Advanced Policy Gradient

Reinforcement Learning

D. Sorokin

March 24, 2025

Bias-Variance trade-off

Reminder: Policy Gradient

$$J(\pi_\theta) := \mathbb{E}_{\mathcal{T}\sim\pi_\theta} \sum_{t\geqslant 0} \gamma^t r_t$$

Reminder: Policy Gradient

$$J(\pi_\theta) := \mathbb{E}_{\mathcal{T}\sim\pi_\theta} \sum_{t\geqslant 0} \gamma^t r_t$$

$$\nabla_\theta J(\pi_\theta) \approx \underbrace{\mathbb{E}_{\mathcal{T}\sim\pi_\theta} \sum_{t\geqslant 0}}_{\substack{\text{data generated by } \pi_\theta \\ \text{(sample pairs } s,a \text{ from trajectories } \mathcal{T}\sim\pi_\theta)}} \overbrace{\nabla_\theta \log \pi_\theta(a_t \mid s_t)}^{\text{log-likelihood}} \underbrace{(Q^{\pi_\theta}(s_t,a_t) - \overbrace{V^{\pi_\theta}(s_t)}^{\text{baseline}})}_{\substack{A^{\pi_\theta}(s_t,a_t) \\ \text{critic estimation}}}$$

Reminder: Policy Gradient

$$J(\pi_\theta) := \mathbb{E}_{\mathcal{T}\sim\pi_\theta} \sum_{t\geqslant 0} \gamma^t r_t$$

$$\nabla_\theta J(\pi_\theta) \approx \underbrace{\mathbb{E}_{\mathcal{T}\sim\pi_\theta} \sum_{t\geqslant 0}}_{\substack{\text{data generated by } \pi_\theta \\ \text{(sample pairs } s,a \text{ from trajectories } \mathcal{T}\sim\pi_\theta)}} \overbrace{\nabla_\theta \log \pi_\theta(a_t \mid s_t)}^{\text{log-likelihood}} \underbrace{(Q^{\pi_\theta}(s_t, a_t) - \overbrace{V^{\pi_\theta}(s_t)}^{\text{baseline}})}_{\substack{A^{\pi_\theta}(s_t,a_t) \\ \text{critic estimation}}}$$

Critic: an approximation $V_\phi(s) \approx V^\pi(s)$.

Important: it is a **biased** estimation.

Reminder: Policy Gradient

$$J(\pi_\theta) := \mathbb{E}_{\mathcal{T}\sim\pi_\theta} \sum_{t\geqslant 0} \gamma^t r_t$$

$$\nabla_\theta J(\pi_\theta) \approx \underbrace{\mathbb{E}_{\mathcal{T}\sim\pi_\theta} \sum_{t\geqslant 0}}_{\substack{\text{data generated by } \pi_\theta \\ \text{(sample pairs } s,a \text{ from trajectories } \mathcal{T}\sim\pi_\theta)}} \overbrace{\nabla_\theta \log \pi_\theta(a_t \mid s_t)}^{\text{log-likelihood}} \underbrace{(Q^{\pi_\theta}(s_t,a_t) - \overbrace{V^{\pi_\theta}(s_t)}^{\text{baseline}})}_{\substack{A^{\pi_\theta}(s_t,a_t) \\ \text{critic estimation}}}$$

Critic: an approximation $V_\phi(s) \approx V^\pi(s)$.
Important: it is a **biased** estimation.

$Q^{\pi_\theta}(s_t,a_t) \approx y_Q = r + \gamma V_\phi(s_{t+1})$ is also biased.

Bias-Variance trade-off

Given rollout $s, r, s', r', s'', r'' \dots s^{(M)}$ from policy π and approximation of $V(s)$

Bias-Variance trade-off

Given rollout $s, r, s', r', s'', r'' \dots s^{(M)}$ from policy π and approximation of $V(s)$ perform **credit assingment** for state-action pair s, a (was this decision good or bad?)

Bias-Variance trade-off

Given rollout $s, r, s', r', s'', r'' \dots s^{(M)}$ from policy π and approximation of $V(s)$ perform **credit assingment** for state-action pair s, a (was this decision good or bad?)

For Actor:

$$\nabla := \nabla_\theta \log \pi_\theta(a \mid s) \underbrace{\Psi(s,a)}_{\substack{\text{advantage} \\ \text{estimator}}}$$

For Critic:

$$\underbrace{y_Q}_{\substack{\text{target} \\ \text{for regression}}} := \Psi(s,a) + V(s)$$

Bias-Variance trade-off

Given rollout $s, r, s', r', s'', r'' \dots s^{(M)}$ from policy π and approximation of $V(s)$ perform **credit assingment** for state-action pair s, a (was this decision good or bad?)

For Actor:

$$\nabla := \nabla_\theta \log \pi_\theta(a \mid s) \underbrace{\Psi(s,a)}_{\substack{\text{advantage} \\ \text{estimator}}}$$

For Critic:

$$\underbrace{y_Q}_{\substack{\text{target} \\ \text{for regression}}} := \Psi(s,a) + V(s)$$

	$\Psi(s,a)$	**Bias**	**Variance**
Monte Carlo	$\Psi_{(\infty)}(s,a) := r + \gamma r' + \gamma^2 r'' + \dots - V(s)$		

Bias-Variance trade-off

Given rollout $s, r, s', r', s'', r'' \dots s^{(M)}$ from policy π and approximation of $V(s)$ perform **credit assingment** for state-action pair s, a (was this decision good or bad?)

For Actor:

$$\nabla := \nabla_\theta \log \pi_\theta(a \mid s) \underbrace{\Psi(s,a)}_{\substack{\text{advantage} \\ \text{estimator}}}$$

For Critic:

$$\underbrace{y_Q}_{\substack{\text{target} \\ \text{for regression}}} := \Psi(s,a) + V(s)$$

	$\Psi(s,a)$	**Bias**	**Variance**
Monte Carlo	$\Psi_{(\infty)}(s,a) := r + \gamma r' + \gamma^2 r'' + \dots - V(s)$	0	high

Bias-Variance trade-off

Given rollout $s, r, s', r', s'', r'' \dots s^{(M)}$ from policy π and approximation of $V(s)$ perform **credit assingment** for state-action pair s, a (was this decision good or bad?)

For Actor:

$$\nabla := \nabla_\theta \log \pi_\theta(a \mid s) \underbrace{\Psi(s,a)}_{\substack{\text{advantage} \\ \text{estimator}}}$$

For Critic:

$$\underbrace{y_Q}_{\substack{\text{target} \\ \text{for regression}}} := \Psi(s,a) + V(s)$$

	$\Psi(s,a)$	**Bias**	**Variance**
Monte Carlo	$\Psi_{(\infty)}(s,a) := r + \gamma r' + \gamma^2 r'' + \dots - V(s)$	0	high
1-step	$\Psi_{(1)}(s,a) := r + \gamma V(s') - V(s)$		

Bias-Variance trade-off

Given rollout $s, r, s', r', s'', r'' \dots s^{(M)}$ from policy π and approximation of $V(s)$ perform **credit assingment** for state-action pair s, a (was this decision good or bad?)

For Actor:

$$\nabla := \nabla_\theta \log \pi_\theta(a \mid s) \underbrace{\Psi(s,a)}_{\substack{\text{advantage} \\ \text{estimator}}}$$

For Critic:

$$\underbrace{y_Q}_{\substack{\text{target} \\ \text{for regression}}} := \Psi(s,a) + V(s)$$

	$\Psi(s,a)$	**Bias**	**Variance**
Monte Carlo	$\Psi_{(\infty)}(s,a) := r + \gamma r' + \gamma^2 r'' + \dots - V(s)$	0	high
1-step	$\Psi_{(1)}(s,a) := r + \gamma V(s') - V(s)$	high	low

Bias-Variance trade-off

Given rollout $s, r, s', r', s'', r'' \dots s^{(M)}$ from policy π and approximation of $V(s)$ perform **credit assingment** for state-action pair s, a (was this decision good or bad?)

For Actor:

$$\nabla := \nabla_\theta \log \pi_\theta(a \mid s) \underbrace{\Psi(s,a)}_{\substack{\text{advantage} \\ \text{estimator}}}$$

For Critic:

$$\underbrace{y_Q}_{\substack{\text{target} \\ \text{for regression}}} := \Psi(s,a) + V(s)$$

	$\Psi(s,a)$	**Bias**	**Variance**
Monte Carlo	$\Psi_{(\infty)}(s,a) := r + \gamma r' + \gamma^2 r'' + \dots - V(s)$	0	high
N-step	$\Psi_{(N)}(s,a) := r + \gamma r' + \dots + \gamma^N V(s^{(N)}) - V(s)$	intermediate	intermediate
1-step	$\Psi_{(1)}(s,a) := r + \gamma V(s') - V(s)$	high	low

Bias-Variance trade-off

Given rollout $s, r, s', r', s'', r'' \dots s^{(M)}$ from policy π and approximation of $V(s)$ perform **credit assingment** for state-action pair s, a (was this decision good or bad?)

For Actor:

$$\nabla := \nabla_\theta \log \pi_\theta(a \mid s) \underbrace{\Psi(s,a)}_{\substack{\text{advantage} \\ \text{estimator}}}$$

For Critic:

$$\underbrace{y_Q}_{\substack{\text{target} \\ \text{for regression}}} := \Psi(s,a) + V(s)$$

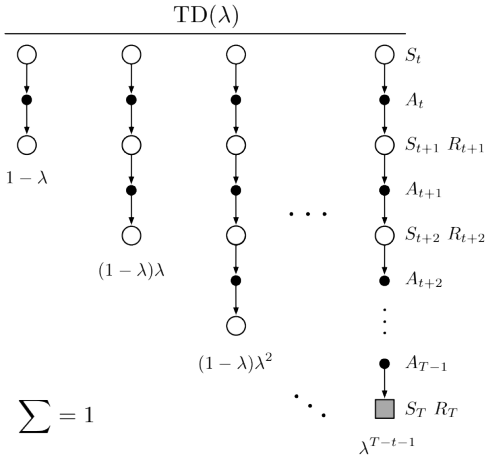
	$\Psi(s,a)$	**Bias**	**Variance**
Monte Carlo	$\Psi_{(\infty)}(s,a) := r + \gamma r' + \gamma^2 r'' + \dots - V(s)$	0	high
N-step	$\Psi_{(N)}(s,a) := r + \gamma r' + \dots + \gamma^N V(s^{(N)}) - V(s)$	intermediate	intermediate
1-step	$\Psi_{(1)}(s,a) := r + \gamma V(s') - V(s)$	high	low

Problem: hard to choose N.

Reminder: TD(λ)

Can't choose? Use them all.

Estimation	Weight
$\Psi_{(1)}(s,a)$	$1-\lambda$
$\Psi_{(2)}(s,a)$	$(1-\lambda)\lambda$
$\Psi_{(3)}(s,a)$	$(1-\lambda)\lambda^2$
$\vdots$	$\vdots$



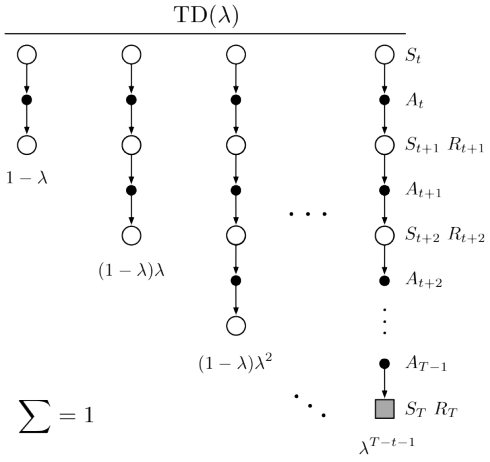
Reminder: TD(λ)

Can't choose? Use them all.

