# Project presentation
# Policy gradient in Deep Reinforcement Learning

Ayoub Ghriss, Van Huy Vo
Instructor: Alessandro Lazaric

Ecole Polytechnique - ENS de Paris-Saclay

June 7, 2018

# Goal

- Understand some of the state-of-the-art algorithms in RL such as REPS, TPRO, A3C and their common ideas.
- Neu et al. (2017) A Unified View of Entropy-Regularized Markov Decision Processes
- Get practical experiences by implementing TPRO on Atari.

# Content

# Content

# Regularized Policy gradient

## Policy gradient

We consider MDP on state space $\mathcal{X}$ and action space $\mathcal{A}$ and reward $r_t : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}$.

- Maximize $\rho(\pi) = \lim_T \mathbb{E}\left[\frac{1}{T} \sum_{i=1}^{T} r_t(X_t, A_t)\right]$
- Optimize the policy directly
- Considers parametric policies $(\pi_\theta)$ on the state-action space
- Optimize the parameters $\theta$ (compared to the policy iteration )

# Regularize Policy gradient

## Policy gradient

Why policy gradient is not enough ? : The step

- Large step : forgets the past
- Small step : slow convergence

# Content

# Relative Entropy Policy Search

Restraining the divergence (entropy)

$$\begin{aligned} \theta_{k+1} &= \underset{\theta}{\operatorname{argmax}}\ \rho(\mu_\theta) \\ \text{subject to}\ \ & DL_{KL}(\mu_\theta \| q) \leq \delta. \end{aligned} \tag{1}$$

# Asynchronous Advantage Actor-Critic

Approximates the gradient

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta \Big( \rho(\mu_\theta) - \eta \sum_{\mathcal{X}} \nu_{\pi_k} \sum_{\mathcal{A}} \pi_\theta(a|x) \log \pi_\theta(a|x) \Big). \quad (2)$$

# Trust Region Policy Optimization

Approximates the average discounted rewards and constrains the entropy

$$\begin{aligned} \theta_{k+1} &= \operatorname*{argmax}_{\theta} \sum_{x} \nu_{\pi_{\theta_k}}(x) \sum_{a} \pi_{\theta}(a|x) A^{\pi_{\theta_k}}(x, a) \\ \text{subject to } & \mathbb{E}_{x \sim \nu_{\pi_{\theta_k}}} [D_{KL}(\pi_{\theta_k}(\cdot|x) \| \pi_{\theta}(\cdot|x))] \leq \delta. \end{aligned} \tag{3}$$

# Content

# MDP and average-reward

- Average reward:

$$\rho(\pi) = \lim_{T} \mathbb{E}\left[\frac{1}{T}\sum_{i=1}^{T} r_t(X_t, A_t)\right] \tag{4}$$

- Unique stationary state distribution $\nu_\pi(x)$, state-action distribution $\mu_\pi(x, a) = \nu_\pi(x)\pi(a|x)$
- Problem

$$\mu^* = \operatorname*{argmax}_{\mu\in\Delta} \rho(\mu) = \operatorname*{argmax}_{\mu\in\Delta} \sum_{x,a} \mu(x, a) r(x, a) \tag{5}$$

# Optimization algorithms

- Mirror descent
  - An iterative algorithm, a version of generalized projected gradient descent.
  - Each step

$$x_{k+1} = \underset{x \in C}{\operatorname{argmin}} \left( \langle x, \nabla f(x_k) \rangle + \alpha_k B_\psi(x, x_k) \right) \tag{6}$$

    where $B_\psi(x, y) = \psi(x) - \psi(y) - \langle x - y, \nabla \psi(y) \rangle$.
  - Converge to the optimal point with proper choices of $\alpha_k$.
  - Apply to solve 5

$$\mu_{k+1} = \underset{\mu \in \Delta}{\operatorname{argmax}} \left( \rho(\mu) - \alpha_k B_\psi(\mu, \mu_k) \right). \tag{7}$$

# Optimization algorithms

- Mirror descent.
- Dual Averaging
  - An iterative algorithm, each step

$$x_{k+1} = \underset{x \in \mathcal{X}}{\operatorname{argmin}}\left(\langle \frac{1}{k+1} \sum_{\tau=0}^{k} \nabla f(x_\tau), x\rangle + \frac{\alpha_k}{k+1} R(x)\right) \tag{8}$$

    where $R$ is a strongly convex function.
  - Apply to solve 5

$$\mu_{k+1} = \underset{\mu \in \Delta}{\operatorname{argmax}} \left(\rho(\mu) - \alpha_k R(\mu)\right). \tag{9}$$

# Regularization functions

- Negative Shannon entropy $R_S(\mu) = \sum_{x,a} \mu(x,a) \log \mu(x,a)$
- Negative conditional entropy $R_C(\mu) = \sum_{x,a} \nu_\mu(x,a)\pi_\mu(a|x) \log \pi_\mu(a|x)$
- ... and their Bregman divergence

$$D_S(\mu \| \mu') = \sum_{x,a} \mu(x,a) \log \frac{\mu(x,a)}{\mu'(x,a)} \tag{10}$$

,

$$D_C(\mu \| \mu') = \sum_{x} \nu_\pi(x) \sum_{a} \pi(a|x) \log \frac{\pi(x,a)}{\pi'(x,a)}. \tag{11}$$

# Unified view of REPS, TRPO and A3C

- REPS: Mirror descent + $D_S$
  - REPS update

$$\mu_{\pi_{k+1}} = \underset{\mu \in \Delta}{\operatorname{argmax}}\ \rho(\mu)$$
$$\text{subject to} \quad DL_{KL}(\mu \| q) \leq \delta.$$

  - Use Lagrangian strong duality

$$\mu_{\pi_{k+1}} = \underset{\mu \in \Delta}{\operatorname{argmax}} \left(\rho(\mu) - \eta DL_{KL}(\mu \| q)\right) \tag{12}$$

    for some $\eta > 0$.
  - Converge to an optimal policy.

