Carnegie Mellon

School of Computer Science

Deep Reinforcement Learning and Control

# Natural Policy Gradients

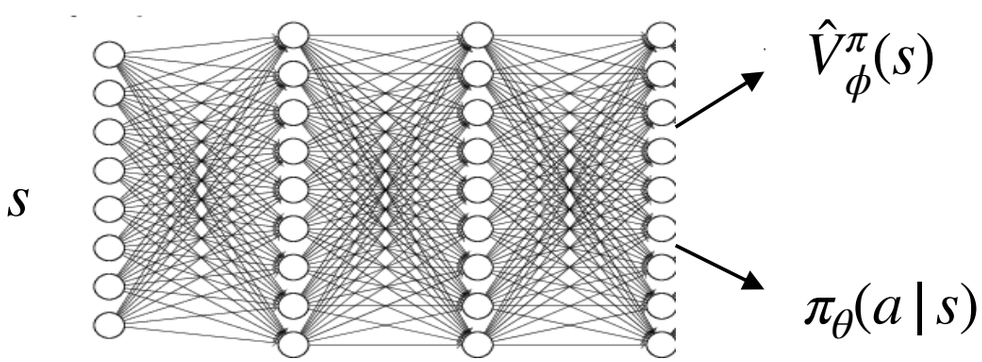
CMU 10-703

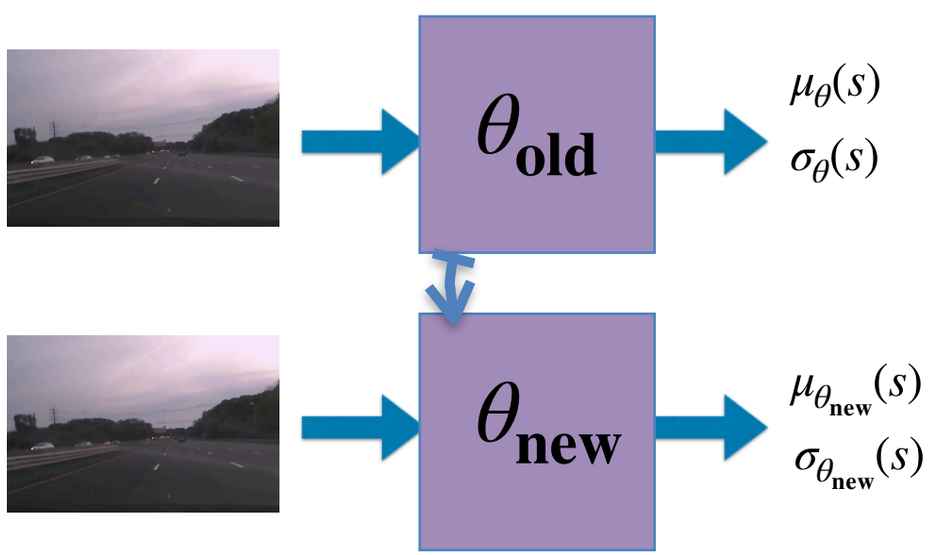
Katerina Fragkiadaki

# Actor-critic

1. Sample trajectories $\{s_t^i, a_t^i\}_{i=0}^T$ by running the current policy $a \sim \pi_\theta(s)$
2. Fit value function $V_\phi^\pi(s)$ by MC or TD estimation (update $\phi$)
3. Compute advantages $A^\pi(s_t^i, a_t^i) = R(s_t^i, a_t^i) + \gamma V_\phi^\pi(s_{t+1}^i) - V_\phi^\pi(s_t^i)$
4. $\nabla_\theta U(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t^i \mid s_t^i) A^\pi(s_t^i, a_t^i)$
5. $\theta' = \theta + \alpha \nabla_\theta U(\theta)$

This lecture is about this stepsize



# Choosing a stepsize

Policy gradients:

$$\hat{g}^{PG} \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_t^{(i)} \mid s_t^{(i)}) A^\pi(s_t^{(i)}, a_t^{(i)}), \quad \tau_i \sim \pi_\theta$$

Compare this to supervised learning using expert actions $\tilde{a} \sim \pi^*$ and a maximum likelihood objective:

$$U^{SL}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\log \pi_\theta(\tilde{a}_t^{(i)} \mid s_t^{(i)}), \quad \tau_i \sim \pi^* \qquad \text{(+regularization)}$$

with gradient:

$$\hat{g}^{SL} \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(\tilde{a}_t^{(i)} \mid s_t^{(i)}), \quad \tau_i \sim \pi^*$$

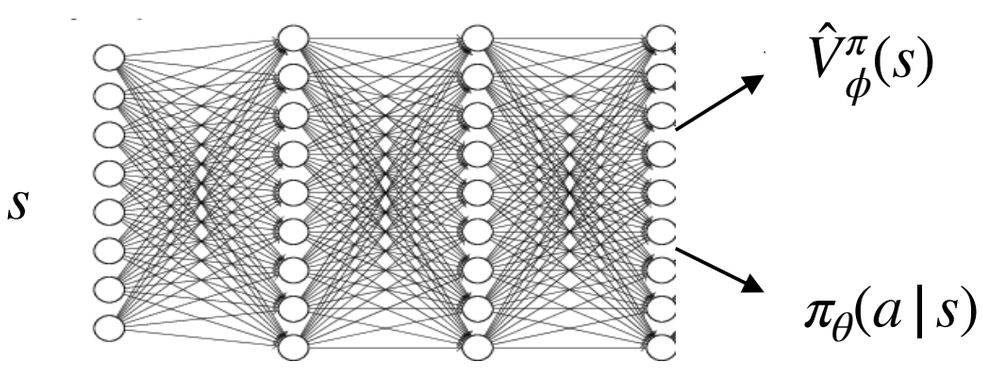
We want to optimize both objectives using gradient descent

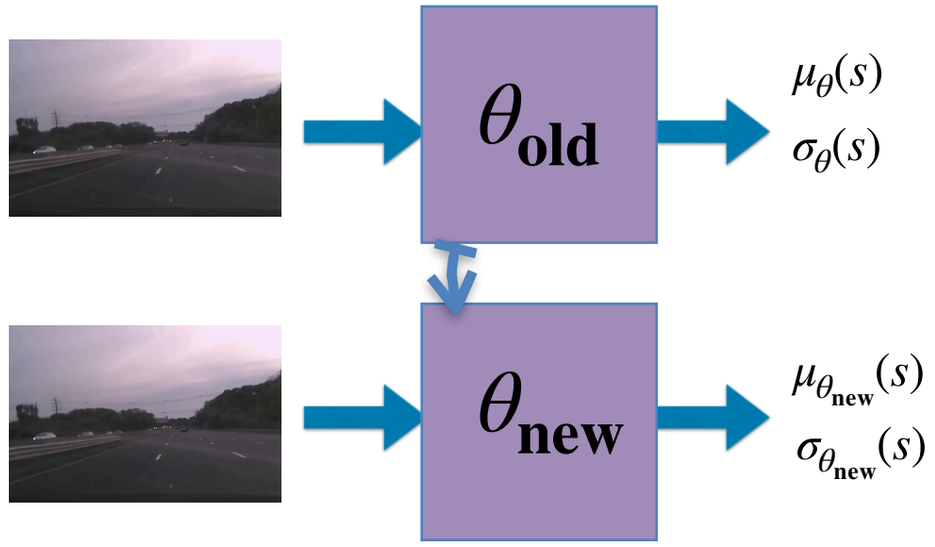
$$\theta' = \theta + \alpha \nabla_\theta U(\theta)$$

Choosing the right  stepsize is more critical for RL than for SL.