Estimation	Weight
$\Psi_{(1)}(s,a)$	$1-\lambda$
$\Psi_{(2)}(s,a)$	$(1-\lambda)\lambda$
$\Psi_{(3)}(s,a)$	$(1-\lambda)\lambda^2$
$\vdots$	$\vdots$

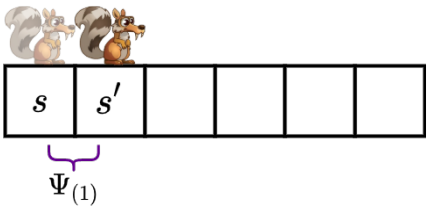
Generalized Advantage Estimation:

$$\Psi_{\mathrm{GAE}}(s,a) = (1-\lambda)\sum_{N=1}^{\infty}\lambda^{N-1}\Psi_{(N)}(s,a)$$



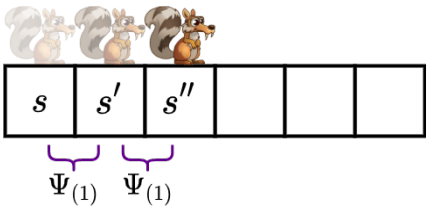
Discounted sum of TD-errors

$$\overbrace{r + \gamma V(s') - V(s)}^{\Psi_{(1)}(s,a)} +$$



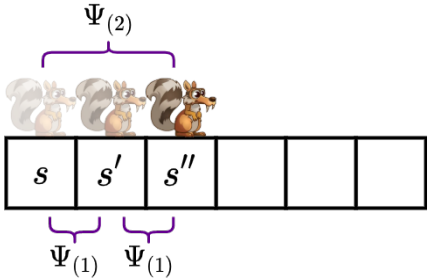
Discounted sum of TD-errors

$$\overbrace{r + \gamma V(s') - V(s)}^{\Psi_{(1)}(s,a)} + \overbrace{\gamma r' + \gamma^2 V(s'') - \gamma V(s')}^{\gamma \Psi_{(1)}(s',a')}$$



Discounted sum of TD-errors

$$\overbrace{r + \gamma V(s') - V(s)}^{\Psi_{(1)}(s,a)} + \overbrace{\gamma r' + \gamma^2 V(s'') - \gamma V(s')}^{\gamma\Psi_{(1)}(s',a')} = \Psi_{(2)}(s,a)$$

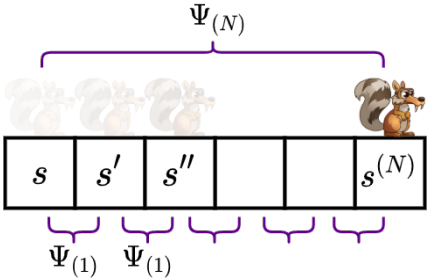


Discounted sum of TD-errors

$$\overbrace{r + \gamma V(s') - V(s)}^{\Psi_{(1)}(s,a)} + \overbrace{\gamma r' + \gamma^2 V(s'') - \gamma V(s')}^{\gamma\Psi_{(1)}(s',a')} = \Psi_{(2)}(s,a)$$

N-step error is a sum of 1-step errors

$$\Psi_{(N)}(s,a) = \sum_{t=0}^{N} \gamma^t \Psi_{(1)}(s^{(t)}, a^{(t)})$$

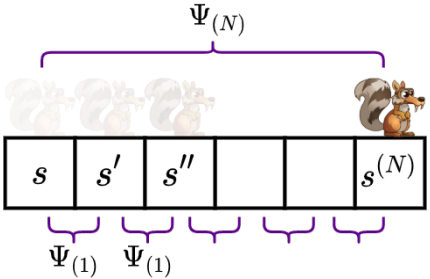


Discounted sum of TD-errors

$$\overbrace{r + \gamma V(s') - V(s)}^{\Psi_{(1)}(s,a)} + \overbrace{\gamma r' + \gamma^2 V(s'') - \gamma V(s')}^{\gamma\Psi_{(1)}(s',a')} = \Psi_{(2)}(s,a)$$

N-step error is a sum of 1-step errors

$$\Psi_{(N)}(s,a) = \sum_{t=0}^{N} \gamma^t \Psi_{(1)}(s^{(t)}, a^{(t)})$$



Rule of thumb: discounted sum of discounted sums is discounted sum =D

Generalized Advantage Estimation (GAE)

Equivalent form of GAE

$$(1-\lambda)\sum_{N=1}^{\infty}\lambda^{N-1}\Psi_{(N)}(s,a) =$$

Generalized Advantage Estimation (GAE)

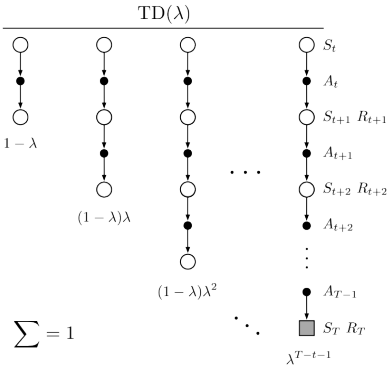
Equivalent form of GAE

$$(1-\lambda)\sum_{N=1}^{\infty}\lambda^{N-1}\Psi_{(N)}(s,a) = \sum_{t=0}^{\infty}(\gamma\lambda)^{t}\Psi_{(1)}(s^{(t)},a^{(t)})$$

Generalized Advantage Estimation (GAE)

Equivalent form of GAE

$$(1-\lambda)\sum_{N=1}^{\infty}\lambda^{N-1}\Psi_{(N)}(s,a) = \sum_{t=0}^{\infty}(\gamma\lambda)^{t}\Psi_{(1)}(s^{(t)},a^{(t)})$$

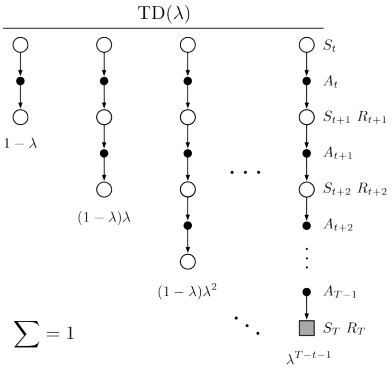


What if for some pair s, a we do not know our future until the end of episode, but only T steps ahead?

Generalized Advantage Estimation (GAE)

Equivalent form of GAE

$$(1-\lambda)\sum_{N=1}^{\infty}\lambda^{N-1}\Psi_{(N)}(s,a) = \sum_{t=0}^{\infty}(\gamma\lambda)^t\Psi_{(1)}(s^{(t)},a^{(t)})$$



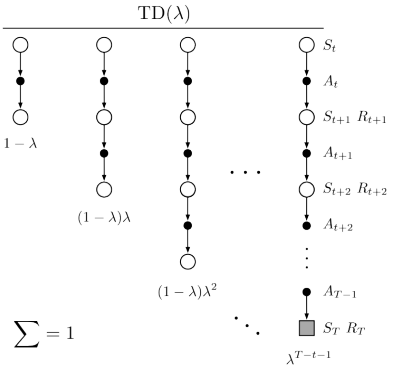
What if for some pair s, a we do not know our future until the end of episode, but only T steps ahead?

$$\Psi^{\mathrm{GAE}}(s,a) := \sum_{t=0}^{T}(\gamma\lambda)^t\Psi_{(1)}(s^{(t)},a^{(t)})$$

Generalized Advantage Estimation (GAE)

Equivalent form of GAE

$$(1-\lambda)\sum_{N=1}^{\infty}\lambda^{N-1}\Psi_{(N)}(s,a) = \sum_{t=0}^{\infty}(\gamma\lambda)^t\Psi_{(1)}(s^{(t)},a^{(t)})$$



What if for some pair s, a we do not know our future until the end of episode, but only T steps ahead?

$$\Psi^{\mathrm{GAE}}(s,a) := \sum_{t=0}^{T}(\gamma\lambda)^t\Psi_{(1)}(s^{(t)},a^{(t)})$$

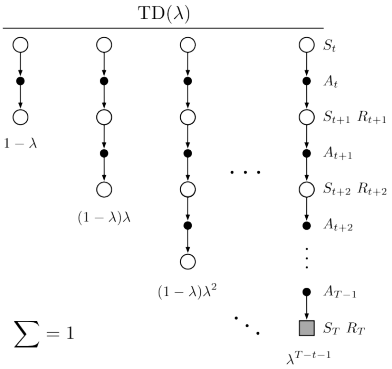
Equation used in practice:

$$\Psi^{\mathrm{GAE}}(s_t,a_t) = \Psi_{(1)}(s_t,a_t) + \\ + \lambda\gamma(1-\mathrm{done}_{t+1})\Psi^{\mathrm{GAE}}(s_{t+1},a_{t+1})$$

Generalized Advantage Estimation (GAE)

Equivalent form of GAE

$$(1-\lambda)\sum_{N=1}^{\infty}\lambda^{N-1}\Psi_{(N)}(s,a) = \sum_{t=0}^{\infty}(\gamma\lambda)^t\Psi_{(1)}(s^{(t)},a^{(t)})$$



What if for some pair s, a we do not know our future until the end of episode, but only T steps ahead?

$$\Psi^{\mathrm{GAE}}(s,a) := \sum_{t=0}^{T}(\gamma\lambda)^t\Psi_{(1)}(s^{(t)},a^{(t)})$$

Equation used in practice:

$$\Psi^{\mathrm{GAE}}(s_t,a_t) = \Psi_{(1)}(s_t,a_t) + \\ + \lambda\gamma(1-\mathrm{done}_{t+1})\Psi^{\mathrm{GAE}}(s_{t+1},a_{t+1})$$

Data collection in A2C

Question: Can we have Monte Carlo estimation of $\mathbb{E}_{\mathcal{T}\sim\pi_\theta}\sum_{t\geqslant 0}(\cdot)$ without playing full episodes?

Data collection in A2C

Question: Can we have Monte Carlo estimation of $\mathbb{E}_{\mathcal{T}\sim\pi_\theta}\sum_{t\geqslant 0}(\cdot)$ without playing full episodes?

State Visitation Frequency

$$\mathbb{E}_{\mathcal{T}\sim\pi_\theta}\sum_{t\geqslant 0}\gamma^t f(s_t, a_t) =$$

Data collection in A2C

Question: Can we have Monte Carlo estimation of $\mathbb{E}_{\mathcal{T}\sim\pi_\theta}\sum_{t\geqslant 0}(\cdot)$ without playing full episodes?

State Visitation Frequency

$$\mathbb{E}_{\mathcal{T}\sim\pi_\theta}\sum_{t\geqslant 0}\gamma^t f(s_t, a_t) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{\pi_\theta}(s)}\mathbb{E}_{a\sim\pi_\theta(a|s)} f(s, a),$$

where $d_\pi(s)$ is **state visitation frequency** of policy π.

Data collection in A2C

Question: Can we have Monte Carlo estimation of $\mathbb{E}_{\mathcal{T}\sim\pi_\theta}\sum_{t\geqslant 0}(\cdot)$ without playing full episodes?

State Visitation Frequency

$$\mathbb{E}_{\mathcal{T}\sim\pi_\theta}\sum_{t\geqslant 0}\gamma^t f(s_t,a_t)=\frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{\pi_\theta}(s)}\mathbb{E}_{a\sim\pi_\theta(a|s)}f(s,a),$$

where $d_\pi(s)$ is **state visitation frequency** of policy π.

We can **sample** from $d_\pi(s)$ by interacting with environment using π.

Data collection in A2C

Question: Can we have Monte Carlo estimation of $\mathbb{E}_{\mathcal{T}\sim\pi_\theta}\sum_{t\geqslant 0}(\cdot)$ without playing full episodes?

State Visitation Frequency

$$\mathbb{E}_{\mathcal{T}\sim\pi_\theta}\sum_{t\geqslant 0}\gamma^t f(s_t,a_t)=\frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{\pi_\theta}(s)}\mathbb{E}_{a\sim\pi_\theta(a|s)}f(s,a),$$

where $d_\pi(s)$ is **state visitation frequency** of policy π.

We can **sample** from $d_\pi(s)$ by interacting with environment using π.

But sequential states from one rollout are correlated.

Data collection in A2C

Question: Can we have Monte Carlo estimation of $\mathbb{E}_{\mathcal{T}\sim\pi_\theta}\sum_{t\geqslant 0}(\cdot)$ without playing full episodes?

State Visitation Frequency

$$\mathbb{E}_{\mathcal{T}\sim\pi_\theta}\sum_{t\geqslant 0}\gamma^t f(s_t, a_t) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{\pi_\theta}(s)}\mathbb{E}_{a\sim\pi_\theta(a|s)} f(s,a),$$

where $d_\pi(s)$ is **state visitation frequency** of policy π.

We can **sample** from $d_\pi(s)$ by interacting with environment using π.

But sequential states from one rollout are correlated.

$$\nabla_\theta J(\pi_\theta) = \frac{1}{1-\gamma}\underbrace{\mathbb{E}_{s\sim d_{\pi_\theta}(s)}\mathbb{E}_{a\sim\pi_\theta(a|s)}}_{\substack{\text{data generated by } \pi_\theta \\ \text{(sample pairs } s,a \text{ from trajectories } \mathcal{T}\sim\pi_\theta)}}\nabla_\theta \log\pi_\theta(a\mid s)A^{\pi_\theta}(s,a)$$

GAE in Advantage Actor-Critic

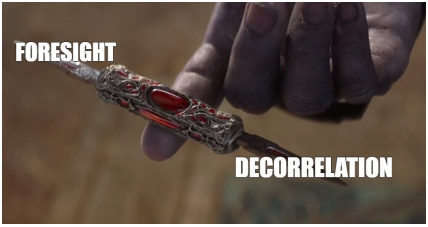
Longer rollouts produce richer GAE ensemble.

GAE in Advantage Actor-Critic

Longer rollouts produce richer GAE ensemble.

GAE in Advantage Actor-Critic

Longer rollouts produce richer GAE ensemble.



