Trust-Region Policy Optimization
Christopher Mutschler
Policy Gradients so far

Loop forever:
1. collect trajectories via policy \( \pi_{\theta} \)
2. Estimate advantage function \( A^{\pi_{\theta}}(a_t|s_t) \)
3. Compute policy gradient:
   \[
   \nabla J(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left( \sum_t \nabla_{\theta} \log \pi_{\theta}(a_t|s_t) A^{\pi_{\theta}}(a_t|s_t) \right)
   \]
4. Update policy parameters \( \theta_{\text{new}} \leftarrow \theta + \alpha \nabla J(\theta) \)

http://ai.berkeley.edu/lecture_slides.html
Policy Gradients so far

Problem

- Run gradient descent/ascent on one batch of collected experience

- Note: the advantage function (which is a noisy estimate) may not be accurate
  - Too large steps may lead to a disaster (even if the gradient is correct)
  - Too small steps are also bad

- Definition and scheduling of learning rates in RL is tricky as the underlying data distribution changes with updates to the policy

- Mathematical formulization:
  - First-order derivatives approximate the (parameter) surface to be flat
  - But if the surface exhibits high curvature it gets dangerous
  - Projection: small changes in parameter space might lead to large changes in policy space!

- Parameters $\theta$ get updated to areas too far out of the range from where previous data was collected (note: a bad policy leads to bad data)

Trust-Region Policy Optimization (TRPO)

“Simple” Idea

Regularize updates to the policy parameters, such that the policy does not change too much.
Motivation: Why trust region optimization?

Image credit: https://medium.com/@jonathan_hui/rl-trust-region-policy-optimization-trpo-explained-a6ee04eeeee9

Line search
(like gradient ascent)

Trust region
Primer: Trust-Region Methods

Optimization in Machine Learning: two classes

1. Line Search, e.g., gradient descent
   - find a (some) direction of improvement
   - (cleverly) select a step length

2. Trust-Region Methods
   - select a trust region (analog to max step length)
   - find a point of improvement in that region
Primer: Trust-Region Methods

• Idea:
  • Approximate the real objective $f$ with something simpler, i.e., $\tilde{f}$
  • Solve $\tilde{x}^* = \arg\min_x \tilde{f}(x)$

• Problem:
  • The optimum $\tilde{x}^*$ might be in a region where $\tilde{f}$ poorly approximates $f$
  • $\tilde{x}^*$ might be far from optimal

• Solution:
  • Restrict the search to a region $tr$ where we trust $\tilde{f}$ to approximate $f$ well
  • Solve $\tilde{x}^* = \arg\min_{x \in tr} \tilde{f}(x)$
Trust-Region Policy Optimization (TRPO)

So back to what we actually do…
The problem(s) of the Policy Gradient (PG) is that
- PG keeps old and new policy close in parameter space, while
- small changes can lead to large differences in performance, and
- “large” step-sizes hurt performance (whatever “large” means…)

Loop forever:
1. collect trajectories via policy $\pi_\theta$
2. Estimate advantage function $A^{\pi_\theta}(s_t|a_t)$
3. Compute policy gradient:
   \[
   \nabla J(\theta) = \mathbb{E}_{t \sim \pi_\theta} \left( \sum_t \nabla_\theta \log \pi_\theta(a_t|s_t) A^{\pi_\theta}(s_t|a_t) \right)
   \]
4. Update policy parameters $\theta_{\text{new}} \leftarrow \theta + a \nabla J(\theta)$

Update carefully → We want improvement and not degradation

Non-stationary input data due to changing policy and reward distribution change

random at beginning
Trust-Region Policy Optimization (TRPO)

• We want to optimize $\eta(\pi)$, i.e., the expected return of policy $\pi$:

$$\eta(\pi) = \mathbb{E}_{s_0 \sim \rho_0,a^t \sim \pi_{old}(\cdot|s_t)} \left[ \sum_{t=0}^{\infty} \gamma^t r_t \right]$$

• We collect data with $\pi_{old}$ and optimize to get a new policy $\pi_{new}$

• Let’s express $\eta(\pi_{new})$ in terms of advantage over the original policy\(^1\):

$$\eta(\pi_{new}) = \eta(\pi_{old}) + \mathbb{E}_{t \sim \pi_{new}} \left[ \sum_{t=0}^{\infty} \gamma^t A_{\pi_{old}}(s_t, a_t) \right]$$

Trust-Region Policy Optimization (TRPO)

• We want to optimize $\eta(\pi)$, i.e., the expected return of policy $\pi$:

$$
\eta(\pi) = \mathbb{E}_{s_0 \sim \rho_0, a^t \sim \pi_{old} \mid s_t} \left[ \sum_{t=0}^{\infty} \gamma^t r_t \right]
$$

• We collect data with $\pi_{old}$ and optimize to get a new policy $\pi_{new}$

• Let’s express $\eta(\pi_{new})$ in terms of advantage over the original policy:

$$
\eta(\pi_{new}) = \eta(\pi_{old}) + \mathbb{E}_{\tau \sim \pi_{new}} \left[ \sum_{t=0}^{\infty} \gamma^t A_{\pi_{old}}(s_t, a_t) \right]
$$

$$
= \eta(\pi_{old}) + \sum_s \rho_{\pi_{new}}(s) \sum_a \pi_{new}(a \mid s) A_{\pi_{old}}(s, a)
$$

