From Policy Gradient to Actor-Critic methods

Proximal Policy Optimization (PPO)

Olivier Sigaud

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

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:

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

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

- Rather use:

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

- 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)

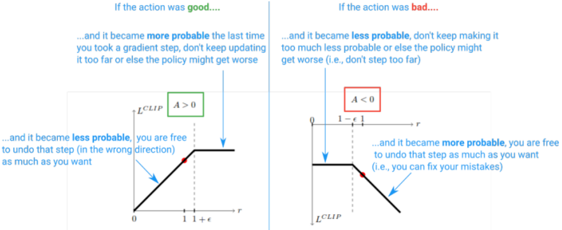


Figure 1: Plots showing one term (i.e., a single timestep) of the surrogate function L^{CLIP} as a function of the probability ratio r, for positive advantages (left) and negative advantages (right). The red circle on each plot shows the starting point for the optimization, i.e., $r = 1$. Note that L^{CLIP}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

PPO and A2C

- PPO is very similar to A2C but
- One needs to store the previous policy $\pi_{\boldsymbol{\theta}old}(a|s)$
- The actor loss uses the clipped ratio $\frac{\pi_{\boldsymbol{\theta}}(a|s)}{\pi_{\boldsymbol{\theta}old}(a|s)}$ instead of the log probability $log(\pi_{\boldsymbol{\theta}}(a|s))$
- Several additional context-dependent tricks have been added, see: https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/

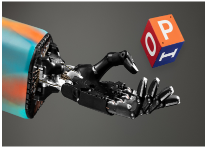
Is PPO actor-critic?

- 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 $\rightarrow$ more stable, less sample efficient
  - Still on-policy, $\pi_{\boldsymbol{\theta}}$ and $\pi_{\boldsymbol{\theta}old}$ cannot differ much

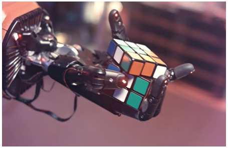
Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019

PPO applications

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



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



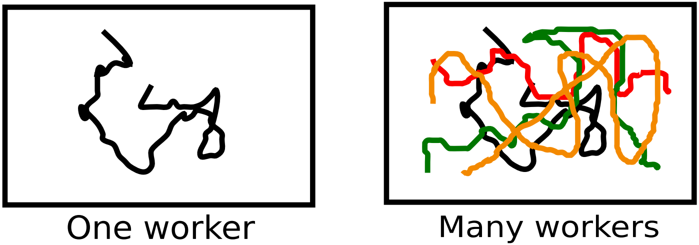
64 V100 GPU + 900 workers, with 32 CPU cores, several months, 13.000 years

- 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

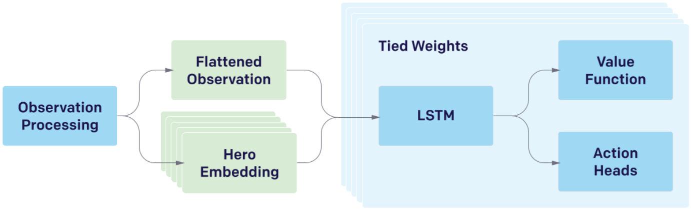


- Several workers in parallel: more i.i.d and faster exploration
- The acceleration is better than linear in the number of workers
- No need for a replay buffer (as in A3C), but loss of sample efficiency

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*

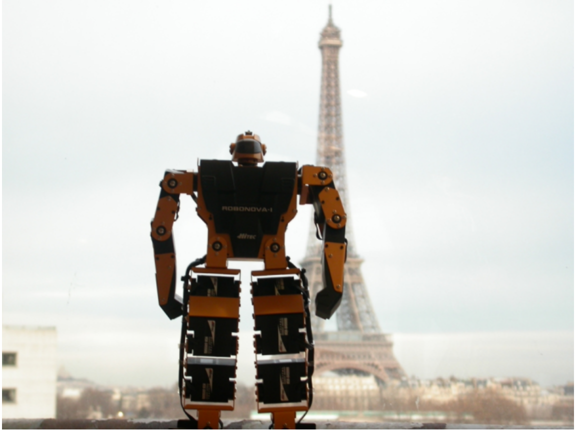
OpenIA five



- 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

Any question?



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

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.

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., & Kavukcuoglu, K. (2018).
IMPALA: scalable distributed deep-rl with importance weighted actor-learner architectures.
In J. G. Dy & A. Krause (Eds.), *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* (pp. 1406–1415).: PMLR.

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.

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.

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