In A2C rollouts are usually short, so $\lambda = 1$ is common choice.
(sometimes called **max-trace** estimation)

Policy Gradient VS Value-based

Value-based (DQN+)

Policy Gradient

Policy Gradient VS Value-based

Value-based (DQN+)

✗ trains $Q^*(s, a)$;
(complicated intermediate stage)

Policy Gradient

✓ trains policy directly;
(requires only $V^\pi(s)$, which is much simpler)

Policy Gradient VS Value-based

Value-based (DQN+)

✗ trains $Q^*(s, a)$;
(complicated intermediate stage)

✗ exploration-exploitation issues;
(since Value Iteration works with deterministic policies)

Policy Gradient

✓ trains policy directly;
(requires only $V^\pi(s)$, which is much simpler)

✓ «natural» exploration;
(sampling from stochastic policy $\pi(a \mid s)$)

Policy Gradient VS Value-based

Value-based (DQN+)

✗ trains $Q^*(s, a)$;
(complicated intermediate stage)

✗ exploration-exploitation issues;
(since Value Iteration works with deterministic policies)

✗ 1-step targets;
(modifications try to solve this, but complicated corrections required)

Policy Gradient

✓ trains policy directly;
(requires only $V^\pi(s)$, which is much simpler)

✓ «natural» exploration;
(sampling from stochastic policy $\pi(a \mid s)$)

✓ ∞-step targets;
(can use GAE for both critic and actor)

Policy Gradient VS Value-based

Value-based (DQN+)

✗ trains $Q^*(s, a)$;
(complicated intermediate stage)

✗ exploration-exploitation issues;
(since Value Iteration works with deterministic policies)

✗ 1-step targets;
(modifications try to solve this, but complicated corrections required)

✓ off-policy;
(can use experience replay)

Policy Gradient

✓ trains policy directly;
(requires only $V^\pi(s)$, which is much simpler)

✓ «natural» exploration;
(sampling from stochastic policy $\pi(a \mid s)$)

✓ ∞-step targets;
(can use GAE for both critic and actor)

✗ on-policy;
(all data is useless after each SGD step)

Proximal Policy Optimization (PPO)

Off-policy Policy Gradient?

$$\nabla_\theta J(\pi_\theta) = \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi_\theta}(s)} \mathbb{E}_{a\sim\pi_\theta(a|s)}}_{\text{data generated by } \pi_\theta} \nabla_\theta \log \pi_\theta(a \mid s) A^{\pi_\theta}(s,a)$$

Suppose we:

- want to optimize π_θ (compute gradient for current θ);

Off-policy Policy Gradient?

$$\nabla_\theta J(\pi_\theta) = \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi_\theta}(s)} \mathbb{E}_{a\sim\pi_\theta(a|s)}}_{\text{data generated by } \pi_\theta} \nabla_\theta \log \pi_\theta(a \mid s) A^{\pi_\theta}(s,a)$$

Suppose we:

- want to optimize π_θ (compute gradient for current θ);
- have data (trajectory samples) from policy π^{old};

Off-policy Policy Gradient?

$$\nabla_\theta J(\pi_\theta) = \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi_\theta}(s)} \mathbb{E}_{a\sim\pi_\theta(a|s)}}_{\text{data generated by } \pi_\theta} \nabla_\theta \log \pi_\theta(a \mid s) A^{\pi_\theta}(s,a)$$

Suppose we:

- want to optimize π_θ (compute gradient for current θ);
- have data (trajectory samples) from policy π^{old};
  - i.e. we can estimate $\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}$ and $\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}$;

Off-policy Policy Gradient?

$$\nabla_\theta J(\pi_\theta) = \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi_\theta}(s)} \mathbb{E}_{a\sim\pi_\theta(a|s)}}_{\text{data generated by } \pi_\theta} \nabla_\theta \log \pi_\theta(a \mid s) A^{\pi_\theta}(s,a)$$

Suppose we:

- want to optimize π_θ (compute gradient for current θ);
- have data (trajectory samples) from policy π^{old};
  - i.e. we can estimate $\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}$ and $\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}$;
  - i.e. we can train $V^{\pi^{\text{old}}}(s)$ and thus estimate $A^{\pi^{\text{old}}}(s,a)$, using GAE;

Off-policy Policy Gradient?

$$\nabla_\theta J(\pi_\theta) = \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi_\theta}(s)} \mathbb{E}_{a\sim\pi_\theta(a|s)}}_{\text{data generated by } \pi_\theta} \nabla_\theta \log \pi_\theta(a \mid s) A^{\pi_\theta}(s,a)$$

Suppose we:

- want to optimize π_θ (compute gradient for current θ);
- have data (trajectory samples) from policy π^{old};
  - i.e. we can estimate $\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}$ and $\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}$;
  - i.e. we can train $V^{\pi^{\text{old}}}(s)$ and thus estimate $A^{\pi^{\text{old}}}(s,a)$, using GAE;

May be we can somehow express $J(\pi_\theta)$ using $A^{\pi^{\text{old}}}(s,a)$?

Connecting two policies

Let's perform **reward shaping** using another policy value function!

Connecting two policies

Let's perform **reward shaping** using another policy value function!

We will utilize the following trick (**telescoping sums**): for any sequence a_t, s.t. $\lim_{t\to\infty} a_t = 0$:

$$\sum_{t\geqslant 0}^{\infty} (a_{t+1} - a_t) = -a_0$$

Connecting two policies

Let's perform **reward shaping** using another policy value function!

We will utilize the following trick (**telescoping sums**): for any sequence a_t, s.t. $\lim_{t\to\infty} a_t = 0$:

$$\sum_{t\geqslant 0}^{\infty} (a_{t+1} - a_t) = -a_0$$

So, for arbitrary trajectory $\mathcal{T} \coloneqq s_0, a_0, s_1, a_1, \dots$ and arbitrary policy π:

$$-V^{\pi}(s_0) = \sum_{t\geqslant 0} \left[\gamma^{t+1} V^{\pi}(s_{t+1}) - \gamma^t V^{\pi}(s_t)\right] \tag{1}$$

Relative Performance Identity

$$V^{\pi_\theta}(s) - V^{\pi^{\text{old}}}(s) =$$

Relative Performance Identity

$$V^{\pi_\theta}(s) - V^{\pi^{\text{old}}}(s) = \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t r_t - V^{\pi^{\text{old}}}(s) =$$

Relative Performance Identity

$$V^{\pi_\theta}(s) - V^{\pi^{\text{old}}}(s) = \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t r_t - V^{\pi^{\text{old}}}(s) =$$

$$= \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \left[\sum_{t\geqslant 0} \gamma^t r_t - V^{\pi^{\text{old}}}(s_0) \right] =$$

Relative Performance Identity

$$V^{\pi_\theta}(s) - V^{\pi^{\text{old}}}(s) = \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t r_t - V^{\pi^{\text{old}}}(s) =$$

$$= \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \left[\sum_{t\geqslant 0} \gamma^t r_t - V^{\pi^{\text{old}}}(s_0) \right] =$$

$$\{\text{telescoping sums (1)}\} = \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \left[\sum_{t\geqslant 0} \gamma^t r_t + \sum_{t\geqslant 0} \left[\gamma^{t+1} V^{\pi^{\text{old}}}(s_{t+1}) - \gamma^t V^{\pi^{\text{old}}}(s_t) \right] \right] =$$

Relative Performance Identity

$$V^{\pi_\theta}(s) - V^{\pi^{\text{old}}}(s) = \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t r_t - V^{\pi^{\text{old}}}(s) =$$

$$= \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \left[\sum_{t\geqslant 0} \gamma^t r_t - V^{\pi^{\text{old}}}(s_0) \right] =$$

$$\{\text{telescoping sums (1)}\} = \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \left[\sum_{t\geqslant 0} \gamma^t r_t + \sum_{t\geqslant 0} \left[\gamma^{t+1} V^{\pi^{\text{old}}}(s_{t+1}) - \gamma^t V^{\pi^{\text{old}}}(s_t) \right] \right] =$$

$$\{\text{regroup terms}\} = \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t \left(r_t + \gamma V^{\pi^{\text{old}}}(s_{t+1}) - V^{\pi^{\text{old}}}(s_t) \right) =$$

Relative Performance Identity

$$V^{\pi_\theta}(s) - V^{\pi^{\text{old}}}(s) = \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t r_t - V^{\pi^{\text{old}}}(s) =$$

$$= \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \left[\sum_{t\geqslant 0} \gamma^t r_t - V^{\pi^{\text{old}}}(s_0) \right] =$$

$$\{\text{telescoping sums (1)}\} = \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \left[\sum_{t\geqslant 0} \gamma^t r_t + \sum_{t\geqslant 0} \left[\gamma^{t+1} V^{\pi^{\text{old}}}(s_{t+1}) - \gamma^t V^{\pi^{\text{old}}}(s_t) \right] \right] =$$

$$\{\text{regroup terms}\} = \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t \left(r_t + \gamma V^{\pi^{\text{old}}}(s_{t+1}) - V^{\pi^{\text{old}}}(s_t) \right) =$$

$$\{\text{trick } \mathbb{E}_x f(x) = \mathbb{E}_x \mathbb{E}_x f(x)\} = \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t \left(r_t + \gamma \mathbb{E}_{s_{t+1}} V^{\pi^{\text{old}}}(s_{t+1}) - V^{\pi^{\text{old}}}(s_t) \right) =$$

Relative Performance Identity

$$V^{\pi_\theta}(s) - V^{\pi^{\text{old}}}(s) = \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t r_t - V^{\pi^{\text{old}}}(s) =$$

$$= \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \left[\sum_{t\geqslant 0} \gamma^t r_t - V^{\pi^{\text{old}}}(s_0) \right] =$$

$$\{\text{telescoping sums (1)}\} = \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \left[\sum_{t\geqslant 0} \gamma^t r_t + \sum_{t\geqslant 0} \left[\gamma^{t+1} V^{\pi^{\text{old}}}(s_{t+1}) - \gamma^t V^{\pi^{\text{old}}}(s_t) \right] \right] =$$

$$\{\text{regroup terms}\} = \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t \left(r_t + \gamma V^{\pi^{\text{old}}}(s_{t+1}) - V^{\pi^{\text{old}}}(s_t) \right) =$$

$$\{\text{trick } \mathbb{E}_x f(x) = \mathbb{E}_x \mathbb{E}_x f(x)\} = \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t \left(r_t + \gamma \mathbb{E}_{s_{t+1}} V^{\pi^{\text{old}}}(s_{t+1}) - V^{\pi^{\text{old}}}(s_t) \right) =$$

$$\{\text{definition of Q-function}\} = \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t \left(Q^{\pi^{\text{old}}}(s_t, a_t) - V^{\pi^{\text{old}}}(s_t) \right)$$

Relative Performance Identity

$$
\begin{aligned}
V^{\pi_\theta}(s) - V^{\pi^{\text{old}}}(s) &= \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t r_t - V^{\pi^{\text{old}}}(s) = \\
&= \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \left[\sum_{t\geqslant 0} \gamma^t r_t - V^{\pi^{\text{old}}}(s_0) \right] = \\
\{\text{telescoping sums (1)}\} &= \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \left[\sum_{t\geqslant 0} \gamma^t r_t + \sum_{t\geqslant 0} \left[\gamma^{t+1} V^{\pi^{\text{old}}}(s_{t+1}) - \gamma^t V^{\pi^{\text{old}}}(s_t) \right] \right] = \\
\{\text{regroup terms}\} &= \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t \left(r_t + \gamma V^{\pi^{\text{old}}}(s_{t+1}) - V^{\pi^{\text{old}}}(s_t) \right) = \\
\{\text{trick } \mathbb{E}_x f(x) = \mathbb{E}_x \mathbb{E}_x f(x)\} &= \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t \left(r_t + \gamma \mathbb{E}_{s_{t+1}} V^{\pi^{\text{old}}}(s_{t+1}) - V^{\pi^{\text{old}}}(s_t) \right) = \\
\{\text{definition of Q-function}\} &= \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t \left(Q^{\pi^{\text{old}}}(s_t, a_t) - V^{\pi^{\text{old}}}(s_t) \right) \\
\{\text{definition of advantage function}\} &= \mathbb{E}_{\mathcal{T}\sim\pi_\theta|s_0=s} \sum_{t\geqslant 0} \gamma^t A^{\pi^{\text{old}}}(s_t, a_t)
\end{aligned}
$$

Relative Performance Identity

We can switch our optimization function to:

$$J(\pi_\theta) - J(\pi^{\text{old}}) = \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi_\theta}(s)} \mathbb{E}_{a\sim\pi_\theta(a|s)}}_{\text{collect data with } \pi_\theta \text{ (wrong! we don't have it!)}} \overbrace{A^{\pi^{\text{old}}}(s,a)}^{\text{old critic! (good: can train it!)}}$$

Relative Performance Identity

We can switch our optimization function to:

$$J(\pi_\theta) - J(\pi^{\text{old}}) = \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi_\theta}(s)} \mathbb{E}_{a\sim\pi_\theta(a|s)}}_{\substack{\text{collect data with } \pi_\theta \\ \text{(wrong! we don't have it!)}}} \overbrace{A^{\pi^{\text{old}}}(s,a)}^{\substack{\text{old critic!} \\ \text{(good: can train it!)}}}$$

✓ can solve problem with $\mathbb{E}_{a\sim\pi_\theta(a|s)}$ using **importance sampling**;

Relative Performance Identity

We can switch our optimization function to:

$$J(\pi_\theta) - J(\pi^{\text{old}}) = \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi_\theta}(s)} \mathbb{E}_{a\sim\pi_\theta(a|s)}}_{\text{collect data with } \pi_\theta \text{ (wrong! we don't have it!)}} \overbrace{A^{\pi^{\text{old}}}(s,a)}^{\text{old critic! (good: can train it!)}}$$

✓ can solve problem with $\mathbb{E}_{a\sim\pi_\theta(a|s)}$ using **importance sampling**;

✗ can't solve problem with $\mathbb{E}_{s\sim d_{\pi_\theta}(s)}$!