# Choosing a stepsize

- Step too big
  Bad policy->data collected under bad policy-> we cannot recover
  (in Supervised Learning, data does not depend on neural network weights)
- Step too small
  Not efficient use of experience
  (in Supervised Learning, data can be trivially re-used)

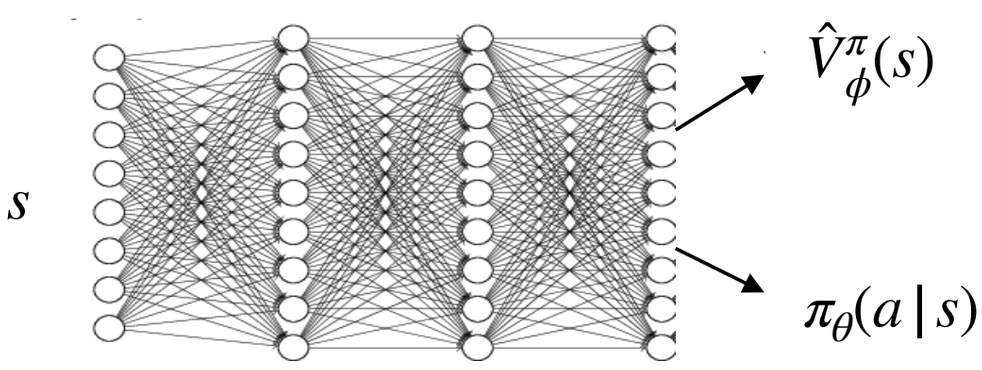


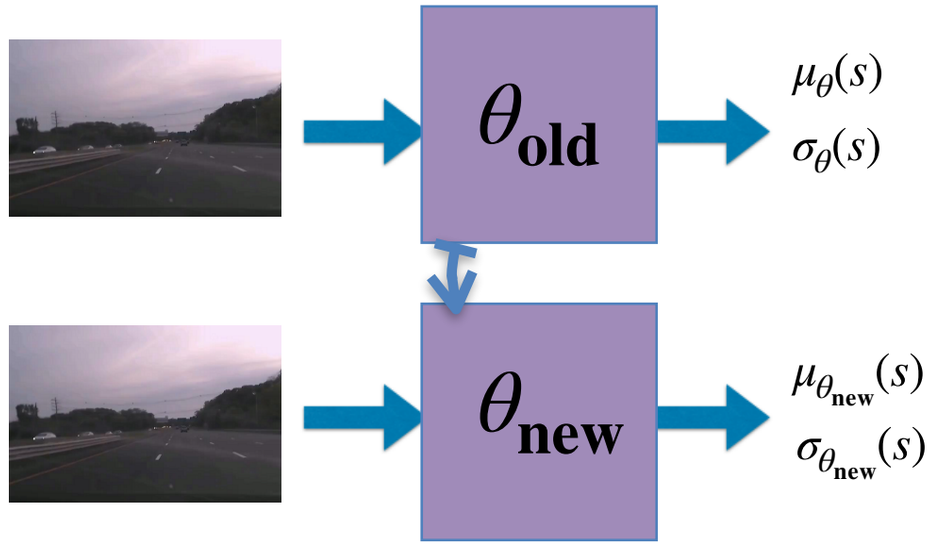



# Choosing a stepsize

- Step too big
  Bad policy->data collected under bad policy-> we cannot recover
  (in Supervised Learning, data does not depend on neural network weights)
- Step too small
  Not efficient use of experience
  (in Supervised Learning, data can be trivially re-used)

Gradient descent in parameter space does not take into account the resulting distance in the (output) policy space between $\pi_{\theta_{\text{old}}}(s)$ and $\pi_{\theta_{\text{new}}}(s)$

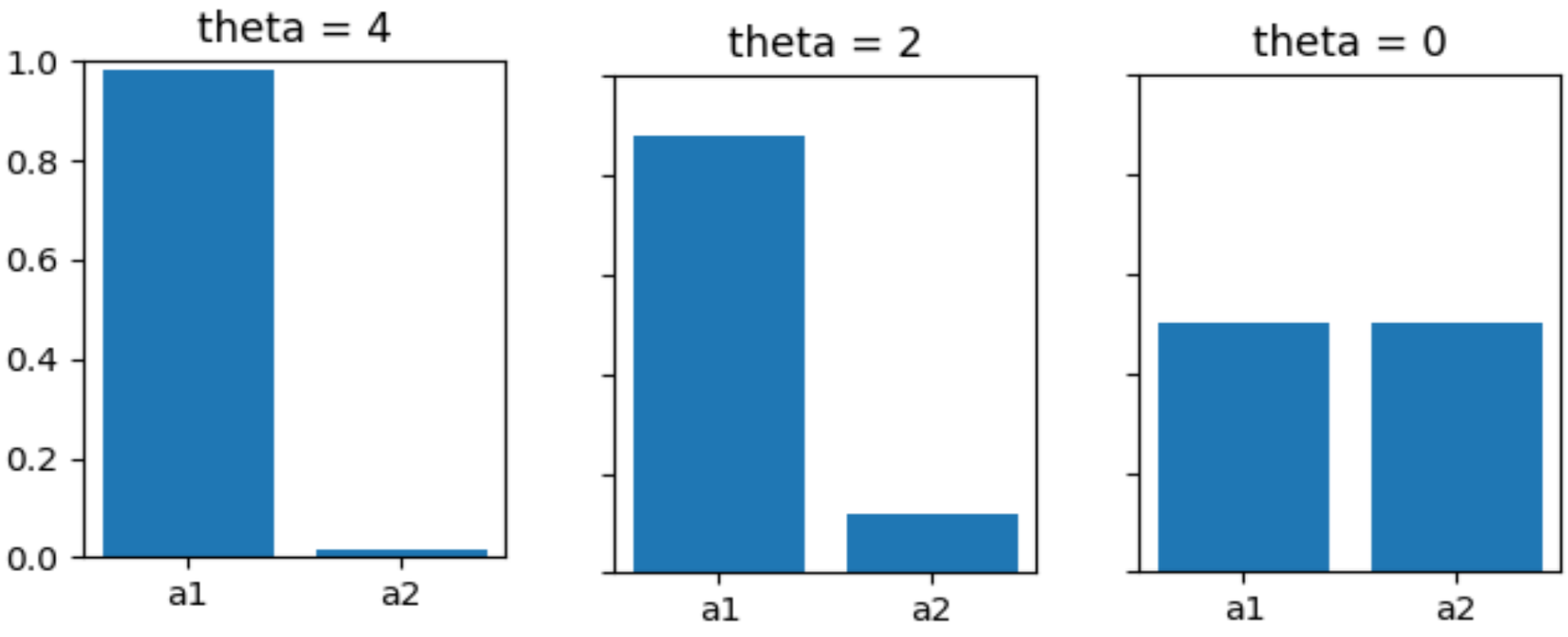


# Hard to choose stepsizes

Consider a family of policies with parametrization:

$$\pi_\theta(a) = \begin{cases} \sigma(\theta) & a = 1 \\ 1 - \sigma(\theta) & a = 2 \end{cases}$$




The same parameter step $\Delta\theta = -2$ changes the policy distribution more or less dramatically depending on where in the parameter space we are.

