CS4400
DEEP REINFORCEMENT LEARNING

Lecture 6: On-Policy Actor-Critic

Wendelin Böhmer
<j.w.bohmer@tudelft.nl>

TU Delft

5th of December 2023

Content of this lecture

Question: Belief MDP

- POMDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \rho, P, r, O \rangle \quad \to \quad$ Belief-MDP $\langle \mathcal{S}', \mathcal{A}', \rho', P', r' \rangle$
  - $\mathcal{S} = \{1, \ldots, n\}, \quad \mathcal{A} = \{1, \ldots, m\}, \quad \mathcal{O} = \{1, \ldots, k\}$
  - belief state $b_s^t := b(s|\tau_t) := \mathbb{P}(s_t = s | b_0, o_0, a_0, \ldots, o_t)$ at time t
  - belief update $f(s'|\boldsymbol{b}^t, a_t, o_{t+1}) \propto O(o_{t+1}|s') \sum_{s=1}^{n} P(s'|s, a_t)\, b_s^t$

Question: Belief MDP

- POMDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \rho, P, r, O \rangle \quad \rightarrow \quad$ Belief-MDP $\langle \mathcal{S}', \mathcal{A}', \rho', P', r' \rangle$
  - $\mathcal{S} = \{1, \dots, n\}, \quad \mathcal{A} = \{1, \dots, m\}, \quad \mathcal{O} = \{1, \dots, k\}$
  - belief state $b_s^t := b(s|\tau_t) := \mathbb{P}(s_t = s | b_0, o_0, a_0, \dots, o_t)$ at time t
  - belief update $f(s'|\boldsymbol{b}^t, a_t, o_{t+1}) \propto O(o_{t+1}|s') \sum_{s=1}^{n} P(s'|s, a_t)\, b_s^t$
- Define the Belief-MDP from the POMDP:
  - $\mathcal{S}' := \left\{ \boldsymbol{b}^t \,\middle|\, \boldsymbol{b}^t \in \mathbb{R}^n,\ \sum_{s=1}^{n} b_s^t = 1,\ b_s^t \geq 0,\ 1 \leq s \leq n \right\} \qquad \mathcal{A}' := \mathcal{A}$
  - $\rho'(\boldsymbol{b}^0) :=$???
  - $r'(\boldsymbol{b}^t, a_t) :=$???
  - $\mathbb{P}(o_{t+1}|\boldsymbol{b}^t, a_t) =$???
  - $\mathbb{P}(\boldsymbol{b}^{t+1}|\boldsymbol{b}^t, a_t, o_{t+1}) :=$???
  - $P'(\boldsymbol{b}^{t+1}|\boldsymbol{b}^t, a_t) :=$???

Question: Belief MDP

- POMDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \rho, P, r, O \rangle \quad \rightarrow \quad$ Belief-MDP $\langle \mathcal{S}', \mathcal{A}', \rho', P', r' \rangle$
  - $\mathcal{S} = \{1, \ldots, n\}, \quad \mathcal{A} = \{1, \ldots, m\}, \quad \mathcal{O} = \{1, \ldots, k\}$
  - belief state $b_s^t := b(s|\tau_t) := \mathbb{P}(s_t = s | b_0, o_0, a_0, \ldots, o_t)$ at time t
  - belief update $f(s'|\boldsymbol{b}^t, a_t, o_{t+1}) \propto O(o_{t+1}|s') \sum_{s=1}^{n} P(s'|s, a_t)\, b_s^t$
- Define the Belief-MDP from the POMDP:
  - $\mathcal{S}' := \left\{ \boldsymbol{b}^t \,\middle|\, \boldsymbol{b}^t \in \mathbb{R}^n,\ \sum_{s=1}^{n} b_s^t = 1,\ b_s^t \geq 0,\ 1 \leq s \leq n \right\} \qquad \mathcal{A}' := \mathcal{A}$
  - $\rho'(\boldsymbol{b}^0) :=$ 1 iff $\boldsymbol{b}^0 = [\rho(1), \ldots, \rho(n)]^\top$ or $\boldsymbol{b}^0 = \frac{1}{n}\mathbf{1}$
  - $r'(\boldsymbol{b}^t, a_t) :=$???
  - $\mathbb{P}(o_{t+1}|\boldsymbol{b}^t, a_t) =$???
  - $\mathbb{P}(\boldsymbol{b}^{t+1}|\boldsymbol{b}^t, a_t, o_{t+1}) :=$???
  - $P'(\boldsymbol{b}^{t+1}|\boldsymbol{b}^t, a_t) :=$???

Question: Belief MDP

- POMDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \rho, P, r, O \rangle \quad \rightarrow \quad$ Belief-MDP $\langle \mathcal{S}', \mathcal{A}', \rho', P', r' \rangle$
  - $\mathcal{S} = \{1, \ldots, n\}, \quad \mathcal{A} = \{1, \ldots, m\}, \quad \mathcal{O} = \{1, \ldots, k\}$
  - belief state $b_s^t := b(s|\tau_t) := \mathbb{P}(s_t = s | b_0, o_0, a_0, \ldots, o_t)$ at time t
  - belief update $f(s'|\boldsymbol{b}^t, a_t, o_{t+1}) \propto O(o_{t+1}|s') \sum_{s=1}^{n} P(s'|s, a_t)\, b_s^t$
- Define the Belief-MDP from the POMDP:
  - $\mathcal{S}' := \left\{ \boldsymbol{b}^t \,\middle|\, \boldsymbol{b}^t \in \mathbb{R}^n,\ \sum_{s=1}^{n} b_s^t = 1,\ b_s^t \geq 0,\ 1 \leq s \leq n \right\} \qquad \mathcal{A}' := \mathcal{A}$
  - $\rho'(\boldsymbol{b}^0) :=$ 1 iff $\boldsymbol{b}^0 = [\rho(1), \ldots, \rho(n)]^\top$ or $\boldsymbol{b}^0 = \frac{1}{n}\mathbf{1}$
  - $r'(\boldsymbol{b}^t, a_t) := \sum_{s=1}^{n} r(s, a_t)\, b_s^t$
  - $\mathbb{P}(o_{t+1}|\boldsymbol{b}^t, a_t) =$???
  - $\mathbb{P}(\boldsymbol{b}^{t+1}|\boldsymbol{b}^t, a_t, o_{t+1}) :=$???
  - $P'(\boldsymbol{b}^{t+1}|\boldsymbol{b}^t, a_t) :=$???

Question: Belief MDP

- POMDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \rho, P, r, O \rangle \quad \rightarrow \quad$ Belief-MDP $\langle \mathcal{S}', \mathcal{A}', \rho', P', r' \rangle$
  - $\mathcal{S} = \{1, \ldots, n\}, \quad \mathcal{A} = \{1, \ldots, m\}, \quad \mathcal{O} = \{1, \ldots, k\}$
  - belief state $b_s^t := b(s|\tau_t) := \mathbb{P}(s_t = s | b_0, o_0, a_0, \ldots, o_t)$ at time t
  - belief update $f(s'|\boldsymbol{b}^t, a_t, o_{t+1}) \propto O(o_{t+1}|s') \sum\limits_{s=1}^{n} P(s'|s, a_t)\, b_s^t$
- Define the Belief-MDP from the POMDP:
  - $\mathcal{S}' := \Big\{ \boldsymbol{b}^t \,\Big|\, \boldsymbol{b}^t \in \mathbb{R}^n,\ \sum\limits_{s=1}^{n} b_s^t = 1,\ b_s^t \geq 0,\ 1 \leq s \leq n \Big\} \qquad \mathcal{A}' := \mathcal{A}$
  - $\rho'(\boldsymbol{b}^0) :=$ 1 iff $\boldsymbol{b}^0 = [\rho(1), \ldots, \rho(n)]^\top$ or $\boldsymbol{b}^0 = \frac{1}{n}\mathbf{1}$
  - $r'(\boldsymbol{b}^t, a_t) := \sum\limits_{s=1}^{n} r(s, a_t)\, b_s^t$
  - $\mathbb{P}(o_{t+1}|\boldsymbol{b}^t, a_t) = \sum\limits_{s'=1}^{n} O(o_{t+1}|s') \sum\limits_{s=1}^{n} P(s'|s, a_t)\, b_s^t$
  - $\mathbb{P}(\boldsymbol{b}^{t+1}|\boldsymbol{b}^t, a_t, o_{t+1}) :=$???
  - $P'(\boldsymbol{b}^{t+1}|\boldsymbol{b}^t, a_t) :=$???