Surrogate objective

Consider the following *local approximation* (**«surrogate objective»**):

$$J(\pi_\theta) - J(\pi^{\text{old}}) \approx L_{\pi^{\text{old}}}(\theta) :=$$

Surrogate objective

Consider the following *local approximation* (**«surrogate objective»**):

$$J(\pi_\theta) - J(\pi^{\text{old}}) \approx L_{\pi^{\text{old}}}(\theta) := \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)} \mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}}_{\text{data generated by } \pi^{\text{old}}} \overbrace{\frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}}^{\text{importance sampling correction}} \underbrace{A^{\pi^{\text{old}}}(s,a)}_{\text{do not require fresh critic}}$$

Surrogate objective

Consider the following *local approximation* (**«surrogate objective»**):

$$J(\pi_\theta) - J(\pi^{\text{old}}) \approx L_{\pi^{\text{old}}}(\theta) := \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)} \mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}}_{\text{data generated by } \pi^{\text{old}}} \overbrace{\frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}}^{\text{importance sampling correction}} \underbrace{A^{\pi^{\text{old}}}(s,a)}_{\text{do not require fresh critic}}$$

- we can work with it;

Surrogate objective

Consider the following *local approximation* (**«surrogate objective»**):

$$J(\pi_\theta) - J(\pi^{\text{old}}) \approx L_{\pi^{\text{old}}}(\theta) := \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)} \mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}}_{\text{data generated by } \pi^{\text{old}}} \overbrace{\frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}}^{\text{importance sampling correction}} \underbrace{A^{\pi^{\text{old}}}(s,a)}_{\text{do not require fresh critic}}$$

- we can work with it;
- directs to policy improvement of π^{old}:
  - optimizing θ with fixed π^{old} will learn $\operatorname{argmax}_a A^{\pi^{\text{old}}}(s,a)$

Surrogate objective

Consider the following *local approximation* (**«surrogate objective»**):

$$J(\pi_\theta) - J(\pi^{\text{old}}) \approx L_{\pi^{\text{old}}}(\theta) := \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)} \mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}}_{\text{data generated by } \pi^{\text{old}}} \overbrace{\frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}}^{\text{importance sampling correction}} \underbrace{A^{\pi^{\text{old}}}(s,a)}_{\text{do not require fresh critic}}$$

- we can work with it;
- directs to policy improvement of π^{old}:
  - optimizing θ with fixed π^{old} will learn $\operatorname{argmax}_a A^{\pi^{\text{old}}}(s,a)$
  - optimizing θ with fixed *data* (fixed $\Psi(s,a) \approx A^{\pi^{\text{old}}}(s,a)$) will learn $\pi_\theta(a \mid s) = 1$ if $\Psi(s,a) > 0$, $\pi_\theta(a \mid s) = 0$ otherwise.

Theoretic Error Bound

$$J(\pi_\theta) - J(\pi^{\text{old}}) \approx L_{\pi^{\text{old}}}(\theta)$$

We want to optimize some *another* objective. **How wrong are we?**

Theoretic Error Bound

$$J(\pi_\theta) - J(\pi^{\text{old}}) \approx L_{\pi^{\text{old}}}(\theta)$$

We want to optimize some *another* objective. **How wrong are we?**

Theorem

$$|J(\pi_\theta) - J(\pi^{\text{old}}) - L_{\pi^{\text{old}}}(\theta)| \leqslant C\,\mathrm{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta),$$

Theoretic Error Bound

$$J(\pi_\theta) - J(\pi^{\text{old}}) \approx L_{\pi^{\text{old}}}(\theta)$$

We want to optimize some *another* objective. **How wrong are we?**

Theorem

$$|J(\pi_\theta) - J(\pi^{\text{old}}) - L_{\pi^{\text{old}}}(\theta)| \leqslant C\,\mathrm{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta),$$

where:

$$\mathrm{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta) := \max_s \mathrm{KL}(\pi^{\text{old}}(\cdot \mid s) \parallel \pi_\theta(\cdot \mid s)))$$

Theoretic Error Bound

$$J(\pi_\theta) - J(\pi^{\text{old}}) \approx L_{\pi^{\text{old}}}(\theta)$$

We want to optimize some *another* objective. **How wrong are we?**

Theorem

$$|J(\pi_\theta) - J(\pi^{\text{old}}) - L_{\pi^{\text{old}}}(\theta)| \leqslant C\,\mathrm{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta),$$

where:

$$\mathrm{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta) := \max_s \mathrm{KL}(\pi^{\text{old}}(\cdot \mid s) \parallel \pi_\theta(\cdot \mid s)))$$

$$C := \frac{4\gamma \max_{s,a} |A^{\pi^{\text{old}}}(s,a)|}{(1-\gamma)^2}$$

Minorization-maximization algorithm

We discovered a **variational lower bound** for our objective:

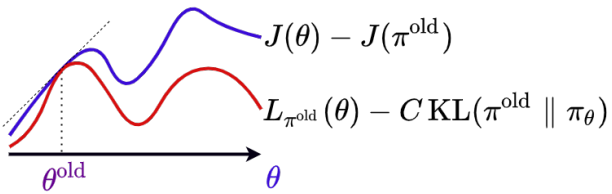
$$J(\pi_\theta) - J(\pi^{\text{old}}) \geqslant L_{\pi^{\text{old}}}(\theta) - C\,\mathrm{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta)$$

Minorization-maximization algorithm

We discovered a **variational lower bound** for our objective:

$$J(\pi_\theta) - J(\pi^{\text{old}}) \geqslant L_{\pi^{\text{old}}}(\theta) - C\,\mathrm{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta)$$

- **Minorization**: construct a new lower bound; in our case simply use $\pi^{\text{old}} \leftarrow \pi_\theta$.

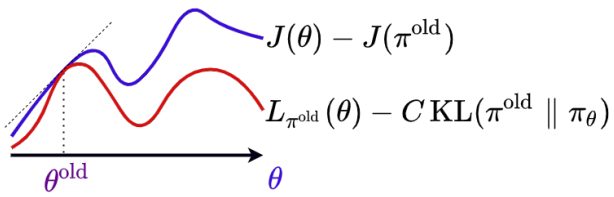


Minorization-maximization algorithm

We discovered a **variational lower bound** for our objective:

$$J(\pi_\theta) - J(\pi^{\text{old}}) \geqslant L_{\pi^{\text{old}}}(\theta) - C\,\mathrm{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta)$$

- **Minorization**: construct a new lower bound; in our case simply use $\pi^{\text{old}} \leftarrow \pi_\theta$.
- **Maximization**: optimize lower bound (as long as you want).

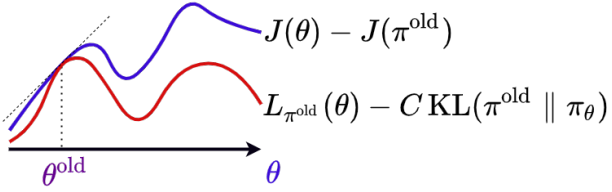


Minorization-maximization algorithm

We discovered a **variational lower bound** for our objective:

$$J(\pi_\theta) - J(\pi^{\text{old}}) \geqslant L_{\pi^{\text{old}}}(\theta) - C\,\text{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta)$$

- **Minorization**: construct a new lower bound; in our case simply use $\pi^{\text{old}} \leftarrow \pi_\theta$.
- **Maximization**: optimize lower bound (as long as you want).



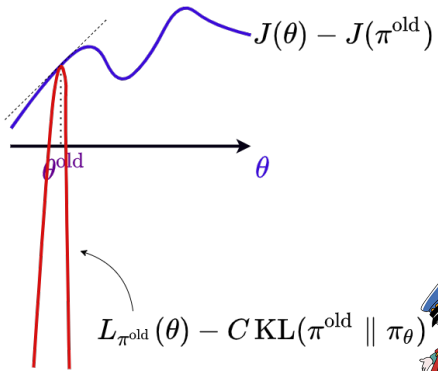
✓ guarantees monotonic improvement!

Minorization-maximization algorithm

We discovered a **variational lower bound** for our objective:

$$J(\pi_\theta) - J(\pi^{\text{old}}) \geqslant L_{\pi^{\text{old}}}(\theta) - C\,\mathrm{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta)$$

- **Minorization**: construct a new lower bound; in our case simply use $\pi^{\text{old}} \leftarrow \pi_\theta$.
- **Maximization**: optimize lower bound (as long as you want).



✗ impractical: real C is too huge!

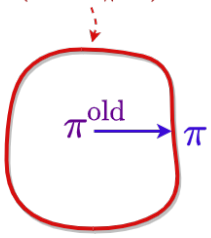
Trust-Region optimization

Optimize $L_{\pi^{\text{old}}}(\theta)$, but guarantee that
π_θ does not change too much from π^{old}!

Trust-Region optimization

Optimize $L_{\pi^{\text{old}}}(\theta)$, but guarantee that
π_θ does not change too much from π^{old}!

$$\mathrm{KL}(\pi^{\text{old}} \,\|\, \pi) \leq \delta$$

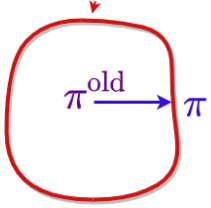


$$\begin{cases} L_{\pi^{\text{old}}}(\theta) \to \max\limits_{\theta} \\ \mathbb{E}_{s \sim d_{\pi^{\text{old}}}(s)} \mathrm{KL}(\pi^{\text{old}}(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s))) \leqslant \delta \end{cases}$$

Trust-Region optimization

Optimize $L_{\pi^{\text{old}}}(\theta)$, but guarantee that
π_θ does not change too much from π^{old}!

$$\mathrm{KL}(\pi^{\text{old}} \,\|\, \pi) \leq \delta$$



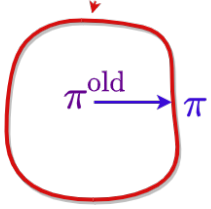
$$\begin{cases} L_{\pi^{\text{old}}}(\theta) \to \max\limits_{\theta} \\ \mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)} \mathrm{KL}(\pi^{\text{old}}(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s))) \leqslant \delta \end{cases}$$

- $L_{\pi^{\text{old}}}(\theta)$ is the **model**;
- condition is **trust region** to this model;

Trust-Region optimization

Optimize $L_{\pi^{\text{old}}}(\theta)$, but guarantee that π_θ does not change too much from π^{old}!

$$\mathrm{KL}(\pi^{\text{old}} \,\|\, \pi) \leq \delta$$



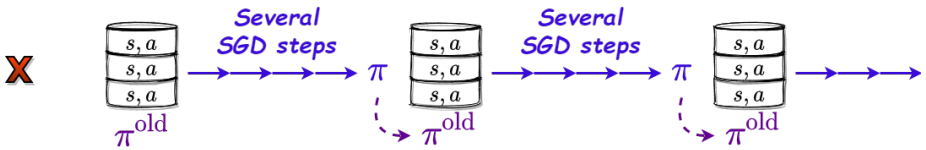
$$\begin{cases} L_{\pi^{\text{old}}}(\theta) \to \max\limits_{\theta} \\ \mathbb{E}_{s \sim d_{\pi^{\text{old}}}(s)} \mathrm{KL}(\pi^{\text{old}}(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s))) \leqslant \delta \end{cases}$$

- $L_{\pi^{\text{old}}}(\theta)$ is the **model**;
- condition is **trust region** to this model;

Algorithm **TRPO** attempts to solve this conditioned task approximately. *Complicated implementation*.