# Notation

We will use the following to denote values of parameters and corresponding policies before and after an update:

$$\theta_{old} \rightarrow \theta_{new}$$
$$\pi_{old} \rightarrow \pi_{new}$$

$$\theta \rightarrow \theta'$$
$$\pi \rightarrow \pi'$$

# Gradient Descent in Parameter Space

The stepwise in gradient descent results from solving the following optimization problem:

$$d* = \arg\max_{\|d\| \leq \epsilon} J(\theta + d)$$

SGD: $\theta_{new} = \theta_{old} + d*$

Euclidean distance in parameter space

It is hard to predict the result on the parameterized distribution.. hard to pick the threshold epsilon

# Gradient Descent in Distribution Space

The stepwise in gradient descent results from solving the following optimization problem:

$$d* = \arg\max_{\|d\| \le \epsilon} J(\theta + d)$$

SGD: $\theta_{new} = \theta_{old} + d*$

Euclidean distance in parameter space

It is hard to predict the result on the parameterized distribution.. hard to pick the threshold epsilon

Natural gradient descent: the stepwise in parameter space is determined by considering the KL divergence in the distributions before and after the update:

$$d* = \arg\max_{d,\ s.t.\ \mathrm{KL}(\pi_\theta \| \pi_{\theta+d}) \le \epsilon} J(\theta + d)$$

KL divergence in distribution space

Easier to pick the distance threshold!

$$D_{\mathrm{KL}}(P\|Q) = \sum_i P(i)\ \log\left(\frac{P(i)}{Q(i)}\right)$$

$$D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x)\ \log\left(\frac{p(x)}{q(x)}\right) dx$$

# Solving the KL Constrained Problem

Unconstrained penalized objective:

$$d* = \arg\max_d U(\theta + d) - \lambda(\mathrm{D}_{\mathrm{KL}}\left[\pi_\theta \| \pi_{\theta+d}\right] - \epsilon)$$

$$\approx \arg\max_d U(\theta_{old}) + \nabla_\theta U(\theta)\,|_{\theta=\theta_{old}} \cdot d - \frac{1}{2}\lambda(d^\top \nabla^2_\theta \mathrm{D}_{\mathrm{KL}}\left[\pi_{\theta_{old}} \| \pi_\theta\right]|_{\theta=\theta_{old}} d) + \lambda\epsilon$$

(First order Taylor expansion for the loss and second order for the KL)

# Taylor expansion of KL

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) \approx \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_{\theta_{old}}) + d^\top \nabla_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} + \frac{1}{2} d^\top \nabla_\theta^2 \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} d$$

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) = \mathbb{E}_{x\sim p_{\theta_{old}}} \log\left(\frac{P_{\theta_{old}}(x)}{P_\theta(x)}\right)$$

# Taylor expansion of KL

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) \approx \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_{\theta_{old}}) + d^\top \nabla_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} + \frac{1}{2} d^\top \nabla^2_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} d$$

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) = \mathbb{E}_{x\sim p_{\theta_{old}}} \log\left(\frac{P_{\theta_{old}}(x)}{P_\theta(x)}\right)$$

# Taylor expansion of KL

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) \approx \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_{\theta_{old}}) + d^\top \nabla_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} + \frac{1}{2} d^\top \nabla_\theta^2 \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} d$$

$$\nabla_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} = -\nabla_\theta \mathbb{E}_{x\sim p_{\theta_{old}}} \log P_\theta(x)|_{\theta=\theta_{old}} + \nabla_\theta \mathbb{E}_{x\sim p_{\theta_{old}}} \log P_{\theta_{old}}(x)|_{\theta=\theta_{old}}$$

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) = \mathbb{E}_{x\sim p_{\theta_{old}}} \log \left( \frac{P_{\theta_{old}}(x)}{P_\theta(x)} \right)$$

# Taylor expansion of KL

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) \approx \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_{\theta_{old}}) + d^\top \nabla_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} + \frac{1}{2} d^\top \nabla_\theta^2 \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} d$$

$$\begin{aligned}\nabla_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} &= -\nabla_\theta \mathbb{E}_{x\sim p_{\theta_{old}}} \log P_\theta(x)|_{\theta=\theta_{old}} + \nabla_\theta \mathbb{E}_{x\sim p_{\theta_{old}}} \log P_{\theta_{old}}(x)|_{\theta=\theta_{old}} \\ &= -\mathbb{E}_{x\sim p_{\theta_{old}}} \nabla_\theta \log P_\theta(x)|_{\theta=\theta_{old}}\end{aligned}$$

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) = \mathbb{E}_{x\sim p_{\theta_{old}}} \log\left(\frac{P_{\theta_{old}}(x)}{P_\theta(x)}\right)$$

# Taylor expansion of KL

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) \approx \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_{\theta_{old}}) + d^\top \nabla_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} + \frac{1}{2} d^\top \nabla^2_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} d$$

$$\begin{aligned}
\nabla_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} &= -\nabla_\theta \mathbb{E}_{x\sim p_{\theta_{old}}} \log P_\theta(x)|_{\theta=\theta_{old}} + \nabla_\theta \mathbb{E}_{x\sim p_{\theta_{old}}} \log P_{\theta_{old}}(x)|_{\theta=\theta_{old}} \\
&= -\mathbb{E}_{x\sim p_{\theta_{old}}} \nabla_\theta \log P_\theta(x)|_{\theta=\theta_{old}} \\
&= -\mathbb{E}_{x\sim p_{\theta_{old}}} \frac{1}{P_{\theta_{old}}(x)} \nabla_\theta P_\theta(x)|_{\theta=\theta_{old}}
\end{aligned}$$

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) = \mathbb{E}_{x\sim p_{\theta_{old}}} \log\left(\frac{P_{\theta_{old}}(x)}{P_\theta(x)}\right)$$

# Taylor expansion of KL

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) \approx \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_{\theta_{old}}) + d^\top \nabla_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} + \frac{1}{2} d^\top \nabla^2_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} d$$

$$\begin{aligned}
\nabla_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} &= -\nabla_\theta \mathbb{E}_{x\sim p_{\theta_{old}}} \log P_\theta(x)|_{\theta=\theta_{old}} + \nabla_\theta \mathbb{E}_{x\sim p_{\theta_{old}}} \log P_{\theta_{old}}(x)|_{\theta=\theta_{old}} \\
&= -\mathbb{E}_{x\sim p_{\theta_{old}}} \nabla_\theta \log P_\theta(x)|_{\theta=\theta_{old}} \\
&= -\mathbb{E}_{x\sim p_{\theta_{old}}} \frac{1}{P_{\theta_{old}}(x)} \nabla_\theta P_\theta(x)|_{\theta=\theta_{old}} \\
&= \int_x P_{\theta_{old}}(x) \frac{1}{P_{\theta_{old}}(x)} \nabla_\theta P_\theta(x)|_{\theta=\theta_{old}}
\end{aligned}$$

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) = \mathbb{E}_{x\sim p_{\theta_{old}}} \log\left(\frac{P_{\theta_{old}}(x)}{P_\theta(x)}\right)$$