Question: Belief MDP

- POMDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \rho, P, r, O \rangle \quad \rightarrow \quad$ Belief-MDP $\langle \mathcal{S}', \mathcal{A}', \rho', P', r' \rangle$
  - $\mathcal{S} = \{1, \ldots, n\}, \quad \mathcal{A} = \{1, \ldots, m\}, \quad \mathcal{O} = \{1, \ldots, k\}$
  - belief state $b_s^t := b(s|\tau_t) := \mathbb{P}(s_t = s | b_0, o_0, a_0, \ldots, o_t)$ at time t
  - belief update $f(s'|\boldsymbol{b}^t, a_t, o_{t+1}) \propto O(o_{t+1}|s') \sum_{s=1}^{n} P(s'|s, a_t)\, b_s^t$
- Define the Belief-MDP from the POMDP:
  - $\mathcal{S}' := \left\{ \boldsymbol{b}^t \,\middle|\, \boldsymbol{b}^t \in \mathbb{R}^n,\ \sum_{s=1}^{n} b_s^t = 1,\ b_s^t \geq 0,\ 1 \leq s \leq n \right\} \qquad \mathcal{A}' := \mathcal{A}$
  - $\rho'(\boldsymbol{b}^0) :=$ 1 iff $\boldsymbol{b}^0 = [\rho(1), \ldots, \rho(n)]^\top$ or $\boldsymbol{b}^0 = \frac{1}{n}\mathbf{1}$
  - $r'(\boldsymbol{b}^t, a_t) := \sum_{s=1}^{n} r(s, a_t)\, b_s^t$
  - $\mathbb{P}(o_{t+1}|\boldsymbol{b}^t, a_t) = \sum_{s'=1}^{n} O(o_{t+1}|s') \sum_{s=1}^{n} P(s'|s, a_t)\, b_s^t$
  - $\mathbb{P}(\boldsymbol{b}^{t+1}|\boldsymbol{b}^t, a_t, o_{t+1}) :=$ 1 iff $\boldsymbol{b}_{t+1} = \boldsymbol{f}(\boldsymbol{b}^t, a_t, o_{t+1})$
  - $P'(\boldsymbol{b}^{t+1}|\boldsymbol{b}^t, a_t) :=$???

Question: Belief MDP

- POMDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \rho, P, r, O \rangle \quad \rightarrow \quad$ Belief-MDP $\langle \mathcal{S}', \mathcal{A}', \rho', P', r' \rangle$
  - $\mathcal{S} = \{1, \ldots, n\}, \quad \mathcal{A} = \{1, \ldots, m\}, \quad \mathcal{O} = \{1, \ldots, k\}$
  - belief state $b^t_s := b(s|\tau_t) := \mathbb{P}(s_t = s | b_0, o_0, a_0, \ldots, o_t)$ at time t
  - belief update $f(s'|\boldsymbol{b}^t, a_t, o_{t+1}) \propto O(o_{t+1}|s') \sum\limits_{s=1}^{n} P(s'|s, a_t)\, b^t_s$
- Define the Belief-MDP from the POMDP:
  - $\mathcal{S}' := \left\{ \boldsymbol{b}^t \,\middle|\, \boldsymbol{b}^t \in \mathbb{R}^n,\ \sum\limits_{s=1}^{n} b^t_s = 1,\ b^t_s \geq 0,\ 1 \leq s \leq n \right\} \qquad \mathcal{A}' := \mathcal{A}$
  - $\rho'(\boldsymbol{b}^0) := 1 \quad$ iff $\quad \boldsymbol{b}^0 = [\rho(1), \ldots, \rho(n)]^\top \quad$ or $\quad \boldsymbol{b}^0 = \frac{1}{n}\mathbf{1}$
  - $r'(\boldsymbol{b}^t, a_t) := \sum\limits_{s=1}^{n} r(s, a_t)\, b^t_s$
  - $\mathbb{P}(o_{t+1}|\boldsymbol{b}^t, a_t) = \sum\limits_{s'=1}^{n} O(o_{t+1}|s') \sum\limits_{s=1}^{n} P(s'|s, a_t)\, b^t_s$
  - $\mathbb{P}(\boldsymbol{b}^{t+1}|\boldsymbol{b}^t, a_t, o_{t+1}) := 1 \quad$ iff $\quad \boldsymbol{b}_{t+1} = \boldsymbol{f}(\boldsymbol{b}^t, a_t, o_{t+1})$
  - $P'(\boldsymbol{b}^{t+1}|\boldsymbol{b}^t, a_t) := \sum\limits_{o'=1}^{k} \mathbb{P}(\boldsymbol{b}^{t+1}|\boldsymbol{b}^t, a_t, o')\, \mathbb{P}(o'|\boldsymbol{b}^t, a_t)$

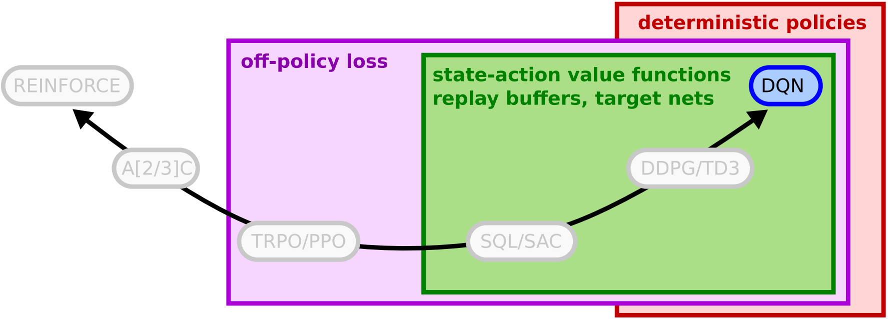
6.1

On-Policy Actor-Critic
Stochastic policy gradients

TU Delft

6.1 Deep RL algorithms so far

- So far we only looked at deep Q-learning (and extensions)
- Maximization in DQN requires discrete actions
- Is there another way?