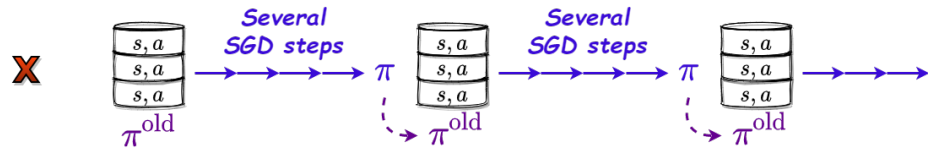
Proximal Policy Optimization (PPO): Pipeline

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}\frac{\pi_\theta(a\mid s)}{\pi^{\text{old}}(a\mid s)}A^{\pi^{\text{old}}}(s,a) - C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$



Proximal Policy Optimization (PPO): Pipeline

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}\frac{\pi_\theta(a\mid s)}{\pi^{\text{old}}(a\mid s)}A^{\pi^{\text{old}}}(s,a) - C\,\mathrm{KL}(\pi^{\text{old}}\parallel\pi_\theta)\to\max_\theta$$



Clipping Objective

$$L_{\pi^{\text{old}}}(\theta) - C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

Default surrogate function:

$$\rho(\theta) := \frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}$$

$$L_{\pi^{\text{old}}}(\theta) := \mathbb{E}_{s,a}\rho(\theta)A^{\pi^{\text{old}}}(s,a)$$

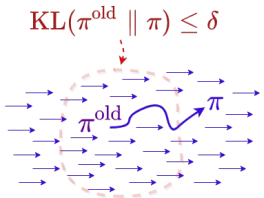
Clipping Objective

$$L_{\pi^{\text{old}}}(\theta) - C\,\mathrm{KL}(\pi^{\text{old}} \,\|\, \pi_\theta) \to \max_\theta$$

Default surrogate function:

$$\rho(\theta) := \frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}$$

$$L_{\pi^{\text{old}}}(\theta) := \mathbb{E}_{s,a}\rho(\theta)A^{\pi^{\text{old}}}(s,a)$$



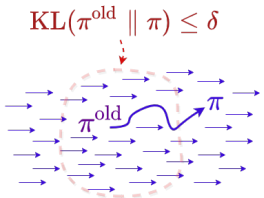
Clipping Objective

$$L_{\pi^{\text{old}}}(\theta) - C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

Default surrogate function:

$$\rho(\theta) := \frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}$$

$$L_{\pi^{\text{old}}}(\theta) := \mathbb{E}_{s,a}\rho(\theta)A^{\pi^{\text{old}}}(s,a)$$



Clipped surrogate function:

$$\rho^{\text{clip}}(\theta) := \mathrm{clip}(\rho(\theta), 1-\epsilon, 1+\epsilon)$$

$$L^{\text{clip}}_{\pi^{\text{old}}}(\theta) := \mathbb{E}_{s,a}\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)$$

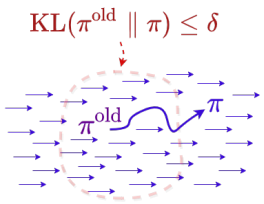
Clipping Objective

$$L_{\pi^{\text{old}}}(\theta) - C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

Default surrogate function:

$$\rho(\theta) := \frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}$$

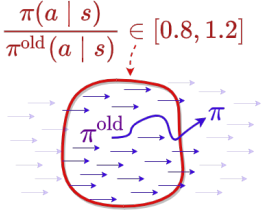
$$L_{\pi^{\text{old}}}(\theta) := \mathbb{E}_{s,a}\rho(\theta)A^{\pi^{\text{old}}}(s,a)$$



Clipped surrogate function:

$$\rho^{\text{clip}}(\theta) := \mathrm{clip}(\rho(\theta), 1-\epsilon, 1+\epsilon)$$

$$L^{\text{clip}}_{\pi^{\text{old}}}(\theta) := \mathbb{E}_{s,a}\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)$$



Recalling lower bound intuition

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}\min(\overbrace{\rho(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{original term}},\ \overbrace{\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)}^{\substack{\text{term with clipped}\\ \text{importance sampling weight}}}) - \overbrace{C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

Recalling lower bound intuition

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}\min(\overbrace{\rho(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{original term}},\ \overbrace{\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{term with clipped importance sampling weight}}) - \overbrace{C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$			

Recalling lower bound intuition

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}\min(\overbrace{\rho(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{original term}},\ \overbrace{\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)}^{\substack{\text{term with clipped}\\ \text{importance sampling weight}}}) - \overbrace{C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$	$\pi_\theta(a \mid s) \Uparrow$		

Recalling lower bound intuition

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}\min(\overbrace{\rho(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{original term}},\ \overbrace{\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{term with clipped importance sampling weight}}) - \overbrace{C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$	$\pi_\theta(a \mid s) \Uparrow$	$\rho(\theta) > 1.2$	

Recalling lower bound intuition

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}\min(\overbrace{\rho(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{original term}},\ \overbrace{\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{term with clipped importance sampling weight}}) - \overbrace{C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$	$\pi_\theta(a \mid s) \Uparrow$	$\rho(\theta) > 1.2$	0

Recalling lower bound intuition

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}\min(\overbrace{\rho(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{original term}},\ \overbrace{\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{term with clipped importance sampling weight}}) - \overbrace{C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$	$\pi_\theta(a \mid s) \Uparrow$	$\rho(\theta) > 1.2$ $\rho(\theta) < 0.8$	0

Recalling lower bound intuition

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}\min(\overbrace{\rho(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{original term}},\ \overbrace{\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{term with clipped importance sampling weight}}) - \overbrace{C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$	$\pi_\theta(a \mid s) \Uparrow$	$\rho(\theta) > 1.2$ $\rho(\theta) < 0.8$	0 same

Recalling lower bound intuition

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}\min(\overbrace{\rho(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{original term}},\ \overbrace{\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)}^{\substack{\text{term with clipped}\\ \text{importance sampling weight}}}) - \overbrace{C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$	$\pi_\theta(a \mid s) \Uparrow$	$\rho(\theta) > 1.2$	0
		$\rho(\theta) < 0.8$	same
$A^{\pi^{\text{old}}}(s,a) < 0$	$\pi_\theta(a \mid s) \Downarrow$	$\rho(\theta) > 1.2$	
		$\rho(\theta) < 0.8$	

Recalling lower bound intuition

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}\min(\overbrace{\rho(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{original term}},\ \overbrace{\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)}^{\substack{\text{term with clipped}\\ \text{importance sampling weight}}}) - \overbrace{C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

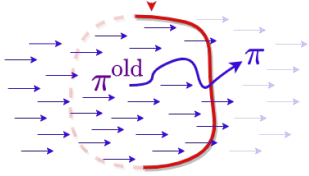
Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$	$\pi_\theta(a \mid s) \Uparrow$	$\rho(\theta) > 1.2$	0
		$\rho(\theta) < 0.8$	same
$A^{\pi^{\text{old}}}(s,a) < 0$	$\pi_\theta(a \mid s) \Downarrow$	$\rho(\theta) > 1.2$	same
		$\rho(\theta) < 0.8$	0

Recalling lower bound intuition

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}\min(\overbrace{\rho(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{original term}},\ \overbrace{\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{term with clipped importance sampling weight}}) - \overbrace{C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

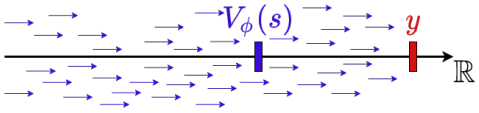
Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$	$\pi_\theta(a \mid s) \Uparrow$	$\rho(\theta) > 1.2$	0
		$\rho(\theta) < 0.8$	same
$A^{\pi^{\text{old}}}(s,a) < 0$	$\pi_\theta(a \mid s) \Downarrow$	$\rho(\theta) > 1.2$	same
		$\rho(\theta) < 0.8$	0

$$\frac{\pi(a \mid s)}{\pi^{\text{old}}(a \mid s)} \in [0.8, 1.2]$$



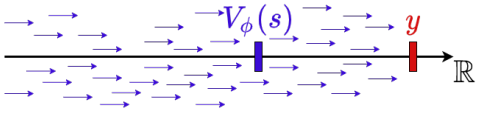
Clipped Critic Loss

$\text{Loss}(\phi) := (y - V(\phi))^2 =$



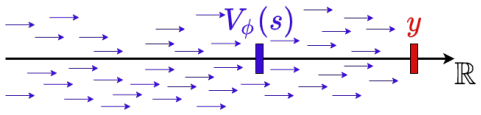
Clipped Critic Loss

$$\text{Loss}(\phi) := (y - V(\phi))^2 =$$
$$= (y - V^{\text{old}} + V^{\text{old}} - V(\phi))^2$$

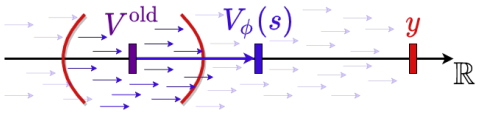


Clipped Critic Loss

$$\text{Loss}(\phi) := (y - V(\phi))^2 =$$
$$= (y - V^{\text{old}} + V^{\text{old}} - V(\phi))^2$$

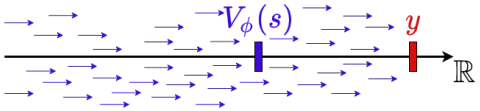


$$\text{Loss}^{\text{clip}}(\phi) := (y - V^{\text{old}} + \text{clip}(V^{\text{old}} - V(\phi), -\hat{\epsilon}, \hat{\epsilon}))^2$$

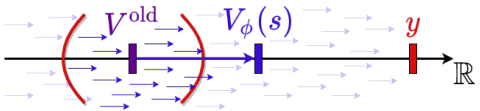


Clipped Critic Loss

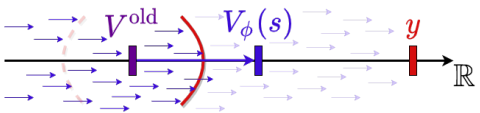
$$\text{Loss}(\phi) := (y - V(\phi))^2 =$$
$$= (y - V^{\text{old}} + V^{\text{old}} - V(\phi))^2$$



$$\text{Loss}^{\text{clip}}(\phi) := (y - V^{\text{old}} + \text{clip}(V^{\text{old}} - V(\phi), -\hat{\epsilon}, \hat{\epsilon}))^2$$



$$\max(\text{Loss}(\phi), \text{Loss}^{\text{clip}}(\phi))$$



Proximal Policy Optimization: implementation matters

Key elements:

- ✓ Clipped policy loss
- ✓ Clipped critic loss
- ✓ GAE

Pipeline details:

- ! Advantage normalization in mini-batches
- No KL regularization
- Entropy loss

Other hacks:

- ! Reward normalization[1] and clipping
- Observations normalization and clipping[2]
- Orthogonal initialization of layers
- ϵ (clipping parameter) annealing

Standard tricks:

- Adam, learning rate annealing
- Tanh activation functions
- ! Gradient clipping

[1] divided by running std of collected cumulative rewards
[2] can be critical in continuous control

Full Pipeline: pt.I

Proximal Policy Optimization (PPO)

Initialize $\pi(a \mid s, \theta), V_\phi(s)$;

Full Pipeline: pt.I

Proximal Policy Optimization (PPO)

Initialize $\pi(a \mid s, \theta), V_\phi(s)$;

for $k = 0, 1, 2 \ldots$

- collect several rollouts $s_0, a_0, r_0, s_1, \mathrm{done}_1, a_1 \ldots s_N, \mathrm{done}_N$ using $\pi(a \mid s, \theta)$;
  store probabilities of selected actions as $\pi^{\mathrm{old}}(a_t \mid s_t) := \pi(a_t \mid s_t, \theta)$
  store critic output as $V^{\mathrm{old}}(s_t) := V_\phi(s_t)$

Full Pipeline: pt.I

Proximal Policy Optimization (PPO)

Initialize $\pi(a \mid s, \theta), V_\phi(s)$;

for $k = 0, 1, 2 \ldots$

- collect several rollouts $s_0, a_0, r_0, s_1, \mathrm{done}_1, a_1 \ldots s_N, \mathrm{done}_N$ using $\pi(a \mid s, \theta)$;
  store probabilities of selected actions as $\pi^{\mathrm{old}}(a_t \mid s_t) := \pi(a_t \mid s_t, \theta)$
  store critic output as $V^{\mathrm{old}}(s_t) := V_\phi(s_t)$
- compute 1-step errors: $\Psi_{(1)}(s_t, a_t) := r_t + \gamma(1 - \mathrm{done}_{t+1}) V_\phi(s_{t+1}) - V_\phi(s_t)$

Full Pipeline: pt.I

Proximal Policy Optimization (PPO)

Initialize $\pi(a \mid s, \theta), V_\phi(s)$;

for $k = 0, 1, 2 \ldots$

- collect several rollouts $s_0, a_0, r_0, s_1, \mathrm{done}_1, a_1 \ldots s_N, \mathrm{done}_N$ using $\pi(a \mid s, \theta)$;
  store probabilities of selected actions as $\pi^{\mathrm{old}}(a_t \mid s_t) := \pi(a_t \mid s_t, \theta)$
  store critic output as $V^{\mathrm{old}}(s_t) := V_\phi(s_t)$
- compute 1-step errors: $\Psi_{(1)}(s_t, a_t) := r_t + \gamma(1 - \mathrm{done}_{t+1}) V_\phi(s_{t+1}) - V_\phi(s_t)$
- compute GAE advantage estimations: $\Psi^{\mathrm{GAE}}(s_{N-1}, a_{N-1}) := \Psi_{(1)}(s_{N-1}, a_{N-1})$
- for t from $N-2$ to 0:
  - $\Psi^{\mathrm{GAE}}(s_t, a_t) :=$

Full Pipeline: pt.I

Proximal Policy Optimization (PPO)