# Taylor expansion of KL

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) \approx \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_{\theta_{old}}) + d^\top \nabla_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} + \frac{1}{2} d^\top \nabla_\theta^2 \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} d$$

$$\begin{aligned}
\nabla_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} &= -\nabla_\theta \mathbb{E}_{x\sim p_{\theta_{old}}} \log P_\theta(x)|_{\theta=\theta_{old}} + \nabla_\theta \mathbb{E}_{x\sim p_{\theta_{old}}} \log P_{\theta_{old}}(x)|_{\theta=\theta_{old}} \\
&= -\mathbb{E}_{x\sim p_{\theta_{old}}} \nabla_\theta \log P_\theta(x)|_{\theta=\theta_{old}} \\
&= -\mathbb{E}_{x\sim p_{\theta_{old}}} \frac{1}{P_{\theta_{old}}(x)} \nabla_\theta P_\theta(x)|_{\theta=\theta_{old}} \\
&= \int_x P_{\theta_{old}}(x) \frac{1}{P_{\theta_{old}}(x)} \nabla_\theta P_\theta(x)|_{\theta=\theta_{old}} \\
&= \int_x \nabla_\theta P_\theta(x)|_{\theta=\theta_{old}}
\end{aligned}$$

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) = \mathbb{E}_{x\sim p_{\theta_{old}}} \log\left(\frac{P_{\theta_{old}}(x)}{P_\theta(x)}\right)$$

# Taylor expansion of KL

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) \approx \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_{\theta_{old}}) + d^\top \nabla_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} + \frac{1}{2} d^\top \nabla_\theta^2 \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} d$$

$$\begin{aligned}
\nabla_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} &= -\nabla_\theta \mathbb{E}_{x\sim p_{\theta_{old}}} \log P_\theta(x)|_{\theta=\theta_{old}} + \nabla_\theta \mathbb{E}_{x\sim p_{\theta_{old}}} \log P_{\theta_{old}}(x)|_{\theta=\theta_{old}} \\
&= -\mathbb{E}_{x\sim p_{\theta_{old}}} \nabla_\theta \log P_\theta(x)|_{\theta=\theta_{old}} \\
&= -\mathbb{E}_{x\sim p_{\theta_{old}}} \frac{1}{P_{\theta_{old}}(x)} \nabla_\theta P_\theta(x)|_{\theta=\theta_{old}} \\
&= \int_x P_{\theta_{old}}(x) \frac{1}{P_{\theta_{old}}(x)} \nabla_\theta P_\theta(x)|_{\theta=\theta_{old}} \\
&= \int_x \nabla_\theta P_\theta(x)|_{\theta=\theta_{old}} \\
&= \nabla_\theta \int_x P_\theta(x)|_{\theta=\theta_{old}} . \\
&= 0
\end{aligned}$$

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) = \mathbb{E}_{x\sim p_{\theta_{old}}} \log\left(\frac{P_{\theta_{old}}(x)}{P_\theta(x)}\right)$$

# Taylor expansion of KL

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) \approx \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_{\theta_{old}}) + d^\top \nabla_\theta \mathrm{KL}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} + \frac{1}{2} d^\top \nabla^2_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} d$$

$$\nabla^2_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} = -\mathbb{E}_{x\sim p_{\theta_{old}}} \nabla^2_\theta \log P_\theta(x)|_{\theta=\theta_{old}}$$

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) = \mathbb{E}_{x\sim p_{\theta_{old}}} \log\left(\frac{P_{\theta_{old}}(x)}{P_\theta(x)}\right)$$

# Taylor expansion of KL

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) \approx \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_{\theta_{old}}) + d^\top \nabla_\theta \mathrm{KL}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} + \frac{1}{2} d^\top \nabla_\theta^2 \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} d$$

$$\nabla_\theta^2 \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} = -\mathbb{E}_{x\sim p_{\theta_{old}}} \nabla_\theta^2 \log P_\theta(x)|_{\theta=\theta_{old}}$$

$$= -\mathbb{E}_{x\sim p_{\theta_{old}}} \nabla_\theta \left( \frac{\nabla_\theta P_\theta(x)}{P_\theta(x)} \right)|_{\theta=\theta_{old}}$$

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) = \mathbb{E}_{x\sim p_{\theta_{old}}} \log \left( \frac{P_{\theta_{old}}(x)}{P_\theta(x)} \right)$$

# Taylor expansion of KL

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) \approx \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_{\theta_{old}}) + d^\top \nabla_\theta \mathrm{KL}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} + \frac{1}{2} d^\top \nabla_\theta^2 \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} d$$

$$\begin{aligned}
\nabla_\theta^2 \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} &= -\mathbb{E}_{x\sim p_{\theta_{old}}} \nabla_\theta^2 \log P_\theta(x)|_{\theta=\theta_{old}} \\
&= -\mathbb{E}_{x\sim p_{\theta_{old}}} \nabla_\theta \left( \frac{\nabla_\theta P_\theta(x)}{P_\theta(x)} \right)|_{\theta=\theta_{old}} \\
&= -\mathbb{E}_{x\sim p_{\theta_{old}}} \left( \frac{\nabla_\theta^2 P_\theta(x) P_\theta(x) - \nabla_\theta P_\theta(x) \nabla_\theta P_\theta(x)^\top}{P_\theta(x)^2} \right)|_{\theta=\theta_{old}}
\end{aligned}$$

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) = \mathbb{E}_{x\sim p_{\theta_{old}}} \log \left( \frac{P_{\theta_{old}}(x)}{P_\theta(x)} \right)$$

# Taylor expansion of KL

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) \approx \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_{\theta_{old}}) + d^\top \nabla_\theta \mathrm{KL}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} + \frac{1}{2} d^\top \nabla_\theta^2 \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} d$$

$$\nabla_\theta^2 \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} = -\mathbb{E}_{x\sim p_{\theta_{old}}} \nabla_\theta^2 \log P_\theta(x)|_{\theta=\theta_{old}}$$

$$= -\mathbb{E}_{x\sim p_{\theta_{old}}} \nabla_\theta \left( \frac{\nabla_\theta P_\theta(x)}{P_\theta(x)} \right)|_{\theta=\theta_{old}}$$

$$= -\mathbb{E}_{x\sim p_{\theta_{old}}} \left( \frac{\nabla_\theta^2 P_\theta(x) P_\theta(x) - \nabla_\theta P_\theta(x) \nabla_\theta P_\theta(x)^\top}{P_\theta(x)^2} \right)|_{\theta=\theta_{old}}$$

$$= -\mathbb{E}_{x\sim p_{\theta_{old}}} \frac{\nabla_\theta^2 P_\theta(x)|_{\theta=\theta_{old}}}{P_{\theta_{old}}(x)} + \mathbb{E}_{x\sim p_{\theta_{old}}} \nabla_\theta \log P_\theta(x) \nabla_\theta \log P_\theta(x)^\top|_{\theta=\theta_{old}}$$

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) = \mathbb{E}_{x\sim p_{\theta_{old}}} \log \left( \frac{P_{\theta_{old}}(x)}{P_\theta(x)} \right)$$