core concept: RL without value functions

- $\max_\theta J[\pi_\theta] = \max_\theta \mathbb{E}\Big[R_0 \,\Big|\, \substack{s_0 \sim \rho(\cdot),\; s_{t+1} \sim P(\cdot|s_t, a_t) \\ a_t \sim \pi_\theta(\cdot|s_t),\; r_t = r(s_t, a_t)}\Big], \qquad R_t := \sum_{k=0}^{\infty} \gamma^k r_{t+k}$

core concept: RL without value functions

- $\max_\theta J[\pi_\theta] = \max_\theta \mathbb{E}\Big[R_0 \,\Big|\, \begin{smallmatrix} s_0 \sim \rho(\cdot),\; s_{t+1} \sim P(\cdot|s_t,a_t) \\ a_t \sim \pi_\theta(\cdot|s_t),\; r_t = r(s_t,a_t) \end{smallmatrix}\Big], \qquad R_t := \sum_{k=0}^{\infty} \gamma^k r_{t+k}$

$$\nabla_\theta J[\pi_\theta] \quad = \quad \nabla_\theta \iint \rho(s_0)\, \pi_\theta(a_0|s_0)\, \mathbb{E}_{\pi_\theta}\big[R_0 \big|{}^{s_0}_{a_0}\big]\, ds_0\, da_0$$

core concept: RL without value functions

- $\max\limits_{\theta} J[\pi_\theta] = \max\limits_{\theta} \mathbb{E}\Big[R_0 \,\Big|\, \substack{s_0 \sim \rho(\cdot),\; s_{t+1} \sim P(\cdot|s_t, a_t) \\ a_t \sim \pi_\theta(\cdot|s_t),\; r_t = r(s_t, a_t)} \Big] \,, \qquad R_t := \sum\limits_{k=0}^{\infty} \gamma^k r_{t+k}$

$$
\begin{aligned}
\nabla_\theta J[\pi_\theta] \quad &= \quad \nabla_\theta \iint \rho(s_0)\, \pi_\theta(a_0|s_0)\, \mathbb{E}_{\pi_\theta}\big[R_0 \big|{}^{s_0}_{a_0}\big]\, ds_0\, da_0 \\
&\;\;\vdots \\
&= \quad \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t R_t \,\nabla_\theta \ln \pi_\theta(a_t|s_t) \,\Big|\, \substack{s_0 \sim \rho(\cdot),\; s_{t+1} \sim P(\cdot|s_t, a_t) \\ a_t \sim \pi_\theta(\cdot|s_t),\; r_t = r(s_t, a_t)} \Big]
\end{aligned}
$$

you will prove the *policy gradient theorem* (Sutton et al., 1999) as homework,

core concept: RL without value functions

- $\max_\theta J[\pi_\theta] = \max_\theta \mathbb{E}\Big[R_0 \,\Big|\, \substack{s_0 \sim \rho(\cdot),\, s_{t+1} \sim P(\cdot|s_t,a_t) \\ a_t \sim \pi_\theta(\cdot|s_t),\, r_t = r(s_t,a_t)}\Big]\,, \qquad R_t := \sum_{k=0}^{\infty} \gamma^k r_{t+k}$

$$\begin{aligned} \nabla_\theta J[\pi_\theta] &= \nabla_\theta \iint \rho(s_0)\, \pi_\theta(a_0|s_0)\, \mathbb{E}_{\pi_\theta}\big[R_0 \big|^{s_0}_{a_0}\big]\, ds_0\, da_0 \\ &\;\vdots \\ &= \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t R_t\, \nabla_\theta \ln \pi_\theta(a_t|s_t) \,\Big|\, \substack{s_0 \sim \rho(\cdot),\, s_{t+1} \sim P(\cdot|s_t,a_t) \\ a_t \sim \pi_\theta(\cdot|s_t),\, r_t = r(s_t,a_t)}\Big] \end{aligned}$$

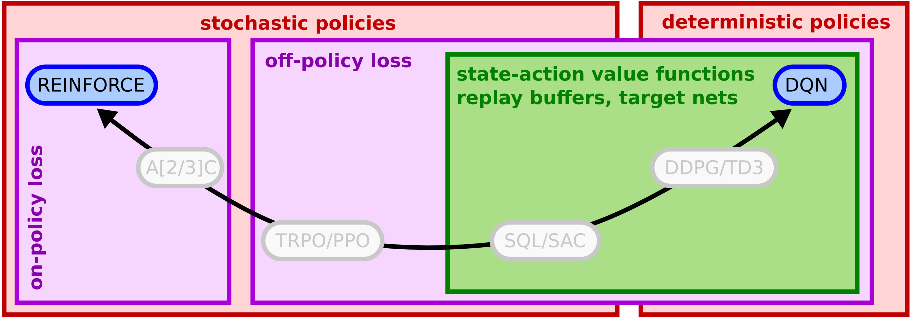
- REINFORCE estimates expectation with m **rollouts** of π_θ
  - no gradient flow through samples R_t^i of return

$$\nabla_\theta \mathcal{L}_\pi[\theta] := -\mathbb{E}_\pi\Big[\sum_{t=0}^{\infty} \gamma^t R_t \nabla_\theta \ln \pi_\theta(a_t|s_t)\Big] \approx \nabla_\theta - \frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{n-1} \gamma^t R_t^i \ln \pi_\theta(a_t^i|s_t^i)$$

you will prove the *policy gradient theorem* (Sutton et al., 1999) as homework, REINFORCE algorithm by (Williams, 1992)

6.1 REINFORCE vs. DQN

- Opposites on the on-policy off-policy spectrum
  - DQN: sample efficient vs. complex Q-value
    - when value functions generalize well
  - REINFORCE no value function vs. sample inefficient
    - when value functions do not generalize



e.g. Kool et al. (2019) use REINFORCE to solve traveling salesman problems

6.1 Actor-critic algorithms

- Sample m episodes containing n transitions $\langle s_t^i, a_t^i, R_t^i \rangle$ with π_θ

$$\mathcal{L}_\pi[\theta] \quad \approx \quad -\frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{n-1} \gamma^t \ln \pi_\theta(a_t^i | s_t^i) \;\; R_t^i$$

- Sampled returns $R_t^i = \sum_{j=0}^{\infty} \gamma^j r_{t+j}^i$ have high variance

Actor-critic approach originally by Sutton et al. (1999), see Grondman et al. (2012) for an overview
see `spinningup.openai.com/en/latest/algorithms/vpg.html` for a version that drops γ^t

6.1 Actor-critic algorithms

- Sample m episodes containing n transitions $\langle s_t^i, a_t^i, R_t^i \rangle$ with π_θ

$$\mathcal{L}'_\pi[\theta] \quad := \quad -\frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{n-1} \gamma^t \ln \pi_\theta(a_t^i | s_t^i) \left(R_t^i - v_\phi(s_t^i) \right)$$

- Sampled returns $R_t^i = \sum_{j=0}^{\infty} \gamma^j r_{t+j}^i$ have high variance

- Variance reduced by subtracting a bias $v_\phi(s_t^i) \approx V^{\pi_\theta}(s_t^i)$
  - bias free estimate: no change in gradient

$$\int \pi_\theta(a|s) \, \nabla_\theta \ln \pi_\theta(a|s) \, da = \int \pi_\theta(a|s) \, \frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)} \, da = \nabla_\theta \int \pi_\theta(a|s) \, da = 0$$

Actor-critic approach originally by Sutton et al. (1999), see Grondman et al. (2012) for an overview
see `spinningup.openai.com/en/latest/algorithms/vpg.html` for a version that drops γ^t

6.1 Actor-critic algorithms

- Sample m episodes containing n transitions $\langle s_t^i, a_t^i, r_t^i, s_{t+1}^i \rangle$ with π_θ

$$\mathcal{L}''_\pi[\theta] \quad := \quad -\frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{n-1} \gamma^t \ln \pi_\theta(a_t^i | s_t^i) \underbrace{\left(r_t^i + \gamma v_\phi(s_{t+1}^i) - v_\phi(s_t^i) \right)}_{\text{Advantage } A_\phi(s_t^i, r_t^i, s_{t+1}^i)}$$

- Sampled returns $R_t^i = \sum_{j=0}^{\infty} \gamma^j r_{t+j}^i$ have high variance
- Variance reduced by subtracting a bias $v_\phi(s_t^i) \approx V^{\pi_\theta}(s_t^i)$
  - bias free estimate: no change in gradient

$$\int \pi_\theta(a|s) \, \nabla_\theta \ln \pi_\theta(a|s) \, da = \int \pi_\theta(a|s) \, \frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)} \, da = \nabla_\theta \int \pi_\theta(a|s) \, da = 0$$

- *Advantage Actor-Critic* (A2C) reduce var. further by *bootstrapping*
  - replaces return R_{t+1}^i with $v_\phi(s_{t+1}^i) \approx V^{\pi_\theta}(s_{t+1}^i) = \mathbb{E}_{\pi_\theta}[R_{t+1}^i | s_{t+1}^i]$
  - approximated values introduce bias!