Discounted visitation frequency according to new policy:

$$
\rho_{\pi_{new}}(s) = P(s_0 = s) + \gamma P(s_1 = s) + \gamma^2 P(s_2 = s) + \ldots
$$

Trust-Region Policy Optimization (TRPO)

• We want to optimize \( \eta(\pi) \), i.e., the expected return of policy \( \pi \):

\[
\eta(\pi) = \mathbb{E}_{s_0 \sim \rho_0, a_t \sim \pi_{old} \mid s_t} \left[ \sum_{t=0}^{\infty} \gamma^t r_t \right]
\]

• We collect data with \( \pi_{old} \) and optimize to get a new policy \( \pi_{new} \)

• Let’s express \( \eta(\pi_{new}) \) in terms of advantage over the original policy:

\[
\eta(\pi_{new}) = \eta(\pi_{old}) + \mathbb{E}_{t \sim \pi_{new}} \left[ \sum_{t=0}^{\infty} \gamma^t A_{\pi_{old}}(s_t, a_t) \right]
\]

\[
= \eta(\pi_{old}) + \sum_s \rho_{\pi_{new}}(s) \sum_a \pi_{new}(a \mid s) A_{\pi_{old}}(s, a)
\]

\[> 0 \quad \text{If we can guarantee this…} \]

→ New objective guarantees improvement from \( \pi_{old} \rightarrow \pi_{new} \)
Trust-Region Policy Optimization (TRPO)

- We want to optimize $\eta(\pi)$, i.e., the expected return of policy $\pi$:

$$
\eta(\pi) = \mathbb{E}_{s_0 \sim \rho_0, a_t \sim \pi_{old}(\cdot | s_t)} \left[ \sum_{t=0}^{\infty} \gamma^t r_t \right]
$$

- We collect data with $\pi_{old}$ and optimize to get a new policy $\pi_{new}$

- Let’s express $\eta(\pi_{new})$ in terms of advantage over the original policy:

$$
\eta(\pi_{new}) = \eta(\pi_{old}) + \mathbb{E}_{t \sim \pi_{new}} \left[ \sum_{t=0}^{\infty} \gamma^t A_{\pi_{old}}(s_t, a_t) \right]
$$

$$
= \eta(\pi_{old}) + \sum_s \rho_{\pi_{new}}(s) \sum_a \pi_{new}(a | s) A_{\pi_{old}}(s, a)
$$

However, this cannot be easily estimated. The state visitations that we sampled so far are coming from the old policy!

⇒ we cannot optimize this in the current form!
Trust-Region Policy Optimization (TRPO)

\[
\eta(\pi_{new}) = \eta(\pi_{old}) + \sum_s \rho_{\pi_{new}}(s) \sum_a \pi_{new}(a|s) A_{\pi_{old}}(s, a) \\
\approx \sum_s \rho_{\pi_{old}}(s) \sum_a \pi_{new}(a|s) A_{\pi_{old}}(s, a)
\]

• The approximation is accurate within step size \( \delta \) (trust region)
  • \( \delta \) needs to be chosen based on a lower-bound approximation error
• Monotonic improvement guaranteed
  • (within the green region!)

This we already sampled
→ We already have this!

\( \theta_{new} \)
\( \theta_{old} \)
\( \theta_{new} \)

Trust region
→ \( \pi_{\theta_{new}}(s, a) \) does not change too much
Trust-Region Policy Optimization (TRPO)

• If we want to optimize $L(\theta_{new})$ instead of $\eta(\theta_{new})$ …
  with a guarantee of monotonic improvement on $\eta(\theta_{new})$, …
  … we need a bound on $L(\theta_{new})$.

• It can be proven that there exists the following bound\textsuperscript{1,2}:

$$\eta(\pi_{new}) \geq L(\pi_{new}) - C \cdot D_{KL}^{max}(\pi_{old}, \pi_{new}), \text{ where } C = \frac{4\epsilon\gamma}{(1-\gamma)^2}$$

\textsuperscript{1} Schulman et al.: Trust-Region Policy Optimization. ICML 2015.
Trust-Region Policy Optimization (TRPO)

• A monotonically increasing policy can be defined by (minorization-maximization algorithm):

\[ \pi = \arg \max_\pi \left[ L(\pi_{\text{new}}) - C \cdot D_{KL}^{\text{max}}(\pi_{\text{old}}, \pi_{\text{new}}) \right], \text{ where } C = \frac{4\epsilon\gamma}{(1-\gamma)^2} \]

Side-note:
• A constraint on the KL-divergence between new and old policy (i.e., a trust region constraint) allows larger step sizes while being mathematically equivalent:

\[ \pi = \arg \max_\pi L_{\pi_{\text{old}}}, \text{ such that } D_{KL}^{\text{max}}(\pi_{\text{old}}, \pi) \leq \delta \]

• Approximation with \( L \) is accurate within \( \delta \)
  \( \rightarrow \) here, monotonic improvement guaranteed

Trust region
\( \rightarrow \pi_{\theta_{\text{new}}}(s, a) \) does not change too much
PPO (clipping version)

\[ L(s, a, \theta_k, \theta) = \min \left( \frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_\theta_k}(s, a), \ g(\epsilon, A^{\pi_\theta_k}(s, a)) \right), \]

where

\[ g(\epsilon, A) = \begin{cases} (1 + \epsilon)A & A \geq 0 \\ (1 - \epsilon)A & A < 0. \end{cases} \]