# Taylor expansion of KL

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) \approx \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_{\theta_{old}}) + d^\top \nabla_\theta \mathrm{KL}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} + \frac{1}{2} d^\top \nabla_\theta^2 \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} d$$

$$\nabla_\theta^2 \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta)|_{\theta=\theta_{old}} = -\mathbb{E}_{x\sim p_{\theta_{old}}} \nabla_\theta^2 \log P_\theta(x)|_{\theta=\theta_{old}}$$

$$= -\mathbb{E}_{x\sim p_{\theta_{old}}} \nabla_\theta \left( \frac{\nabla_\theta P_\theta(x)}{P_\theta(x)} \right)|_{\theta=\theta_{old}}$$

$$= -\mathbb{E}_{x\sim p_{\theta_{old}}} \left( \frac{\nabla_\theta^2 P_\theta(x) P_\theta(x) - \nabla_\theta P_\theta(x) \nabla_\theta P_\theta(x)^\top}{P_\theta(x)^2} \right)|_{\theta=\theta_{old}}$$

$$= -\mathbb{E}_{x\sim p_{\theta_{old}}} \frac{\nabla_\theta^2 P_\theta(x)|_{\theta=\theta_{old}}}{P_{\theta_{old}}(x)} + \mathbb{E}_{x\sim p_{\theta_{old}}} \nabla_\theta \log P_\theta(x) \nabla_\theta \log P_\theta(x)^\top|_{\theta=\theta_{old}}$$

$$= \mathbb{E}_{x\sim p_{\theta_{old}}} \nabla_\theta \log P_\theta(x) \nabla_\theta \log P_\theta(x)^\top|_{\theta=\theta_{old}}$$

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}}|p_\theta) = \mathbb{E}_{x\sim p_{\theta_{old}}} \log \left( \frac{P_{\theta_{old}}(x)}{P_\theta(x)} \right)$$

# Fisher Information Matrix

Exactly equivalent to the Hessian of KL divergence!

$$\mathbf{F}(\theta) = \mathbb{E}_{x\sim p_\theta}\left[\nabla_\theta \log p_\theta(x)\, \nabla_\theta \log p_\theta(x)^\top\right]$$

$$\mathbf{F}(\theta_{old}) = \nabla^2_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}} | p_\theta)|_{\theta=\theta_{old}}$$

$$\mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}} | p_\theta) \approx \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}} | p_{\theta_{old}}) + d^\top \nabla_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}} | p_\theta)|_{\theta=\theta_{old}} + \frac{1}{2} d^\top \nabla^2_\theta \mathrm{D}_{\mathrm{KL}}(p_{\theta_{old}} | p_\theta)|_{\theta=\theta_{old}} d$$

$$= \frac{1}{2} d^\top \mathbf{F}(\theta_{old}) d$$

$$= \frac{1}{2} (\theta - \theta_{old})^\top \mathbf{F}(\theta_{old})(\theta - \theta_{old})$$

Since KL divergence is roughly analogous to a distance measure between distributions, Fisher information serves as a local distance metric between distributions:  how much you change the distribution if you move the parameters a little bit in a given direction.

# Solving the KL Constrained Problem

Unconstrained penalized objective:

$$d* = \arg\max_d U(\theta + d) - \lambda(\mathrm{D}_{\mathrm{KL}}\left[\pi_\theta \| \pi_{\theta+d}\right] - \epsilon)$$

First order Taylor expansion for the loss and second order for the KL:

$$\approx \arg\max_d U(\theta_{old}) + \nabla_\theta U(\theta)\,|_{\theta=\theta_{old}} \cdot d - \frac{1}{2}\lambda(d^\top \nabla^2_\theta \mathrm{D}_{\mathrm{KL}}\left[\pi_{\theta_{old}} \| \pi_\theta\right]|_{\theta=\theta_{old}} d) + \lambda\epsilon$$

Substitute for the information matrix:

$$= \arg\max_d \nabla_\theta U(\theta)\,|_{\theta=\theta_{old}} \cdot d - \frac{1}{2}\lambda(d^\top \mathbf{F}(\theta_{old})d)$$

$$= \arg\min_d - \nabla_\theta U(\theta)\,|_{\theta=\theta_{old}} \cdot d + \frac{1}{2}\lambda(d^\top \mathbf{F}(\theta_{old})d)$$

# Natural Gradient Descent

Setting the gradient to zero:

$$0 = \frac{\partial}{\partial d}\left(-\nabla_\theta U(\theta)\,|_{\theta=\theta_{old}} \cdot d + \frac{1}{2}\lambda(d^\top \mathbf{F}(\theta_{old})d)\right)$$

$$= -\nabla_\theta U(\theta)\,|_{\theta=\theta_{old}} + \frac{1}{2}\lambda(\mathbf{F}(\theta_{old}))d$$

$$d = \frac{2}{\lambda}\mathbf{F}^{-1}(\theta_{old})\,\nabla_\theta U(\theta)\,|_{\theta=\theta_{old}}$$

The natural gradient: $g_N = \mathbf{F}^{-1}(\theta_{old})\,\nabla_\theta U(\theta)$

$$\theta_{new} = \theta_{old} + \alpha \cdot g_N$$

# Stepsize along the Natural Gradient direction

The natural gradient: $g_N = \mathbf{F}^{-1}(\theta_{old}) \nabla_\theta U(\theta)$

$$\theta_{new} = \theta_{old} + \alpha \cdot g_N$$

Let's solve for the stepzise along the natural gradient direction!

$$\mathrm{D}_{\mathrm{KL}}(\pi_{\theta_{old}} | \pi_\theta) \approx \frac{1}{2}(\theta - \theta_{old})^\top \mathbf{F}(\theta_{old})(\theta - \theta_{old}) = \frac{1}{2}(\alpha g_N)^\top \mathbf{F}(\alpha g_N)$$

I want the KL between old and new policies to be at most $\epsilon$:

$$\frac{1}{2}(\alpha g_N)^\top \mathbf{F}(\alpha g_N) = \epsilon$$

$$\alpha = \sqrt{\frac{2\epsilon}{(g_N^\top \mathbf{F} g_N)}}$$

# Natural Gradient Descent

**Algorithm 1** Natural Policy Gradient

Input: initial policy parameters $\theta_0$

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

Collect set of trajectories $\mathcal{D}_k$ on policy $\pi_k = \pi(\theta_k)$

Estimate advantages $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm

Form sample estimates for

- policy gradient $\hat{g}_k$ (using advantage estimates)
- and KL-divergence Hessian / Fisher Information Matrix $\hat{H}_k$

Compute Natural Policy Gradient update:

$$\theta_{k+1} = \theta_k + \sqrt{\frac{2\,\epsilon}{\hat{g}_k^T \hat{H}_k^{-1} \hat{g}_k}}\, \hat{H}_k^{-1} \hat{g}_k$$