Actor-critic approach originally by Sutton et al. (1999), see Grondman et al. (2012) for an overview
see `spinningup.openai.com/en/latest/algorithms/vpg.html` for a version that drops γ^t

6.1 On-policy value approximation

- On-policy values $V^\pi(s)$ can be approximated in a variety of ways
  - based on sampled trajectory $\tau_\infty = \{s_t, a_t, r_t\}_{t=0}^\infty$

MSE loss: $$\mathcal{L}[\phi] \quad := \quad \sum_{t=0}^{\infty} \Big(v_\phi(s_t) - \underbrace{y_t(\tau_\infty)}_{\text{targets}} \Big)^2$$

assignment sheet 3

image source: publicdomainvectors.org

6.1 On-policy value approximation

- On-policy values $V^\pi(s)$ can be approximated in a variety of ways
  - based on sampled trajectory $\tau_\infty = \{s_t, a_t, r_t\}_{t=0}^\infty$

MSE loss: $$\mathcal{L}[\phi] \quad := \quad \sum_{t=0}^{\infty} \Big(v_\phi(s_t) - \underbrace{y_t(\tau_\infty)}_{\text{targets}} \Big)^2$$

- n-step targets $y_t^n(\tau_n) \;:=\; \sum_{k=0}^{n-1} \gamma^k r_{t+k} + \gamma^n \, v_{\phi'}(s_{t+n})$
  - looks farther into the future of given trajectory
  - in expectation $V^\pi(s_t) = \mathbb{E}_\pi[y_t^n(\tau_\infty)], \forall n \in \mathbb{N}$

assignment sheet 3

image source: publicdomainvectors.org

6.1 On-policy value approximation

- On-policy values $V^\pi(s)$ can be approximated in a variety of ways
  - based on sampled trajectory $\tau_\infty = \{s_t, a_t, r_t\}_{t=0}^\infty$

MSE loss: $$\mathcal{L}[\phi] \quad := \quad \sum_{t=0}^{\infty} \Big(v_\phi(s_t) - \underbrace{y_t(\tau_\infty)}_{\text{targets}} \Big)^2$$

- n-step targets $y_t^n(\tau_n) \; := \; \sum_{k=0}^{n-1} \gamma^k r_{t+k} + \gamma^n \, v_{\phi'}(s_{t+n})$
  - looks farther into the future of given trajectory
  - in expectation $V^\pi(s_t) = \mathbb{E}_\pi[y_t^n(\tau_\infty)], \forall n \in \mathbb{N}$
- Monte-Carlo targets $y_t^{\text{MC}}(\tau_\infty) := \lim\limits_{n\to\infty} y_t^n(\tau_\infty)$
  - fast but high variance

assignment sheet 3

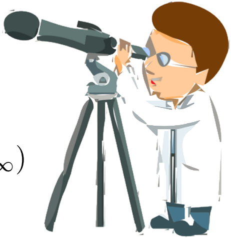
image source: publicdomainvectors.org

6.1 On-policy value approximation

- On-policy values $V^\pi(s)$ can be approximated in a variety of ways
  - based on sampled trajectory $\tau_\infty = \{s_t, a_t, r_t\}_{t=0}^\infty$

MSE loss: $$\mathcal{L}[\phi] \quad := \quad \sum_{t=0}^{\infty} \Big(v_\phi(s_t) - \underbrace{y_t(\tau_\infty)}_{\text{targets}} \Big)^2$$

- n-step targets $y_t^n(\tau_n) \;:=\; \sum_{k=0}^{n-1} \gamma^k r_{t+k} + \gamma^n \, v_{\phi'}(s_{t+n})$
  - looks farther into the future of given trajectory
  - in expectation $V^\pi(s_t) = \mathbb{E}_\pi[y_t^n(\tau_\infty)], \forall n \in \mathbb{N}$
- Monte-Carlo targets $y_t^{\text{MC}}(\tau_\infty) := \lim_{n\to\infty} y_t^n(\tau_\infty)$
  - fast but high variance
- Eligibility traces $y_t^\lambda(\tau_\infty) \;:=\; (1-\lambda) \sum_{n=0}^{\infty} \lambda^n \, y_t^{n+1}(\tau_\infty)$
  - $\lambda = 0 \Rightarrow y_t^\lambda(\tau_\infty) = y_t^1(\tau_\infty)$, slow, low variance
  - $\lambda = 1 \Rightarrow y_t^\lambda(\tau_\infty) = y_t^{\text{MC}}(\tau_\infty)$, fast, high variance

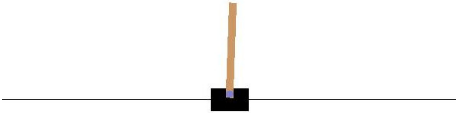


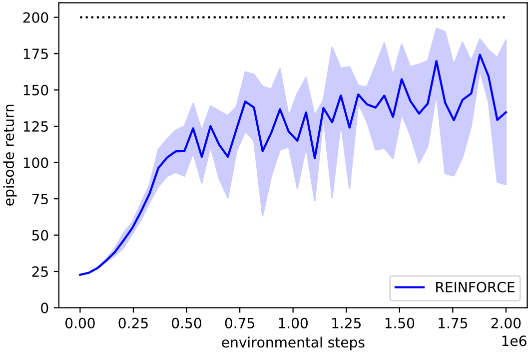
assignment sheet 3

image source: publicdomainvectors.org

6.1 Effect of bias and bootsstrap

- Example: `Cartpole-V0`
  - value and policy heads
- REINFORCE
  - $\mathcal{L}_\pi$ very unstable



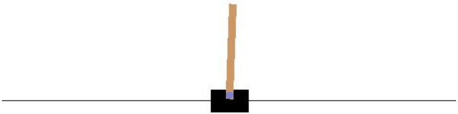


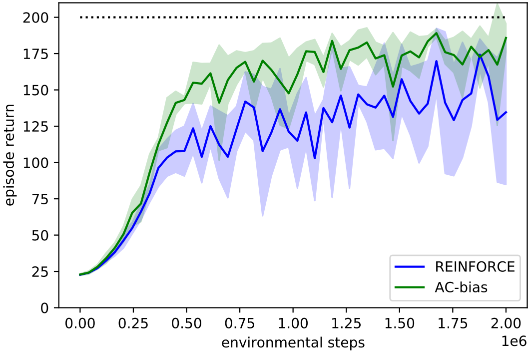
assignment sheet 3

mean and standard deviation over 5 seeds

Effect of bias and bootsstrap

- Example: `Cartpole-V0`
  - value and policy heads
- REINFORCE
  - $\mathcal{L}_\pi$ very unstable
- Value estimate as bias
  - $\mathcal{L}'_\pi$ more stable



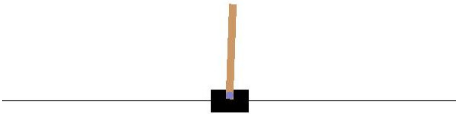


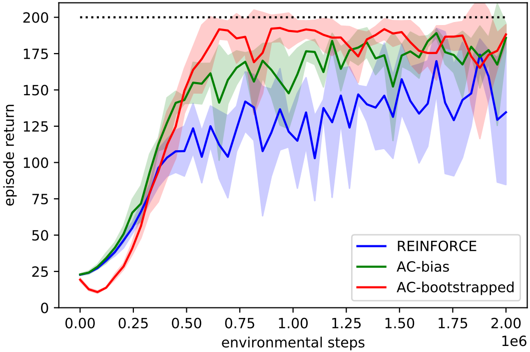
assignment sheet 3

mean and standard deviation over 5 seeds

6.1 Effect of bias and bootsstrap

- Example: `Cartpole-V0`
  - value and policy heads
- REINFORCE
  - $\mathcal{L}_\pi$ very unstable
- Value estimate as bias
  - $\mathcal{L}'_\pi$ more stable
- TD-error as advantage
  - $\mathcal{L}''_\pi$ fast and stable
  - initial performance dip
- Biggest flaws of A2C:
  - sample efficiency
  - stability