Initialize $\pi(a \mid s, \theta), V_\phi(s)$;

for $k = 0, 1, 2 \ldots$

- collect several rollouts $s_0, a_0, r_0, s_1, \mathrm{done}_1, a_1 \ldots s_N, \mathrm{done}_N$ using $\pi(a \mid s, \theta)$;
  store probabilities of selected actions as $\pi^{\mathrm{old}}(a_t \mid s_t) := \pi(a_t \mid s_t, \theta)$
  store critic output as $V^{\mathrm{old}}(s_t) := V_\phi(s_t)$
- compute 1-step errors: $\Psi_{(1)}(s_t, a_t) := r_t + \gamma(1 - \mathrm{done}_{t+1}) V_\phi(s_{t+1}) - V_\phi(s_t)$
- compute GAE advantage estimations: $\Psi^{\mathrm{GAE}}(s_{N-1}, a_{N-1}) := \Psi_{(1)}(s_{N-1}, a_{N-1})$
- for t from $N-2$ to 0:
  - $\Psi^{\mathrm{GAE}}(s_t, a_t) := \Psi_{(1)}(s_t, a_t) + \lambda\gamma(1 - \mathrm{done}_{t+1}) \Psi^{\mathrm{GAE}}(s_{t+1}, a_{t+1})$

Full Pipeline: pt.I

Proximal Policy Optimization (PPO)

Initialize $\pi(a \mid s, \theta), V_\phi(s)$;

for $k = 0, 1, 2 \ldots$

- collect several rollouts $s_0, a_0, r_0, s_1, \mathrm{done}_1, a_1 \ldots s_N, \mathrm{done}_N$ using $\pi(a \mid s, \theta)$;
  store probabilities of selected actions as $\pi^{\mathrm{old}}(a_t \mid s_t) := \pi(a_t \mid s_t, \theta)$
  store critic output as $V^{\mathrm{old}}(s_t) := V_\phi(s_t)$
- compute 1-step errors: $\Psi_{(1)}(s_t, a_t) := r_t + \gamma(1 - \mathrm{done}_{t+1}) V_\phi(s_{t+1}) - V_\phi(s_t)$
- compute GAE advantage estimations: $\Psi^{\mathrm{GAE}}(s_{N-1}, a_{N-1}) := \Psi_{(1)}(s_{N-1}, a_{N-1})$
- for t from $N-2$ to 0:
  - $\Psi^{\mathrm{GAE}}(s_t, a_t) := \Psi_{(1)}(s_t, a_t) + \lambda\gamma(1 - \mathrm{done}_{t+1})\Psi^{\mathrm{GAE}}(s_{t+1}, a_{t+1})$
- compute critic targets: $y(s_t) := \Psi^{\mathrm{GAE}}(s_t, a_t) + V_\phi(s_t)$

Full Pipeline: pt.I

Proximal Policy Optimization (PPO)

Initialize $\pi(a \mid s, \theta), V_\phi(s)$;

for $k = 0, 1, 2 \ldots$

- collect several rollouts $s_0, a_0, r_0, s_1, \mathrm{done}_1, a_1 \ldots s_N, \mathrm{done}_N$ using $\pi(a \mid s, \theta)$;
  store probabilities of selected actions as $\pi^{\mathrm{old}}(a_t \mid s_t) := \pi(a_t \mid s_t, \theta)$
  store critic output as $V^{\mathrm{old}}(s_t) := V_\phi(s_t)$
- compute 1-step errors: $\Psi_{(1)}(s_t, a_t) := r_t + \gamma(1 - \mathrm{done}_{t+1}) V_\phi(s_{t+1}) - V_\phi(s_t)$
- compute GAE advantage estimations: $\Psi^{\mathrm{GAE}}(s_{N-1}, a_{N-1}) := \Psi_{(1)}(s_{N-1}, a_{N-1})$
- for t from $N-2$ to 0:
  - $\Psi^{\mathrm{GAE}}(s_t, a_t) := \Psi_{(1)}(s_t, a_t) + \lambda\gamma(1 - \mathrm{done}_{t+1})\Psi^{\mathrm{GAE}}(s_{t+1}, a_{t+1})$
- compute critic targets: $y(s_t) := \Psi^{\mathrm{GAE}}(s_t, a_t) + V_\phi(s_t)$
- construct dataset of $(s_t, a_t, \Psi^{\mathrm{GAE}}(s_t, a_t), y(s_t), \pi^{\mathrm{old}}(a_t \mid s_t), V^{\mathrm{old}}(s_t))$

Full Pipeline: pt.II

Proximal Policy Optimization (PPO) -- cont.

- go through dataset $\mathrm{n\_epochs}$ times, sampling mini-batches of size B; for each mini-batch:

Full Pipeline: pt.II

Proximal Policy Optimization (PPO) -- cont.

- go through dataset $\mathrm{n\_epochs}$ times, sampling mini-batches of size B; for each mini-batch:
  - normalize $\Psi^{\mathrm{GAE}}(s, a)$ in the batch by subtracting mean and dividing by std

Full Pipeline: pt.II

Proximal Policy Optimization (PPO) -- cont.

- go through dataset $\mathrm{n\_epochs}$ times, sampling mini-batches of size B; for each mini-batch:
  - normalize $\Psi^{\mathrm{GAE}}(s, a)$ in the batch by subtracting mean and dividing by std
  - compute importance sampling weights:

$$\rho(s, a, \theta) := \frac{\pi(a \mid s, \theta)}{\pi^{\mathrm{old}}(a \mid s)}, \qquad \rho^{\mathrm{clip}}(s, a, \theta) = \mathrm{clip}(\rho(s, a, \theta), 1 - \epsilon, 1 + \epsilon)$$

Full Pipeline: pt.II

Proximal Policy Optimization (PPO) -- cont.

- go through dataset $\mathrm{n\_epochs}$ times, sampling mini-batches of size B; for each mini-batch:
  - normalize $\Psi^{\mathrm{GAE}}(s,a)$ in the batch by subtracting mean and dividing by std
  - compute importance sampling weights:

$$\rho(s,a,\theta) := \frac{\pi(a \mid s,\theta)}{\pi^{\mathrm{old}}(a \mid s)}, \qquad \rho^{\mathrm{clip}}(s,a,\theta) = \mathrm{clip}(\rho(s,a,\theta), 1-\epsilon, 1+\epsilon)$$

  - update actor:

$$L_1(s,a,\theta) := \rho(s,a,\theta)\Psi^{\mathrm{GAE}}(s,a), \qquad L_2(s,a,\theta) := \rho^{\mathrm{clip}}(s,a,\theta)\Psi^{\mathrm{GAE}}(s,a)$$

$$\theta \leftarrow \theta + \alpha \nabla_\theta \frac{1}{B} \sum_{s,a} \min(L_1(s,a,\theta), L_2(s,a,\theta))$$

Full Pipeline: pt.II

Proximal Policy Optimization (PPO) -- cont.

- go through dataset $\mathrm{n\_epochs}$ times, sampling mini-batches of size B; for each mini-batch:
  - normalize $\Psi^{\mathrm{GAE}}(s,a)$ in the batch by subtracting mean and dividing by std
  - compute importance sampling weights:

$$\rho(s,a,\theta) := \frac{\pi(a \mid s,\theta)}{\pi^{\mathrm{old}}(a \mid s)}, \qquad \rho^{\mathrm{clip}}(s,a,\theta) = \mathrm{clip}(\rho(s,a,\theta), 1-\epsilon, 1+\epsilon)$$

  - update actor:

$$L_1(s,a,\theta) := \rho(s,a,\theta)\Psi^{\mathrm{GAE}}(s,a), \qquad L_2(s,a,\theta) := \rho^{\mathrm{clip}}(s,a,\theta)\Psi^{\mathrm{GAE}}(s,a)$$

$$\theta \leftarrow \theta + \alpha \nabla_\theta \frac{1}{B} \sum_{s,a} \min(L_1(s,a,\theta), L_2(s,a,\theta))$$

  - update critic:

$$\mathrm{Loss}_1(s,\phi) := (y(s) - V_\phi(s))^2$$

$$\mathrm{Loss}_2(s,\phi) := \left(y(s) - V^{\mathrm{old}}(s) - \mathrm{clip}(V_\phi(s) - V^{\mathrm{old}}(s), \hat{\epsilon}, -\hat{\epsilon})\right)^2$$

$$\phi \leftarrow \phi - \alpha \nabla_\phi \frac{1}{B} \sum_{s} \max(\mathrm{Loss}_1(s,\phi), \mathrm{Loss}_2(s,\phi))$$

Go-to Model Free Algorithms

	Discrete Action Space	*Continuous Action Space*
On-policy	**PPO**	**PPO**
Off-policy	**DQN+**	**TD3 / SAC** *(next time)*

Go-to Model Free Algorithms

	Discrete Action Space	*Continuous Action Space*
On-policy	**PPO**	**PPO**
Off-policy	**DQN+**	**TD3 / SAC** *(next time)*

Hometask 5:

Continuous control; you can choose PPO, TD3 or SAC.

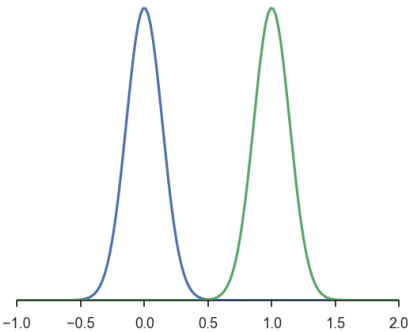
Literature

- High-Dimensional Continuous Control Using Generalized Advantage Estimation;
- Trust Region Policy Optimization;
- Proximal Policy Optimization Algorithms;
- Implementation Matters in Deep Policy Gradients: A Case Study on PPO and TRPO;
- OpenAI Five;

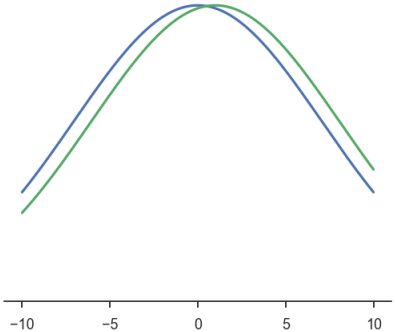
Appendix: Trust-Region Policy Optimization (TRPO)

Parameter space vs distribution space

Two gaussians: $\mathcal{N}(0, 0.2), \mathcal{N}(1, 0.2)$

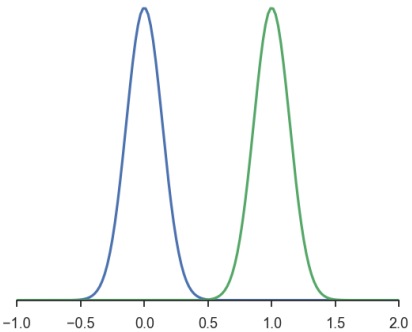


Two gaussians: $\mathcal{N}(0, 10), \mathcal{N}(1, 10)$

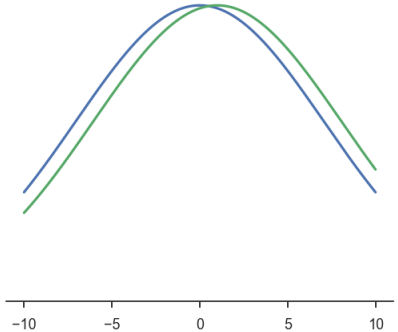


Parameter space vs distribution space

Two gaussians: $\mathcal{N}(0, 0.2), \mathcal{N}(1, 0.2)$



Two gaussians: $\mathcal{N}(0, 10), \mathcal{N}(1, 10)$



Euclidean distance: $\sqrt{(\mu_1 - \mu_2)^2 + (\sigma_1 - \sigma_2)^2}$

Statistical distance: e.g. $\mathrm{KL}(\mathcal{N}(\mu_1, \sigma_1) \parallel \mathcal{N}(\mu_2, \sigma_2))$

Step size issue: riding from the cliff

$$\theta_{k+1} = \theta_k + \alpha \hat{g}_k$$

If step size too large:

- step too far leads to bad policy;
- we start *collecting data* with bad policy;
- $\Longrightarrow$ **performance collapse**

Step size issue: riding from the cliff

$$\theta_{k+1} = \theta_k + \alpha \hat{\mathbf{g}}_k$$

If step size too large:

- step too far leads to bad policy;
- we start *collecting data* with bad policy;
- $\Longrightarrow$ **performance collapse**

We need optimization method that measures distance in **distribution space**, not in **parameter space**!