**end for**

Both use samples from the current policy $\pi_k = \pi(\theta_k)$

# Natural Gradient Descent

**Algorithm: Natural Gradient Descent**

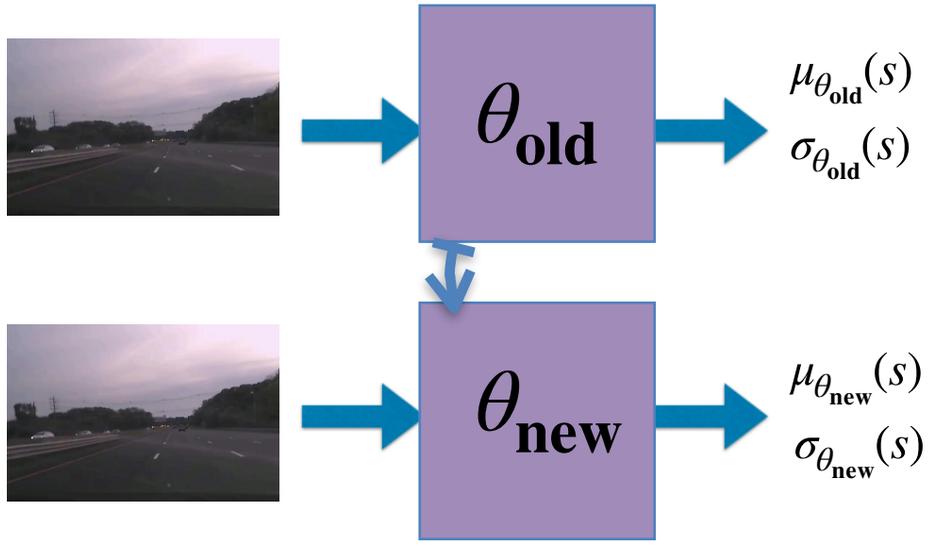
1. Repeat:
    1. Do forward pass on our model and compute loss $\mathcal{L}(\theta)$.
    2. Compute the gradient $\nabla_\theta \mathcal{L}(\theta)$.
    3. Compute the Fisher Information Matrix F, or its empirical version (wrt. our training data).
    4. Compute the natural gradient $\tilde{\nabla}_\theta \mathcal{L}(\theta) = \mathrm{F}^{-1} \nabla_\theta \mathcal{L}(\theta)$.
    5. Update the parameter: $\theta = \theta - \alpha \, \tilde{\nabla}_\theta \mathcal{L}(\theta)$, where $\alpha$ is the learning rate.
2. Until convergence.

# Policy Gradients

Monte Carlo Policy Gradients (REINFORCE), gradient direction: $\hat{g} = \hat{\mathbb{E}}_t \left[ \nabla_\theta \log \pi_\theta(a_t \mid s_t) \hat{A}_t \right]$

Actor-Critic Policy Gradient: $\hat{g} = \hat{\mathbb{E}}_t \left[ \nabla_\theta \log \pi_\theta(a_t \mid s_t) A_{\mathbf{w}}(s_t) \right]$

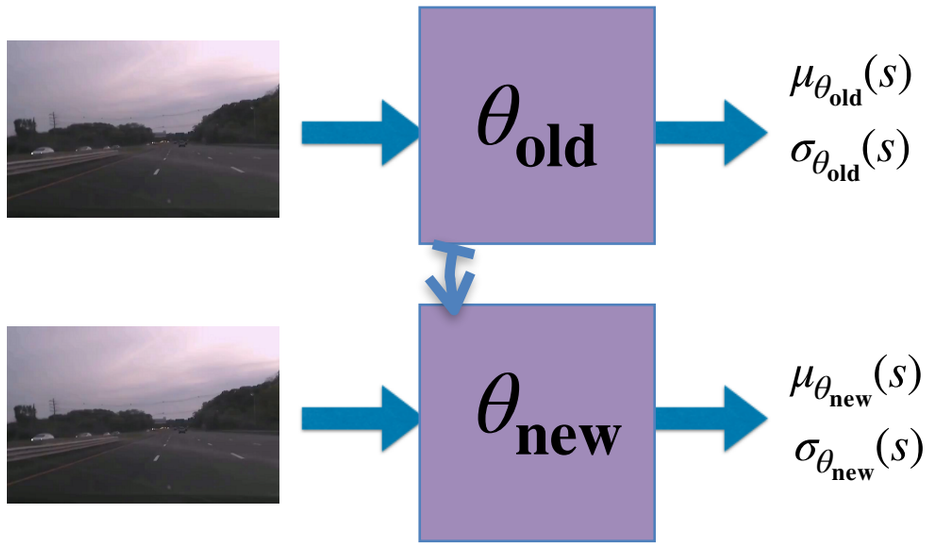
1. Collect trajectories for policy $\pi_{\theta_{old}}$
2. Estimate advantages $A$
3. Compute policy gradient $\hat{g}$
4. Update policy parameters $\theta_{new} = \theta_{old} + \epsilon \cdot \hat{g}$
5. GOTO 1



# Policy Gradients

1. Collect trajectories for policy $\pi_{\theta_{old}}$
2. Estimate advantages $A$
3. Compute policy gradient $\hat{g}$
4. Update policy parameters $\theta_{new} = \theta_{old} + \epsilon \cdot \hat{g}$
5. GOTO 1

- On policy learning can be extremely inefficient
- The policy changes only a little bit with each gradient step
- I want to be able to use earlier data..how to do that?



# Off-policy learning with Importance Sampling

$$
\begin{aligned}
U(\theta) &= \mathbb{E}_{\tau \sim \pi_\theta(\tau)} \left[ R(\tau) \right] \\
&= \sum_{\tau} \pi_\theta(\tau) R(\tau)
\end{aligned}
$$

# Off-policy learning with Importance Sampling

$$
\begin{aligned}
U(\theta) &= \mathbb{E}_{\tau \sim \pi_\theta(\tau)} \left[ R(\tau) \right] \\
&= \sum_{\tau} \pi_\theta(\tau) R(\tau) \\
&= \sum_{\tau} \pi_{\theta_{old}}(\tau) \frac{\pi_\theta(\tau)}{\pi_{\theta_{old}}(\tau)} R(\tau)
\end{aligned}
$$

# Off-policy learning with Importance Sampling

$$
\begin{aligned}
U(\theta) &= \mathbb{E}_{\tau \sim \pi_\theta(\tau)} \left[ R(\tau) \right] \\
&= \sum_\tau \pi_\theta(\tau) R(\tau) \\
&= \sum_\tau \pi_{\theta_{old}}(\tau) \frac{\pi_\theta(\tau)}{\pi_{\theta_{old}}(\tau)} R(\tau) \\
&= \mathbb{E}_{\tau \sim \pi_{\theta_{old}}} \frac{\pi_\theta(\tau)}{\pi_{\theta_{old}}(\tau)} R(\tau)
\end{aligned}
$$

# Off-policy learning with Importance Sampling

$$
\begin{aligned}
U(\theta) &= \mathbb{E}_{\tau \sim \pi_\theta(\tau)} \left[ R(\tau) \right] \\
&= \sum_\tau \pi_\theta(\tau) R(\tau) \\
&= \sum_\tau \pi_{\theta_{old}}(\tau) \frac{\pi_\theta(\tau)}{\pi_{\theta_{old}}(\tau)} R(\tau) \\
&= \mathbb{E}_{\tau \sim \pi_{\theta_{old}}} \frac{\pi_\theta(\tau)}{\pi_{\theta_{old}}(\tau)} R(\tau)
\end{aligned}
$$

$$
\nabla_\theta U(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta_{old}}} \frac{\nabla_\theta \pi_\theta(\tau)}{\pi_{\theta_{old}}(\tau)} R(\tau)
$$

# Off-policy learning with Importance Sampling

$$
\begin{aligned}
U(\theta) &= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[R(\tau)\right] \\
&= \sum_\tau \pi_\theta(\tau)R(\tau) \\
&= \sum_\tau \pi_{\theta_{old}}(\tau)\frac{\pi_\theta(\tau)}{\pi_{\theta_{old}}(\tau)}R(\tau) \\
&= \mathbb{E}_{\tau\sim\pi_{\theta_{old}}}\frac{\pi_\theta(\tau)}{\pi_{\theta_{old}}(\tau)}R(\tau)
\end{aligned}
$$

$$
\nabla_\theta U(\theta) = \mathbb{E}_{\tau\sim\pi_{\theta_{old}}}\frac{\nabla_\theta\pi_\theta(\tau)}{\pi_{\theta_{old}}(\tau)}R(\tau)
$$

$$
\nabla_\theta U(\theta)\,|_{\theta=\theta_{old}} = \mathbb{E}_{\tau\sim\pi_{\theta_{old}}}\nabla_\theta \log \pi_\theta(\tau)\,|_{\theta=\theta_{old}} R(\tau)
$$

Gradient evaluated at $\theta_{old}$ is unchanged.