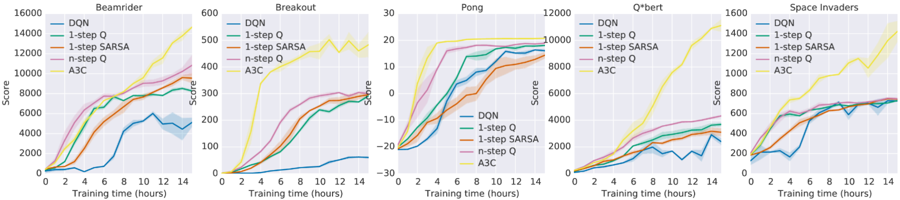


assignment sheet 3

mean and standard deviation over 5 seeds

Asynchronous Advantage Actor-Critic

- Bootstrapping with $y_t := r_t + \gamma v_\phi(s_{t+1})$ is often too slow
  - n-step bootstrapping $y_t^n := \sum\limits_{k=0}^{n-1} \gamma^k r_{t+k} + \gamma^n v_\phi(s_{t+n})$
  - TD(λ) bootstrapping $y_t^\lambda = (1-\lambda) \sum\limits_{n=0}^{\infty} \lambda^n \, y_t^n$
- Transitions in episodes not i.i.d.
  - run multiple environments in parallel
  - asynchronous updates can be scaled to large clusters



A3C and A2C (ATARI results from Mnih et al., 2016) are the most widely used stochastic actor-critic algorithms

6.1 Convergence of actor-critic methods

- Actor-critic methods converge to a local maximum if:
  1. the MDP is *finite*, *ergodic* and has *bounded reward*
  2. actor and critic are *linear* with suitable basis-functions
  3. the *learning rate* of the actor is *lower* than the critic's
  4. the critic is trained by TD(λ) with sufficiently *large* λ

- No practical guarantees for deep AC methods, but:
  - relative learning rates are important
  - bootstrapping must propagate future reward fast

detailed assumptions and convergence proofs in Konda and Tsitsiklis (2003, Theorem 3.4 and Theorem 6.3)

core concept: RL with continuous actions

- Actor-critic methods can use continuous actions $\boldsymbol{a} \in \mathcal{A} \subset \mathbb{R}^m$
  - no explicit maximization
  - value function independent of actions

another way to learn distributions, the *reparametrization trick*, will be introduced in Lecture 7

core concept: RL with continuous actions

- Actor-critic methods can use continuous actions $\boldsymbol{a} \in \mathcal{A} \subset \mathbb{R}^m$
  - no explicit maximization
  - value function independent of actions
- Policy network output parameterizes distribution
  - e.g. diagonal Gaussian: $\pi_\theta(\boldsymbol{a}|s) \propto \exp\left(-\sum_{i=0}^{m} \frac{(a_i - \mu_\theta(s)_i)^2}{2\sigma_\theta^2(s)_i}\right)$
  - neural network with m heads for $\boldsymbol{\mu}_\theta(s)$ and m heads for $\boldsymbol{\sigma}_\theta(s)$

another way to learn distributions, the *reparametrization trick*, will be introduced in Lecture 7

core concept: RL with continuous actions

- Actor-critic methods can use continuous actions $\boldsymbol{a} \in \mathcal{A} \subset \mathbb{R}^m$
  - no explicit maximization
  - value function independent of actions
- Policy network output parameterizes distribution
  - e.g. diagonal Gaussian: $\pi_\theta(\boldsymbol{a}|s) \propto \exp\Big(-\sum_{i=0}^{m} \frac{(a_i - \mu_\theta(s)_i)^2}{2\sigma_\theta^2(s)_i}\Big)$
  - neural network with m heads for $\boldsymbol{\mu}_\theta(s)$ and m heads for $\boldsymbol{\sigma}_\theta(s)$
- Exploration samples actions $\boldsymbol{a}_t$ from π_θ
  - maximum entropy regularization

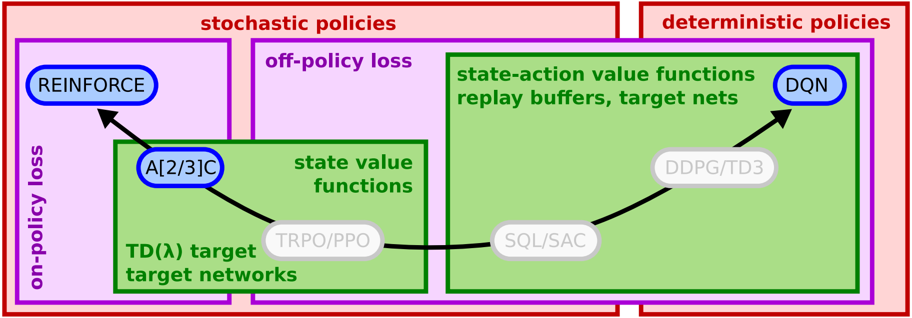
$$\bar{\mathcal{L}}[\theta] := \mathcal{L}[\theta] - \tfrac{1}{\beta}\mathcal{H}[\pi_\theta]\,, \quad \mathcal{H}[\pi_\theta] := -\tfrac{1}{n}\sum_{t=1}^{n-1} \int \pi_\theta(\boldsymbol{a}|s_t)\, \ln \pi_\theta(\boldsymbol{a}|s_t)\, d\boldsymbol{a}$$

$$\nabla_\theta \mathcal{H}[\pi_\theta] = -\tfrac{1}{n}\sum_{t=0}^{n-1} \int \pi_\theta(\boldsymbol{a}|s_t)\, \ln \pi_\theta(\boldsymbol{a}|s_t)\, \nabla_\theta \ln \pi_\theta(\boldsymbol{a}|s_t)\, d\boldsymbol{a}$$

another way to learn distributions, the *reparametrization trick*, will be introduced in Lecture 7

6.1 Actor-critics in comparison

- Advantage actor-critic algorithms (A[2/3]C) extend REINFORCE
  - more sample efficient than REINFORCE due to value function
  - does not replay samples ⇒ *less efficient* than DQN
  - on-policy and state-value ⇒ *more stable* than DQN



6.1 Summary

- REINFORCE learns policy from many rollouts
- Actor-critic algorithms replace rollouts with advantage
- On-policy value estimation can use targets that look farther
- Actor-critcs converge under some specific conditions
- Neural net outputs distribution for continuous actions

Learning Objectives

LO6.1: Derive REINFORCE and the stochastic actor-critic algorithm
LO6.2: Implement and test discussed actor-critic variants
LO6.3: Derive and implement on-policy value estimation with various targets

6.2

On-Policy Actor-Critic
Trust region methods

core concept: Off-policy gradients I

- Stochastic policy gradients requires *on-policy* samples
  - $\xi_t^\pi(s)$: state distribution after t steps with policy π

$$\begin{aligned}
\nabla_\theta \mathcal{L}_\pi[\theta] &= -\mathbb{E}_\pi\Big[\sum_{t=0}^{n-1} \gamma^t R_t \, \nabla_\theta \ln \pi_\theta(a_t|s_t)\Big] \\
&= -\sum_{t=0}^{n-1} \iint \xi_t^\pi(s_t)\, \pi_\theta(a_t|s_t)\, \gamma^t\, \mathbb{E}_\pi\big[R_t|s_t, a_t\big] \, \frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)} \, ds_t \, da_t
\end{aligned}$$