A view on optimization methods

Consider the following optimization task:

$$G(\theta) := G(q_\theta(x)) \to \min_\theta$$

A view on optimization methods

Consider the following optimization task:

$$G(\theta) := G(q_\theta(x)) \to \min_\theta$$

General scheme of trust region optimization methods

- start with arbitrary θ_0;
- **for** $k = 0, 1, 2, \ldots$:
  - construct a **model**, some local approximation of $G(\theta)$ near θ_k:

$$m_k(\theta) \approx G(\theta)$$

A view on optimization methods

Consider the following optimization task:

$$G(\theta) := G(q_\theta(x)) \to \min_\theta$$

General scheme of trust region optimization methods

- start with arbitrary θ_0;
- **for** $k = 0, 1, 2, \ldots$:
  - construct a **model**, some local approximation of $G(\theta)$ near θ_k:
    $$m_k(\theta) \approx G(\theta)$$
  - select some **trust region** where you assume this model is more or less correct:
    $$r(\theta, \theta_k) \leqslant \delta$$

A view on optimization methods

Consider the following optimization task:

$$G(\theta) := G(q_\theta(x)) \to \min_\theta$$

General scheme of trust region optimization methods

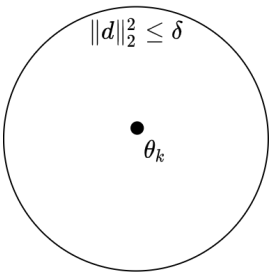
- start with arbitrary θ_0;
- **for** $k = 0, 1, 2, \ldots$:
  - construct a **model**, some local approximation of $G(\theta)$ near θ_k:
  $$m_k(\theta) \approx G(\theta)$$
  - select some **trust region** where you assume this model is more or less correct:
  $$r(\theta, \theta_k) \leqslant \delta$$
  - seek θ_{k+1} as solution of
  $$\begin{cases} m_k(\theta) \to \min\limits_\theta \\ r(\theta, \theta_k) \leqslant \delta \end{cases}$$

Natural gradient descent

$$G(\theta) := G(q_\theta(x)) \to \min_\theta$$

Gradient Descent

$$\begin{cases} G(\theta_k + d) \approx G(\theta_k) + \nabla_\theta G(\theta_k)^T d \to \min\limits_d \\ \|d\|_2 \leqslant \delta \end{cases}$$

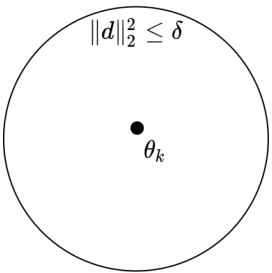


Natural gradient descent

$$G(\theta) := G(q_\theta(x)) \to \min_\theta$$

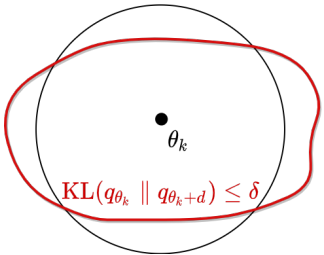
Gradient Descent

$$\begin{cases} G(\theta_k + d) \approx G(\theta_k) + \nabla_\theta G(\theta_k)^T d \to \min\limits_d \\ \|d\|_2 \leqslant \delta \end{cases}$$



Natural Gradient Descent

$$\begin{cases} G(\theta_k + d) \approx G(\theta_k) + \nabla_\theta G(\theta_k)^T d \to \min\limits_d \\ \mathrm{KL}(q_{\theta_k}(x) \parallel q_{\theta_k + d}(x)) \leqslant \delta \end{cases}$$

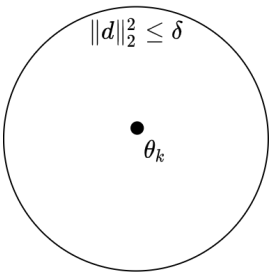


Natural gradient descent

$$G(\theta) := G(q_\theta(x)) \to \min_\theta$$

Gradient Descent

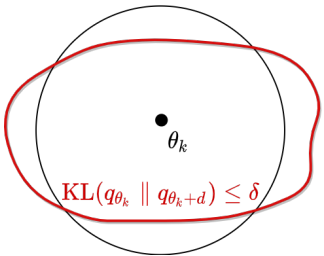
$$\begin{cases} G(\theta_k + d) \approx G(\theta_k) + \nabla_\theta G(\theta_k)^T d \to \min\limits_d \\ \|d\|_2 \leqslant \delta \end{cases}$$



Solution: $d \propto -\nabla_\theta G(\theta_k)$

Natural Gradient Descent

$$\begin{cases} G(\theta_k + d) \approx G(\theta_k) + \nabla_\theta G(\theta_k)^T d \to \min\limits_d \\ \mathrm{KL}(q_{\theta_k}(x) \parallel q_{\theta_k + d}(x)) \leqslant \delta \end{cases}$$

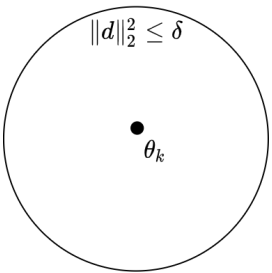


Natural gradient descent

$$G(\theta) := G(q_\theta(x)) \to \min_\theta$$

Gradient Descent

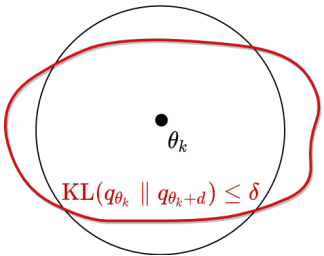
$$\begin{cases} G(\theta_k + d) \approx G(\theta_k) + \nabla_\theta G(\theta_k)^T d \to \min\limits_d \\ \|d\|_2 \leqslant \delta \end{cases}$$



Solution: $d \propto -\nabla_\theta G(\theta_k)$

Natural Gradient Descent

$$\begin{cases} G(\theta_k + d) \approx G(\theta_k) + \nabla_\theta G(\theta_k)^T d \to \min\limits_d \\ \mathrm{KL}(q_{\theta_k}(x) \parallel q_{\theta_k + d}(x)) \leqslant \delta \end{cases}$$



Solution: ?!?

Fisher Matrix

Use Taylor expansion for

$$\mathrm{KL}(q_{\theta_k}(x) \parallel q_{\theta_k+d}(x)) \leqslant \delta$$

Fisher Matrix

Use Taylor expansion for

$$\mathrm{KL}(q_{\theta_k}(x) \parallel q_{\theta_k + d}(x)) \leqslant \delta$$

(!) First order term is actually zero:

$$\nabla_d \, \mathrm{KL}(q_{\theta_k}(x) \parallel q_{\theta_k + d}(x))\big|_{d=0} = 0$$

Fisher Matrix

Use Taylor expansion for

$$\mathrm{KL}(q_{\theta_k}(x) \parallel q_{\theta_k+d}(x)) \leqslant \delta$$

(!) First order term is actually zero:

$$\nabla_d \,\mathrm{KL}(q_{\theta_k}(x) \parallel q_{\theta_k+d}(x))\big|_{d=0} = 0$$

Fisher matrix

Let $F(\theta_k)$ be the **Fisher matrix** of $q_\theta(x)$ at point θ_k:

$$F(\theta_k) := -\mathbb{E}_{x\sim q_\theta(x)} \nabla^2_\theta \log q_\theta(x)$$

Fisher Matrix

Use Taylor expansion for

$$\mathrm{KL}(q_{\theta_k}(x) \parallel q_{\theta_k+d}(x)) \leqslant \delta$$

(!) First order term is actually zero:

$$\nabla_d \,\mathrm{KL}(q_{\theta_k}(x) \parallel q_{\theta_k+d}(x))\big|_{d=0} = 0$$

Fisher matrix

Let $F(\theta_k)$ be the **Fisher matrix** of $q_\theta(x)$ at point θ_k:

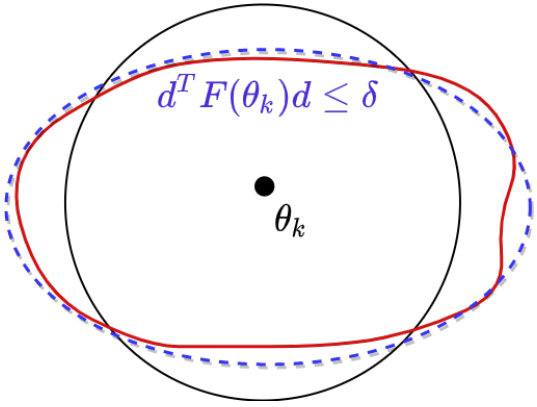
$$F(\theta_k) := -\mathbb{E}_{x\sim q_\theta(x)} \nabla^2_\theta \log q_\theta(x)$$

Then Taylor approximation of second order is:

$$\mathrm{KL}(q_{\theta_k}(x) \parallel q_{\theta_k+d}(x)) \approx \frac{1}{2} d^T F(\theta_k) d$$

Direction of natural gradient

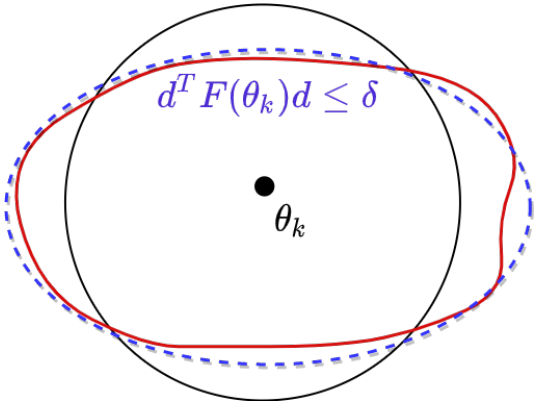
$$\begin{cases} G(\theta_k + d) \approx G(\theta_k) + \nabla_\theta G(\theta_k)^T d \to \min\limits_d \\ d^T F(\theta_k) d \leqslant \delta \end{cases}$$



Direction of natural gradient

$$\begin{cases} G(\theta_k + d) \approx G(\theta_k) + \nabla_\theta G(\theta_k)^T d \to \min\limits_d \\ d^T F(\theta_k) d \leqslant \delta \end{cases}$$

Solution: $d \propto -F^{-1}(\theta_k)\nabla_\theta G(\theta_k)$



Direction of natural gradient

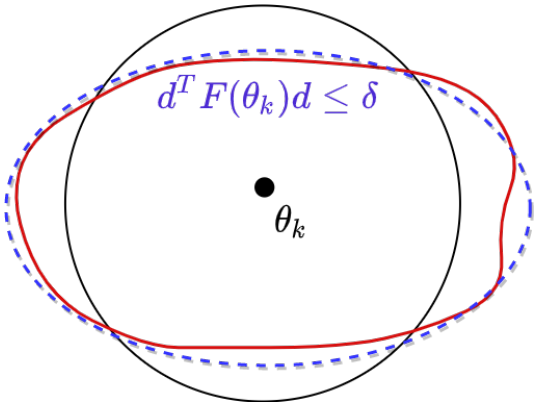
$$\begin{cases} G(\theta_k + d) \approx G(\theta_k) + \nabla_\theta G(\theta_k)^T d \to \min\limits_d \\ d^T F(\theta_k) d \leqslant \delta \end{cases}$$

Solution: $d \propto -F^{-1}(\theta_k)\nabla_\theta G(\theta_k)$

Natural gradient uses the following update:

$$\theta_{k+1} = \theta_k - \alpha F^{-1}(\theta_k)\nabla_\theta G(\theta_k),$$

where α is determined by the choice of δ.



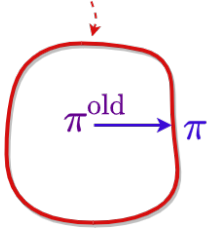
Trust-Region optimization

Optimize $L_{\pi^{\text{old}}}(\theta)$, but guarantee that
π_θ does not change too much from π^{old}!

Trust-Region optimization

Optimize $L_{\pi^{\text{old}}}(\theta)$, but guarantee that
π_θ does not change too much from π^{old}!

$$\mathrm{KL}(\pi^{\text{old}} \,\|\, \pi) \leq \delta$$

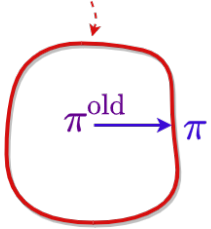


$$\begin{cases} L_{\pi^{\text{old}}}(\theta) \to \max\limits_{\theta} \\ \mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)} \mathrm{KL}(\pi^{\text{old}}(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s))) \leqslant \delta \end{cases}$$

Trust-Region optimization

Optimize $L_{\pi^{\text{old}}}(\theta)$, but guarantee that
π_θ does not change too much from π^{old}!

$$\mathrm{KL}(\pi^{\text{old}} \,\|\, \pi) \leq \delta$$



$$\begin{cases} L_{\pi^{\text{old}}}(\theta) \to \max\limits_{\theta} \\ \mathbb{E}_{s \sim d_{\pi^{\text{old}}}(s)} \mathrm{KL}(\pi^{\text{old}}(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s))) \leqslant \delta \end{cases}$$

- $L_{\pi^{\text{old}}}(\theta)$ is the **model**;
- condition is **trust region** to this model;

Finding approximate solution

Use Taylor expansion for both model and condition to solve this system!