# Off policy learning with Importance Sampling

$$U(\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[R(\tau)\right]$$
$$= \sum_\tau \pi_\theta(\tau)R(\tau)$$
$$= \sum_\tau \pi_{\theta_{old}}(\tau)\frac{\pi_\theta(\tau)}{\pi_{\theta_{old}}(\tau)}R(\tau)$$
$$= \mathbb{E}_{\tau\sim\pi_{\theta_{old}}}\frac{\pi_\theta(\tau)}{\pi_{\theta_{old}}(\tau)}R(\tau)$$

$$\frac{\pi_\theta(\tau)}{\pi_{\theta_{old}}(\tau)} = \frac{\prod_{t=1}^T \pi_\theta(a_t\,|\,s_t)}{\prod_{t=1}^T \pi_{\theta_{old}}(a_t\,|\,s_t)}$$

Using temporal structure:

$$U(\theta) = \mathbb{E}_{\tau\sim\pi_{\theta_{old}}} \sum_{t=1}^T \left(\frac{\prod_{t'=1}^t \pi_\theta(a_{t'}\,|\,s_{t'})}{\prod_{t'=1}^t \pi_{\theta_{old}}(a_{t'}\,|\,s_{t'})}\right)\hat{A}_t$$

$$\nabla_\theta U(\theta) = \mathbb{E}_{\tau\sim\pi_{\theta_{old}}}\frac{\nabla_\theta\pi_\theta(\tau)}{\pi_{\theta_{old}}(\tau)}R(\tau)$$

$$\nabla_\theta U(\theta)\,|_{\theta=\theta_{old}} = \mathbb{E}_{\tau\sim\pi_{\theta_{old}}}\nabla_\theta \log\pi_\theta(\tau)\,|_{\theta=\theta_{old}}R(\tau)$$

Gradient evaluated at $\theta_{old}$ is unchanged.

Now we can use data from the old policy, but the variance has increased by a lot! Those multiplications can explode or vanish!

# Trust region Policy Optimization

Constrained objective:

$$\max_{\theta} . \quad \mathbb{E}_t \left[ \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} A(s_t, a_t) \right]$$

$$\text{subject to } \mathbb{E}_t \left[ \mathrm{D}_{\mathrm{KL}} \left[ \pi_{\theta_{old}}( \cdot \mid s_t) \| \pi_\theta( \cdot \mid s_t) \right] \right] \leq \delta$$

Or unonstrained objective:

$$\max_{\theta} . \quad \mathbb{E}_t \left[ \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} A(s_t, a_t) \right] - \beta \mathbb{E}_t \left[ \mathrm{D}_{\mathrm{KL}} \left[ \pi_{\theta_{old}}( \cdot \mid s_t) \| \pi_\theta( \cdot \mid s_t) \right] \right]$$

J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. "Trust Region Policy Optimization".

# Trust region Policy Optimization

Due to the quadratic approximation, the KL constraint may be violated! What if we just do a line search to find the best stepsize, making sure:

- I am improving my objective $J(\theta)$
- The KL constraint is not violated!

$$\begin{aligned} \underset{\theta}{\text{maximize}} \quad & \hat{\mathbb{E}}_t \left[ \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \hat{A}_t \right] \\ \text{subject to} \quad & \hat{\mathbb{E}}_t[\text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]] \leq \delta. \end{aligned}$$

**Algorithm 2** Line Search for TRPO

Compute proposed policy step $\Delta_k = \sqrt{\frac{2\delta}{\hat{g}_k^T \hat{H}_k^{-1} \hat{g}_k}} \hat{H}_k^{-1} \hat{g}_k$

**for** $j = 0, 1, 2, ..., L$ **do**

  Compute proposed update $\theta = \theta_k + \alpha^j \Delta_k$

  **if** $\mathcal{L}_{\theta_k}(\theta) \geq 0$ and $\bar{D}_{KL}(\theta || \theta_k) \leq \delta$ **then**

    accept the update and set $\theta_{k+1} = \theta_k + \alpha^j \Delta_k$

    break

  **end if**

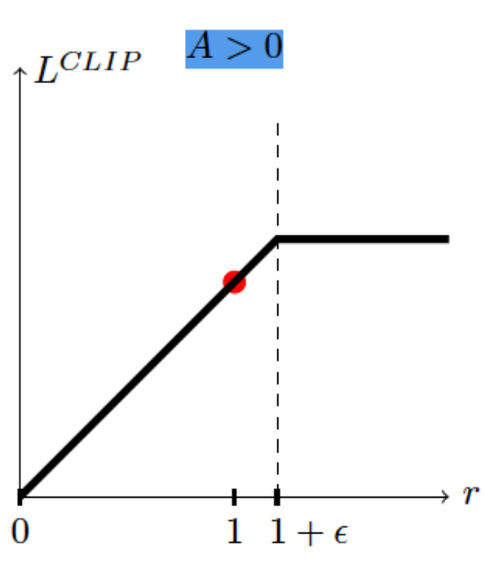
**end for**

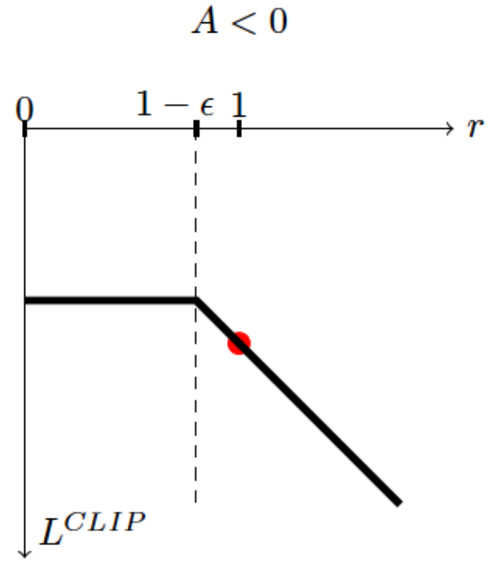
# Proximal Policy Optimization

Can I achieve similar performance without second order information (no Fisher matrix!)