Off-policy actor critic by Degris et al. (2012), with emphatic importance sampling in Sutton et al. (2016) and Zhang et al. (2019)

core concept: Off-policy gradients I

- Stochastic policy gradients requires *on-policy* samples
  - $\xi_t^\pi(s)$: state distribution after t steps with policy π

$$
\begin{aligned}
\nabla_\theta \mathcal{L}_\pi[\theta] &= -\mathbb{E}_\pi\Big[\sum_{t=0}^{n-1} \gamma^t R_t \,\nabla_\theta \ln \pi_\theta(a_t|s_t)\Big] \\
&= -\sum_{t=0}^{n-1} \iint \xi_t^\pi(s_t)\, \pi_\theta(a_t|s_t)\, \gamma^t\, \mathbb{E}_\pi\big[R_t|s_t, a_t\big]\, \frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)}\, ds_t\, da_t \\
&= -\sum_{t=0}^{n-1} \iint \xi_t^\pi(s_t)\, \pi_\theta(a_t|s_t)\, \gamma^t Q^\pi(s_t, a_t)\, \frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)}\, ds_t\, da_t
\end{aligned}
$$

Off-policy actor critic by Degris et al. (2012), with emphatic importance sampling in Sutton et al. (2016) and Zhang et al. (2019)

core concept: Off-policy gradients I

- Stochastic policy gradients requires *on-policy* samples
  - $\xi_t^\pi(s)$: state distribution after t steps with policy π
  - $\mu(a|s)$: new sampling policy with $\frac{\pi_\theta(a|s)}{\mu(a|s)} < \infty, \forall a \in \mathcal{A}, \forall s \in \mathcal{S}$

$$
\begin{aligned}
\nabla_\theta \mathcal{L}_\pi[\theta] &= -\mathbb{E}_\pi\Big[\sum_{t=0}^{n-1} \gamma^t R_t \, \nabla_\theta \ln \pi_\theta(a_t|s_t)\Big] \\
&= -\sum_{t=0}^{n-1} \iint \xi_t^\pi(s_t)\, \pi_\theta(a_t|s_t)\, \gamma^t\, \mathbb{E}_\pi\big[R_t|s_t,a_t\big]\, \frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)}\, ds_t\, da_t \\
&= -\sum_{t=0}^{n-1} \iint \xi_t^\pi(s_t)\, \pi_\theta(a_t|s_t)\, \gamma^t Q^\pi(s_t,a_t)\, \frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)}\, ds_t\, da_t \\
&= -\sum_{t=0}^{n-1} \iint \xi_t^\pi(s_t)\, \mu(a_t|s_t)\, \gamma^t Q^\pi(s_t,a_t)\, \frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\mu(a_t|s_t)}\, ds_t\, da_t
\end{aligned}
$$

Off-policy actor critic by Degris et al. (2012), with emphatic importance sampling in Sutton et al. (2016) and Zhang et al. (2019)

core concept: Off-policy gradients II

$$\nabla_\theta \mathcal{L}_\pi[\theta] \;=\; -\sum_{t=0}^{n-1} \iint \xi_t^\pi(s_t)\, \mu(a_t|s_t)\, \gamma^t Q^\pi(s_t, a_t)\, \frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\mu(a_t|s_t)}\, ds_t\, da_t$$

Off-policy actor critic by Degris et al. (2012), with emphatic importance sampling in Sutton et al. (2016) and Zhang et al. (2019)

core concept: Off-policy gradients II

- Off-policy gradients optimizes policy π_θ on samples of μ
  - requires Q-value function Q^π of π_θ

$$\nabla_\theta \mathcal{L}_\pi[\theta] \;=\; -\sum_{t=0}^{n-1} \iint \xi_t^\pi(s_t)\, \mu(a_t|s_t)\, \gamma^t Q^\pi(s_t, a_t)\, \frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\mu(a_t|s_t)}\, ds_t\, da_t$$

$$= \; -\nabla_\theta\, \mathbb{E}_\mu\Big[\sum_{t=0}^{n-1} \frac{\xi_t^\pi(s_t)}{\xi_t^\mu(s_t)}\, \gamma^t Q^\pi(s_t, a_t)\, \frac{\pi_\theta(a_t|s_t)}{\mu(a_t|s_t)}\Big]$$

Off-policy actor critic by Degris et al. (2012), with emphatic importance sampling in Sutton et al. (2016) and Zhang et al. (2019)

core concept: Off-policy gradients II

- Off-policy gradients optimizes policy π_θ on samples of μ
  - requires Q-value function Q^π (or value V^π) of π_θ

$$
\begin{aligned}
\nabla_\theta \mathcal{L}_\pi[\theta] &= -\sum_{t=0}^{n-1} \iint \xi_t^\pi(s_t)\, \mu(a_t|s_t)\, \gamma^t Q^\pi(s_t, a_t)\, \frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\mu(a_t|s_t)}\, ds_t\, da_t \\
&= -\nabla_\theta\, \mathbb{E}_\mu\Big[\sum_{t=0}^{n-1} \frac{\xi_t^\pi(s_t)}{\xi_t^\mu(s_t)}\, \gamma^t Q^\pi(s_t, a_t)\, \frac{\pi_\theta(a_t|s_t)}{\mu(a_t|s_t)}\Big] \\
&= -\nabla_\theta\, \mathbb{E}_\mu\Big[\sum_{t=0}^{n-1} \frac{\xi_t^\pi(s_t)}{\xi_t^\mu(s_t)} \gamma^t \big(\underbrace{r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)}_{A(s_t, r_t, s_{t+1})}\big) \frac{\pi_\theta(a_t|s_t)}{\mu(a_t|s_t)}\Big]
\end{aligned}
$$

Off-policy actor critic by Degris et al. (2012), with emphatic importance sampling in Sutton et al. (2016) and Zhang et al. (2019)

core concept: Off-policy gradients II

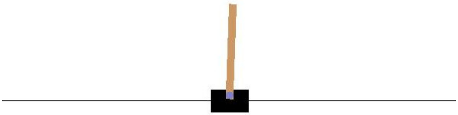
- Off-policy gradients optimizes policy π_θ on samples of μ
  - requires Q-value function Q^π (or value V^π) of π_θ
  - importance sampling $\frac{\xi_t^\pi(s_t)}{\xi_t^\mu(s_t)}$ and discount γ^t often ignored
  - on-policy value v_ϕ approximates V^μ, not V^π

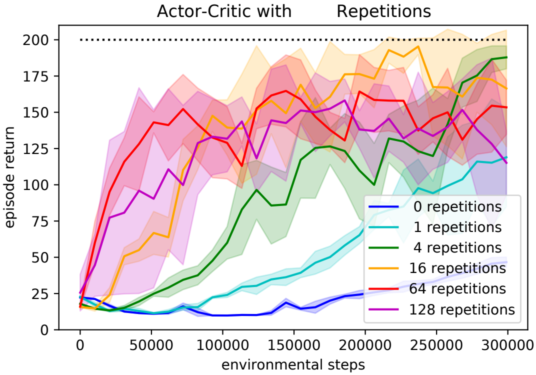
$$
\begin{aligned}
\nabla_\theta \mathcal{L}_\pi[\theta] &= -\sum_{t=0}^{n-1} \iint \xi_t^\pi(s_t)\, \mu(a_t|s_t)\, \gamma^t Q^\pi(s_t, a_t)\, \frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\mu(a_t|s_t)}\, ds_t\, da_t \\
&= -\nabla_\theta\, \mathbb{E}_\mu\Big[\sum_{t=0}^{n-1} \frac{\xi_t^\pi(s_t)}{\xi_t^\mu(s_t)}\, \gamma^t Q^\pi(s_t, a_t)\, \frac{\pi_\theta(a_t|s_t)}{\mu(a_t|s_t)}\Big] \\
&= -\nabla_\theta\, \mathbb{E}_\mu\Big[\sum_{t=0}^{n-1} \frac{\xi_t^\pi(s_t)}{\xi_t^\mu(s_t)} \gamma^t \Big(\underbrace{r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)}_{A(s_t, r_t, s_{t+1})}\Big) \frac{\pi_\theta(a_t|s_t)}{\mu(a_t|s_t)}\Big] \\
&\approx -\nabla_\theta\, \mathbb{E}_\mu\Big[\sum_{t=0}^{n-1} \gamma^t \big(r_t + \gamma v_\phi(s_{t+1}) - v_\phi(s_t)\big) \frac{\pi_\theta(a_t|s_t)}{\mu(a_t|s_t)}\Big] \quad =: \; \nabla_\theta \mathcal{L}_\mu[\theta]
\end{aligned}
$$

