volume 2, pp. 267–274, 2002.


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


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.


Thomas Pierrot, Nicolas Perrin, and Olivier Sigaud.
First-order and second-order variants of the gradient descent: a unified framework.
arXiv preprint arXiv:1810.08102, 2018.


John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel.
Trust region policy optimization.
CoRR, abs/1502.05477, 2015.


John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov.

Proximal policy optimization algorithms.
arXiv preprint arXiv:1707.06347, 2017.

Yuhuai Wu, Elman Mansimov, Shun Liao, Roger Grosse, and Jimmy Ba.

Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation.
arXiv preprint arXiv:1708.05144, 2017.