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}$$

$$\max_\theta . \quad L^{CLIP} = \mathbb{E}_t \left[ \min \left( r_t(\theta) A(s_t, a_t), \text{clip} \left( r_t(\theta), 1-\epsilon, 1+\epsilon \right) A(s_t, a_t) \right) \right]$$




J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. "Proximal Policy Optimization Algorithms". (2017)

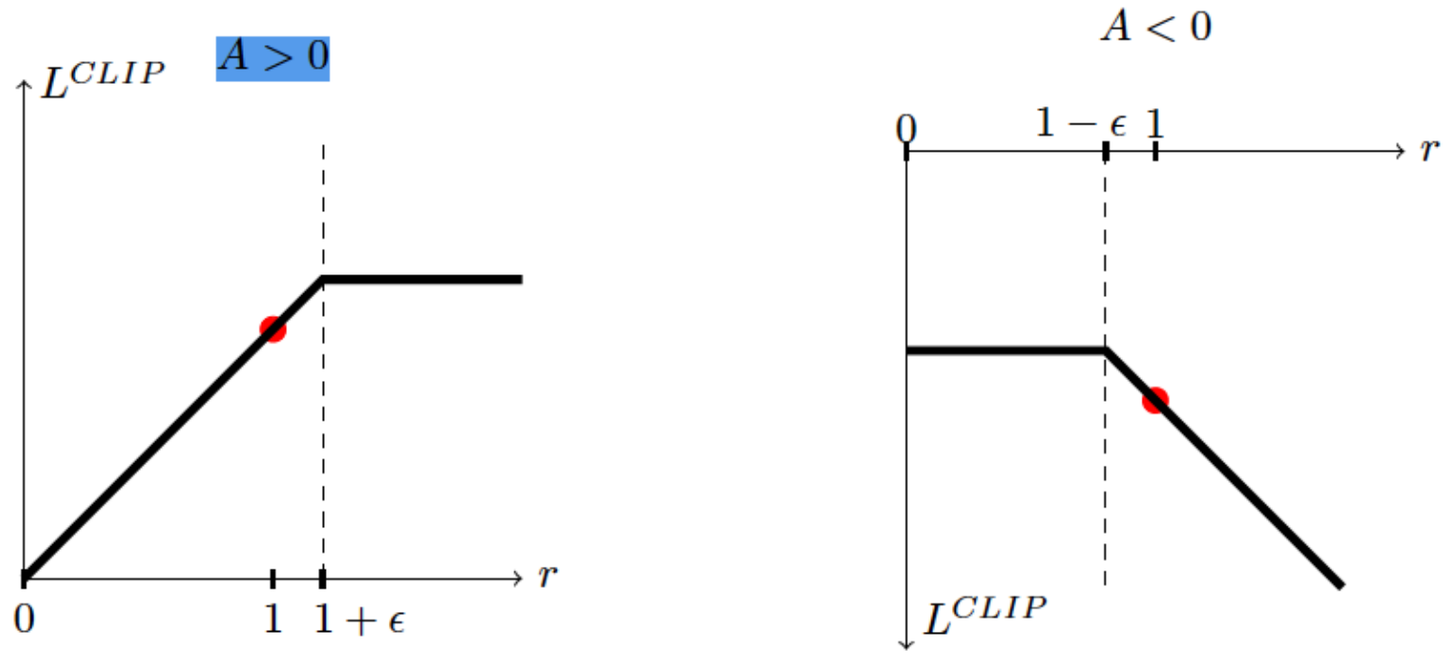
# PPO: Clipped Objective

- Recall the surrogate objective

$$L^{IS}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}_t\left[r_t(\theta)\hat{A}_t\right]. \tag{1}$$

- Form a lower bound via clipped importance ratios

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right] \tag{2}$$




J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. "Proximal Policy Optimization Algorithms". (2017)

# PPO: Clipped Objective

Input: initial policy parameters $\theta_0$, clipping threshold $\epsilon$
**for** $k = 0, 1, 2, \ldots$ **do**
Collect set of partial trajectories $\mathcal{D}_k$ on policy $\pi_k = \pi(\theta_k)$
Estimate advantages $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm
Compute policy update

$$\theta_{k+1} = \arg\max_{\theta} \mathcal{L}_{\theta_k}^{CLIP}(\theta)$$

by taking $K$ steps of minibatch SGD (via Adam), where

$$\mathcal{L}_{\theta_k}^{CLIP}(\theta) = \mathop{\mathrm{E}}_{\tau \sim \pi_k} \left[ \sum_{t=0}^{T} \left[ \min(r_t(\theta)\hat{A}_t^{\pi_k}, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t^{\pi_k}) \right] \right]$$

**end for**

---

- Clipping prevents policy from having incentive to go far away from $\theta_{k+1}$
- Clipping seems to work at least as well as PPO with KL penalty, but is simpler to implement

# PPO: Clipped Objective

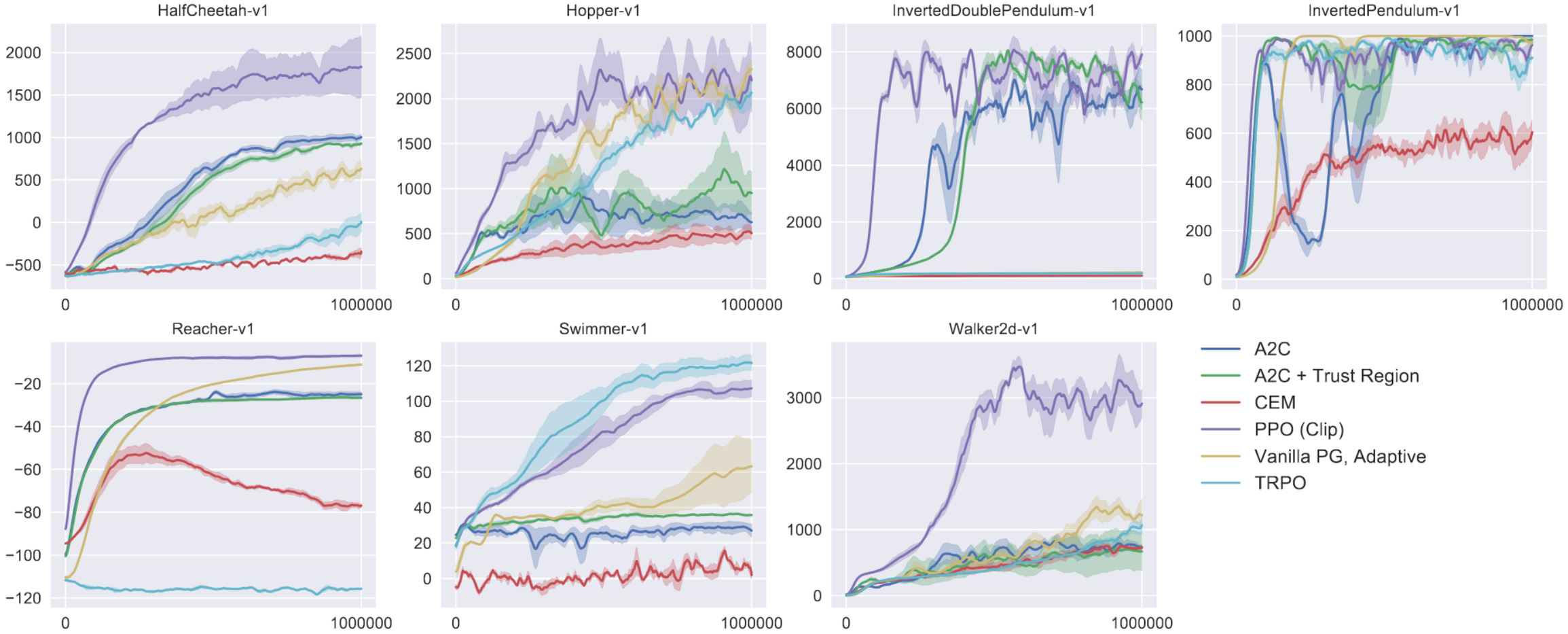



Figure: Performance comparison between PPO with clipped objective and various other deep RL methods on a slate of MuJoCo tasks. [10]

# Summary

- Gradient Descent in Parameter VS distribution space
- Natural gradients: we need to keep track of how the KL changes from iteration to iteration
- Natural policy gradients
- Clipped objective works well