Off-policy actor critic by Degris et al. (2012), with emphatic importance sampling in Sutton et al. (2016) and Zhang et al. (2019)

6.2 Off-policy sample efficiency

- Example `Cartpole-v0`
  - sample $n = 2048$ steps
  - distributed over 4 envs
- Repeat `train()`
  - same mini-batch
  - off-policy loss $\mathcal{L}_\mu$
  - $\text{TD}(1)$ value loss
- Accelerated learning
- Unstable repetitions
  - why?





assignment sheet 3

mean and standard deviation over 5 seeds

6.2 Trust-region policy optimization (TRPO)

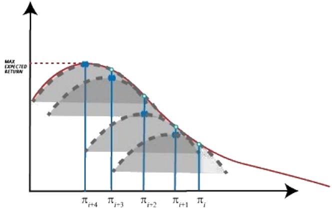
- Changing the policy π_θ also changes the state distributions ξ_t^π
  - $\frac{\xi_t^\pi(s_t)}{\xi_t^\mu(s_t)}$ requires $\xi_t^\mu(s) > 0, \forall s \in \{s \,|\, \xi_t^\pi(s) > 0\} \subseteq \mathcal{S}$
  - but many states π_θ would sample are not in the batch

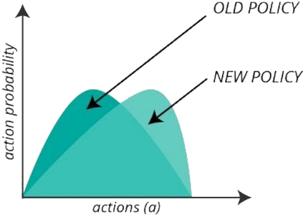
spinningup.openai.com/en/latest/algorithms/trpo.html,

6.2 Trust-region policy optimization (TRPO)

- Changing the policy π_θ also changes the state distributions ξ_t^π
  - $\frac{\xi_t^\pi(s_t)}{\xi_t^\mu(s_t)}$ requires $\xi_t^\mu(s) > 0, \forall s \in \{s \,|\, \xi_t^\pi(s) > 0\} \subseteq \mathcal{S}$
  - but many states π_θ would sample are not in the batch
- Keep new policy π_θ in a *trust region* around old $\mu = \pi_{\theta'}$

$$\min_\theta \mathcal{L}_\mu[\theta] \quad \text{s.t.} \quad \mathbb{E}_\mu\Big[\sum_{t=0}^{n-1} D_{\text{KL}}[\mu(\cdot|s_t) \| \pi_\theta(\cdot|s_t)]\Big] \leq \delta$$





spinningup.openai.com/en/latest/algorithms/trpo.html, image from this explanatory youtube video

6.2 Trust-region policy optimization (TRPO)

- Changing the policy π_θ also changes the state distributions ξ_t^π
  - $\frac{\xi_t^\pi(s_t)}{\xi_t^\mu(s_t)}$ requires $\xi_t^\mu(s) > 0, \forall s \in \{s \,|\, \xi_t^\pi(s) > 0\} \subseteq \mathcal{S}$
  - but many states π_θ would sample are not in the batch
- Keep new policy π_θ in a *trust region* around old $\mu = \pi_{\theta'}$

$$\min_\theta \mathcal{L}_\mu[\theta] \quad \text{s.t.} \quad \mathbb{E}_\mu\Big[\sum_{t=0}^{n-1} D_{\mathsf{KL}}[\mu(\cdot|s_t) \| \pi_\theta(\cdot|s_t)]\Big] \leq \delta$$

- Taylor approximation around θ_μ of loss and constraint
  - for gradient $\boldsymbol{g} := \nabla_\theta \mathcal{L}_\mu[\theta]\big|_{\theta=\theta_\mu}$ and Hessian matrix $\mathbf{H}$
  - $\mathcal{L}_\mu[\theta] \approx \boldsymbol{g}^\top(\theta - \theta_\mu)$ and $D_{\mathsf{KL}}[\mu \| \pi_\theta] \approx \frac{1}{2}(\theta - \theta_\mu)^\top \mathbf{H}(\theta - \theta_\mu)$
  - solution $\theta^* = \theta_\mu + \alpha\sqrt{\frac{2\delta}{\boldsymbol{g}^\top \mathbf{H}^{-1}\boldsymbol{g}}}\mathbf{H}^{-1}\boldsymbol{g}$ called *natural policy gradient*
  - TRPO: natural policy gradient with *line search* over α

Natural Policy Gradient by Kakade (2002), Trust Region Policy Optimization by Schulman et al. (2015)

6.2 Proximal policy optimization (PPO)

- Constrains in TRPO are hard to implement/optimize

$$\min_\theta \quad \mathcal{L}_\mu[\theta] \;=\; -\mathbb{E}_\mu\Big[\sum_{t=0}^{n-1} \gamma^t \big(\overbrace{r_t + \gamma v_\phi(s_{t+1}) - v_\phi(s_t)}^{A_t}\big) \overbrace{\frac{\pi_\theta(a_t|s_t)}{\mu(a_t|s_t)}}^{\text{ratio}}\Big]$$

$$\text{s.t.} \qquad \mathbb{E}_\mu\Big[\sum_{t=0}^{n-1} D_{\mathsf{KL}}[\mu(\cdot|s_t)\|\pi_\theta(\cdot|s_t)]\Big] \leq \delta$$

spinningup.openai.com/en/latest/algorithms/ppo.html

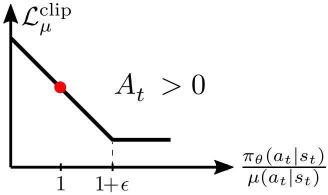
Proximal policy optimization (PPO)

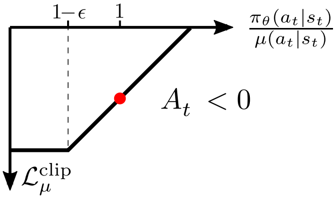
- Constrains in TRPO are hard to implement/optimize

$$\min_{\theta} \quad \mathcal{L}_\mu[\theta] \;=\; -\mathbb{E}_\mu\Big[\sum_{t=0}^{n-1} \gamma^t \big(\overbrace{r_t + \gamma v_\phi(s_{t+1}) - v_\phi(s_t)}^{A_t}\big) \overbrace{\frac{\pi_\theta(a_t|s_t)}{\mu(a_t|s_t)}}^{\text{ratio}}\Big]$$

$$\text{s.t.} \qquad \mathbb{E}_\mu\Big[\sum_{t=0}^{n-1} D_{\mathsf{KL}}[\mu(\cdot|s_t)\|\pi_\theta(\cdot|s_t)]\Big] \leq \delta$$

- PPO: "clip" the policy ratio outside $1 \pm \epsilon$
  - only clip when the loss improves





PPO by Schulman et al. (2017); read this `article` by Jonathan Hui for more information on PPO and TRPO