Finding approximate solution

Use Taylor expansion for both model and condition to solve this system!

$$L_{\pi^{\text{old}}}(\theta) \approx g^T d \to \max_d$$

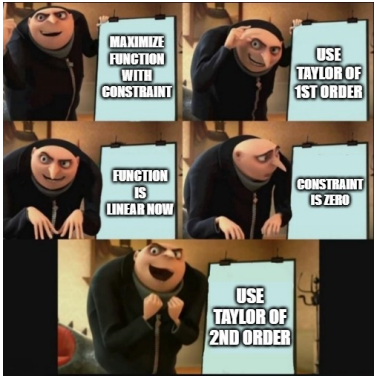
$$d := \theta - \theta^{\text{old}}, \qquad g := \nabla_\theta L_{\pi^{\text{old}}}(\theta)|_{\theta=\theta^{\text{old}}}$$

Finding approximate solution

Use Taylor expansion for both model and condition to solve this system!

$$L_{\pi^{\text{old}}}(\theta) \approx g^T d \rightarrow \max_d$$

$$d := \theta - \theta^{\text{old}}, \qquad g := \nabla_\theta L_{\pi^{\text{old}}}(\theta)\big|_{\theta=\theta^{\text{old}}}$$



Finding approximate solution

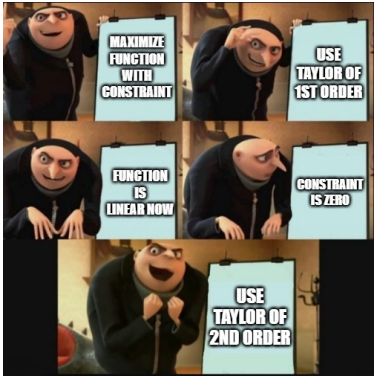
Use Taylor expansion for both model and condition to solve this system!

$$L_{\pi^{\text{old}}}(\theta) \approx g^T d \to \max_d$$

$$d := \theta - \theta^{\text{old}}, \qquad g := \nabla_\theta L_{\pi^{\text{old}}}(\theta)|_{\theta=\theta^{\text{old}}}$$

For KL divergence, first term is actually zero, so use second-order Taylor approximation:

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)} \mathrm{KL}(\pi^{\text{old}}(\cdot \mid s) \parallel \pi_\theta(\cdot \mid s))) \approx \frac{1}{2} d^T H d \leqslant \delta$$



Finding approximate solution

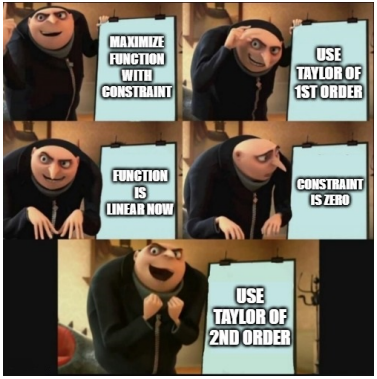
Use Taylor expansion for both model and condition to solve this system!

$$L_{\pi^{\text{old}}}(\theta) \approx g^T d \to \max_d$$

$$d := \theta - \theta^{\text{old}}, \qquad g := \nabla_\theta L_{\pi^{\text{old}}}(\theta)|_{\theta=\theta^{\text{old}}}$$

For KL divergence, first term is actually zero, so use second-order Taylor approximation:

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)} \mathrm{KL}(\pi^{\text{old}}(\cdot \mid s) \parallel \pi_\theta(\cdot \mid s))) \approx \frac{1}{2} d^T H d \leqslant \delta$$



Natural Gradient Ascent

Solution: $d \propto H^{-1} g$

Reality check: can we compute inverse hessian?

$$\theta_{k+1} = \theta_k + \alpha H^{-1} g$$

Danger!

Natural gradient descent is similar to second-order methods.

Reality check: can we compute inverse hessian?

$$\theta_{k+1} = \theta_k + \alpha H^{-1} g$$

Danger!

Natural gradient descent is similar to second-order methods.

Suppose h is the dimension of θ...

- ✗ H is a matrix $h \times h$ (a hessian of KL-divergence);

Reality check: can we compute inverse hessian?

$$\theta_{k+1} = \theta_k + \alpha H^{-1} g$$

Danger!

Natural gradient descent is similar to second-order methods.

Suppose h is the dimension of θ...

- ✗ H is a matrix $h \times h$ (a hessian of KL-divergence);
- ✗ How to inverse it?

Reality check: can we compute inverse hessian?

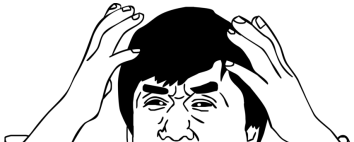
$$\theta_{k+1} = \theta_k + \alpha H^{-1} g$$

Danger!

Natural gradient descent is similar to second-order methods.

Suppose h is the dimension of θ...

- ✗ H is a matrix $h \times h$ (a hessian of KL-divergence);
- ✗ How to inverse it?



Reality check: can we compute inverse hessian?

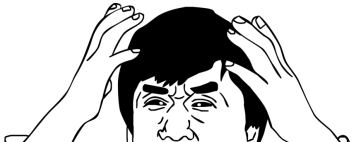
$$\theta_{k+1} = \theta_k + \alpha H^{-1} g$$

Danger!

Natural gradient descent is similar to second-order methods.

Suppose h is the dimension of θ...

- ✗ H is a matrix $h \times h$ (a hessian of KL-divergence);
- ✗ How to inverse it?



Solve the following linear system:

$$Hd = g$$

using **conjugate gradient** method.

Reality check: can we compute inverse hessian?

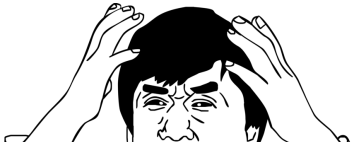
$$\theta_{k+1} = \theta_k + \alpha H^{-1} g$$

Danger!

Natural gradient descent is similar to second-order methods.

Suppose h is the dimension of θ...

- ✗ H is a matrix $h \times h$ (a hessian of KL-divergence);
- ✗ How to inverse it?



Solve the following linear system:

$$Hd = g$$

using **conjugate gradient** method.

1. compute g (as usual in standard gradient descent);

Reality check: can we compute inverse hessian?

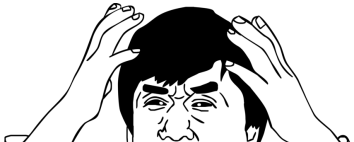
$$\theta_{k+1} = \theta_k + \alpha H^{-1} g$$

Danger!

Natural gradient descent is similar to second-order methods.

Suppose h is the dimension of θ...

- ✗ H is a matrix $h \times h$ (a hessian of KL-divergence);
- ✗ How to inverse it?



Solve the following linear system:

$$Hd = g$$

using **conjugate gradient** method.

1. compute g (as usual in standard gradient descent);
2. initialize d_0 arbitrary;
3. **for** $i = 0, 1, 2 \ldots M$**:**

Reality check: can we compute inverse hessian?

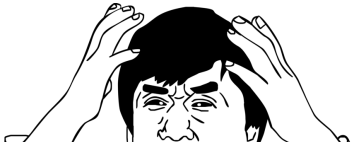
$$\theta_{k+1} = \theta_k + \alpha H^{-1} g$$

Danger!

Natural gradient descent is similar to second-order methods.

Suppose h is the dimension of θ...

- ✗ H is a matrix $h \times h$ (a hessian of KL-divergence);
- ✗ How to inverse it?



Solve the following linear system:

$$Hd = g$$

using **conjugate gradient** method.

1. compute g (as usual in standard gradient descent);
2. initialize d_0 arbitrary;
3. **for** $i = 0, 1, 2 \ldots M$**:**
   - conjugate gradient method gives you new approximation of solution d_{i+1} using g and vector Hd_i.

Reality check: can we compute inverse hessian?

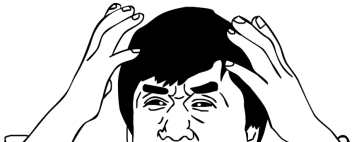
$$\theta_{k+1} = \theta_k + \alpha H^{-1} g$$

Danger!

Natural gradient descent is similar to second-order methods.

Suppose h is the dimension of θ...

- ✗ H is a matrix $h \times h$ (a hessian of KL-divergence);
- ✗ How to inverse it?



Solve the following linear system:

$$Hd = g$$

using **conjugate gradient** method.

1. compute g (as usual in standard gradient descent);
2. initialize d_0 arbitrary;
3. **for** $i = 0, 1, 2 \ldots M$**:**
   - conjugate gradient method gives you new approximation of solution d_{i+1} using g and vector Hd_i.
4. use line-search to determine α_k;

Reality check: can we compute inverse hessian?

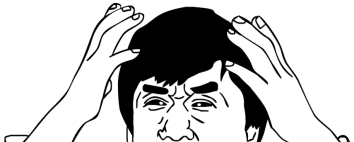
$$\theta_{k+1} = \theta_k + \alpha H^{-1} g$$

Danger!

Natural gradient descent is similar to second-order methods.

Suppose h is the dimension of θ...

- ✗ H is a matrix $h \times h$ (a hessian of KL-divergence);
- ✗ How to inverse it?



Solve the following linear system:

$$Hd = g$$

using **conjugate gradient** method.

1. compute g (as usual in standard gradient descent);
2. initialize d_0 arbitrary;
3. **for** $i = 0, 1, 2 \ldots M$**:**
   - conjugate gradient method gives you new approximation of solution d_{i+1} using g and vector Hd_i.
4. use line-search to determine α_k;
4. $\theta_{k+1} = \theta_k + \alpha_k d_M$

Line Search

How to determine learning rate α_k?

(it is interconnected with δ in trust region view)

$$\theta_{k+1} = \theta_k + \alpha_k d_k$$

Line Search

How to determine learning rate α_k?

(it is interconnected with δ in trust region view)

$$\theta_{k+1} = \theta_k + \alpha_k d_k$$

Use **backtracking** to find α_k, so that:

$$L_{\pi^{\text{old}}}(\theta_{k+1}) > 0, \qquad \mathbb{E}_s \, \mathrm{KL}(\pi^{\text{old}}(\cdot \mid s) \parallel \pi_{\theta_{k+1}}(\cdot \mid s))) \leqslant \delta$$

Line Search

How to determine learning rate α_k?

(it is interconnected with δ in trust region view)

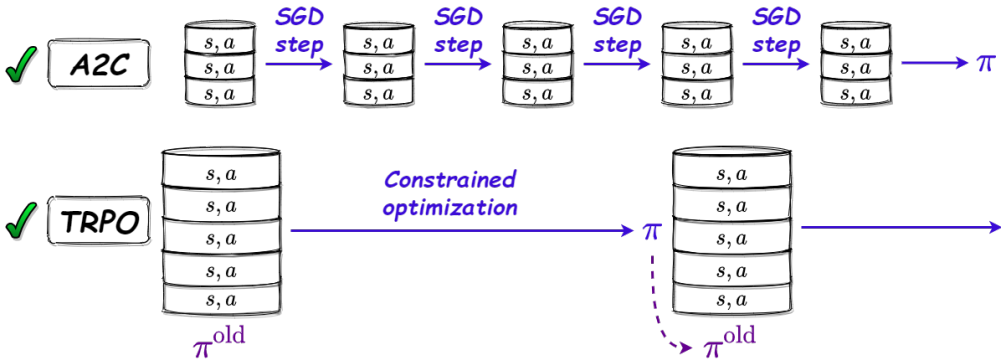
$$\theta_{k+1} = \theta_k + \alpha_k d_k$$

Use **backtracking** to find α_k, so that:

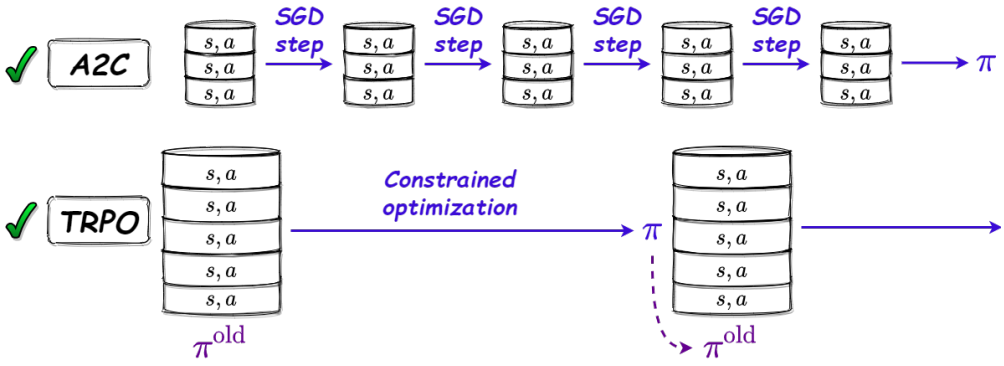
$$L_{\pi^{\text{old}}}(\theta_{k+1}) > 0, \qquad \mathbb{E}_s \, \mathrm{KL}(\pi^{\text{old}}(\cdot \mid s) \parallel \pi_{\theta_{k+1}}(\cdot \mid s))) \leqslant \delta$$

e. g. decrease α_k in half until these constraints are met.

TRPO: pipeline



TRPO: pipeline



- ✓ better utilization of GAE
- ✓ more sample-efficient than A2C
- ✓ considered quite robust

- ✗ critic has to be trained in separate optimization process
- ✗ computationally expensive
- ✗ complicated :{