# Unified view of REPS, TRPO and A3C

- TRPO: Mirror descent + $D_C$
  - Switch $\pi_{\theta_{old}}$ and $\pi_{\theta_{new}}$ in the hard constraint then use Lagrangian strong duality

$$\begin{aligned}
\pi_{k+1} &= \operatorname*{argmax}_{\pi} \sum_{x} \nu_{\pi}(x) \sum_{a} \pi(a|x) \Big( A^{\pi_k}(x,a) - \eta \log \frac{\pi(x|a)}{\pi_k(x|a)} \Big) \\
&= \operatorname*{argmax}_{\pi} \Big( \rho(\pi) - \eta \sum_{x} \nu_{\pi}(x) \sum_{a} \pi(a|x) \log \frac{\pi(x|a)}{\pi_k(x|a)} \Big) \\
&= \operatorname*{argmax}_{\pi} \Big( \rho(\pi) - \eta D_C(\pi \| \pi_k) \Big)
\end{aligned}$$

# Unified view of REPS, TRPO and A3C

- TRPO: Mirror descent + $D_C$
  - Switch $\pi_{\theta_{old}}$ and $\pi_{\theta_{new}}$ in the hard constraint then use Lagrangian strong duality

$$\pi_{k+1} = \operatorname*{argmax}_{\pi} \Big( \rho(\pi) - \eta D_C(\pi \| \pi_k) \Big)$$

  - The exact update

$$\pi_{k+1}(x|a) \propto \pi_k(x|a) e^{\frac{A^{\pi_k}(x,a)}{\eta}}$$

is equivalent to MDP Expert Algorithm of Eyal Even-Dar et al.

# Unified view of REPS, TRPO and A3C

- A3C Dual averaging + $R_C$
  - A3C update rule

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta \Big( \rho(\mu_\theta) - \eta \sum_{\mathcal{X}} \nu_{\pi_k} \sum_{\mathcal{A}} \pi_\theta(a|x) \log \pi_\theta(a|x) \Big).$$

  - A3C objective

$$\rho(\pi_\theta) - \eta \sum_{\mathcal{X}} \nu_{\pi_\theta}(x) \sum_{\mathcal{A}} \pi_\theta(a|x) \log \pi_\theta(a|x) = \rho(\mu_\theta) - \eta R_C(\mu_\theta).$$

  - Does not guarantee to converge.

# Content

# TRPO Algorithm

$$\begin{aligned} &\underset{\theta}{\text{maximize}}\, \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}, a\sim q}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} Q_{\theta_{\text{old}}}(s,a)\right] \\ &\text{subject to } \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}}\left[D_{\text{KL}}(\pi_{\theta_{\text{old}}}(\cdot|s) \parallel \pi_\theta(\cdot|s))\right] \leq \delta. \end{aligned} \tag{13}$$

$$\overline{D}_{\mathrm{KL}}(\theta_{\mathrm{old}}, \theta) \approx \frac{1}{2}(\theta - \theta_{\mathrm{old}})^T A(\theta - \theta_{\mathrm{old}})$$

**Require:** A parameterized policy $\pi_\theta$ and $\theta = \theta_0$

**for** $i \leftarrow 1$ to $T$ **do**

Run policy for N trajectories

Estimate advantage function at all time steps

Compute objective gradient $g_\theta$

Compute A

Use Conjugate Gradient to compute $\beta$ and $s$

Compute the rescaled update line search

Apply update to $\theta$

**end for**

# Our results

We were able to reproduce results for some Atari Games but by using different neural structures for policies.

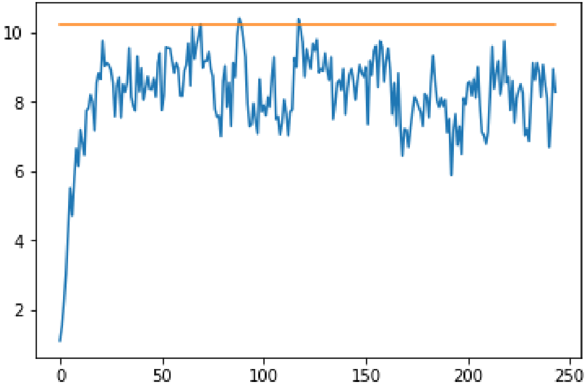


Figure: Average reward for Breakout, Fully connected layers

# Our results

To be compared with 1908 score in the paper

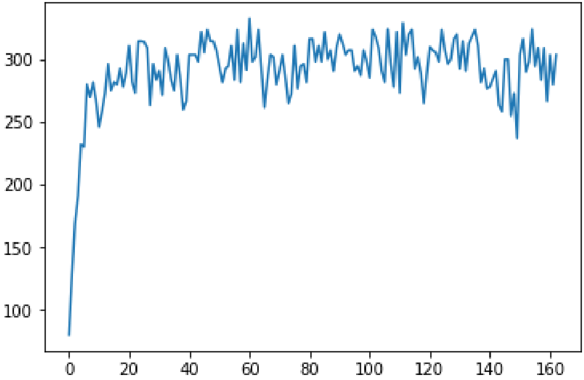


Figure: Average reward for Seaquest, Fully connected layers

# Thank you