6.2 Proximal policy optimization (PPO)

- Constrains in TRPO are hard to implement/optimize

$$\begin{aligned}\min_\theta \quad \mathcal{L}_\mu[\theta] &= -\mathbb{E}_\mu\Big[\sum_{t=0}^{n-1}\gamma^t\big(\overbrace{r_t+\gamma v_\phi(s_{t+1})-v_\phi(s_t)}^{A_t}\big)\overbrace{\tfrac{\pi_\theta(a_t|s_t)}{\mu(a_t|s_t)}}^{\text{ratio}}\Big]\\ \text{s.t.} \qquad & \mathbb{E}_\mu\Big[\sum_{t=0}^{n-1}D_{\mathsf{KL}}[\mu(\cdot|s_t)\|\pi_\theta(\cdot|s_t)]\Big] \leq \delta\end{aligned}$$

- PPO: "clip" the policy ratio outside $1 \pm \epsilon$
  - only clip when the loss improves
  - $\mathrm{clip}(x,a,b) = \min(\max(x,a),b)$

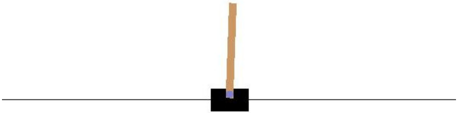
$$\mathcal{L}^{\text{clip}}_\mu[\theta] \;:=\; -\mathbb{E}_\mu\Big[\sum_{t=0}^{n-1}\gamma^t\min\Big(A_t\,\tfrac{\pi_\theta(a_t|s_t)}{\mu(a_t|s_t)},\, A_t\,\mathrm{clip}\big(\tfrac{\pi_\theta(a_t|s_t)}{\mu(a_t|s_t)},1-\epsilon,1+\epsilon\big)\Big)\Big]$$

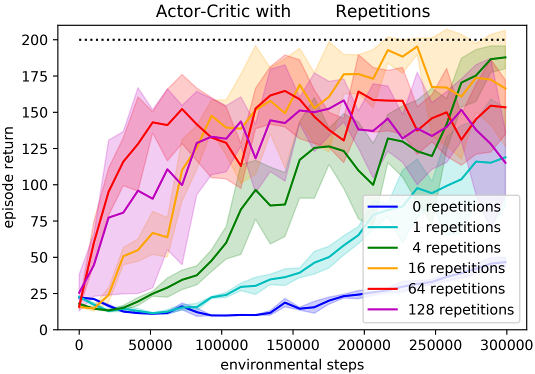
- Clipped values cut the gradient flow
  - "removes" samples with too much divergence

PPO by Schulman et al. (2017); read this `article` by Jonathan Hui for more information on PPO and TRPO

6.2 Effect of PPO clipping

- Example `Cartpole-v0`
  - sample $n = 2048$ steps
- Repeat `train()`
  - same as slide 16



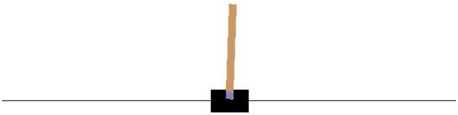


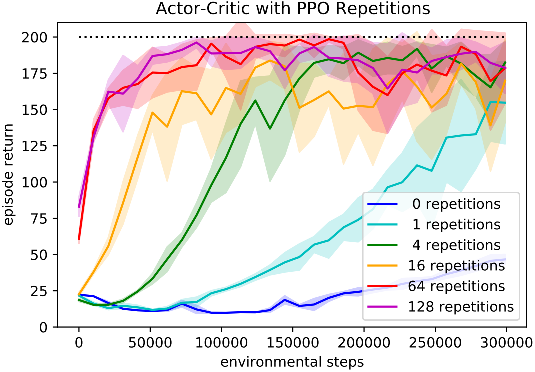
assignment sheet 3

mean and standard deviation over 5 seeds

Effect of PPO clipping

- Example `Cartpole-v0`
  - sample $n = 2048$ steps
- Repeat `train()`
  - same as slide 16
  - clipping at $\epsilon = 0.1$
- Accelerated learning
  - very popular in practice
- Improved stability
  - unstable for large $|\mathcal{A}|$



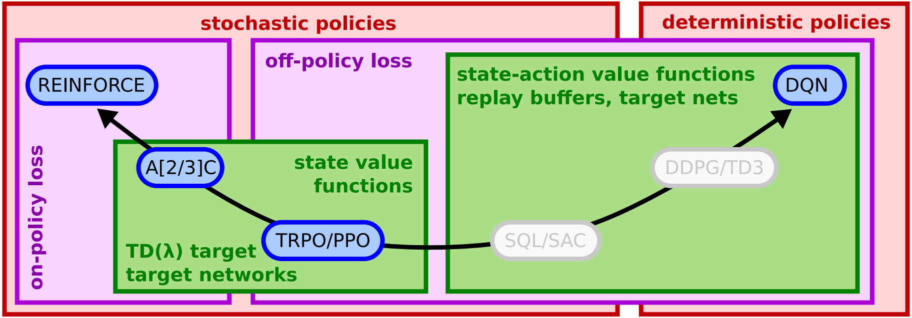


mean and standard deviation over 5 seeds

assignment sheet 3

6.2 TRPO/PPO in comparison

- TRPO and PPO are A[2/3]C with stabilized off-policy optimization
  - more efficient on-policy sampling ⇒ still *less efficient* than DQN
  - on-policy and state-value ⇒ *more stable* than DQN
  - used when state-values are more reliable than state-action values



This Blog compares PPO implementation details. PPO is used extensively in robotics (e.g. Serra-Gómez et al., 2023)

6.2 Application example: PPO for ChatGPT

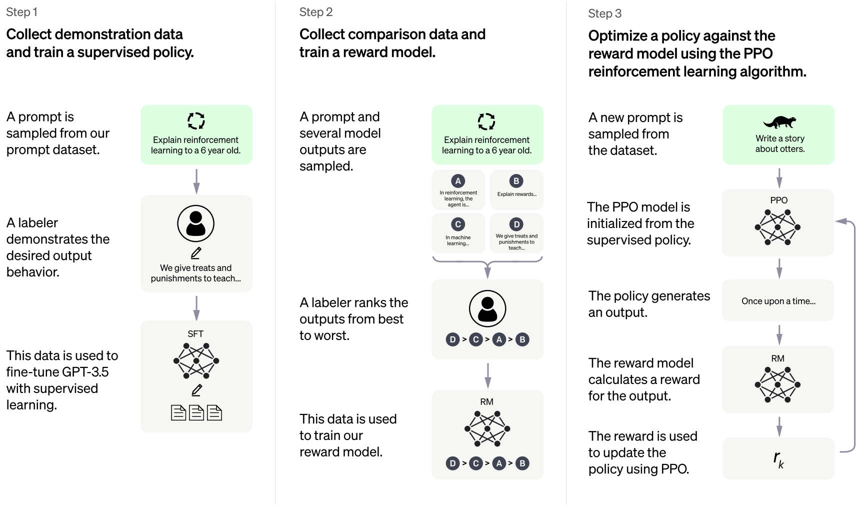


image from OpenAI blog, for GPT-4 see OpenAI (2023), reward is learned from ordered examples (Christiano et al., 2017)

6.2 Summary

- On-policy actor-critics can be approximated with off-policy data
- Unstable due to state-distribution shift
- TRPO constrains how much the policy can shift
- PPO implements this by clipping policy ratios
- Combine *on-policy* sampling with *off-policy* optimization

Learning Objectives

LO6.4: Explain off-policy gradients, TRPO and PPO
LO6.5: Implement and test off-policy gradients and PPO

Next lecture

- Next lecture: **policy gradients with replay buffers**!
- This Thursday is **tutorial**, please submit **assignment 2**!
- Questions? Ask them here:
  `answers.ewi.tudelft.nl`

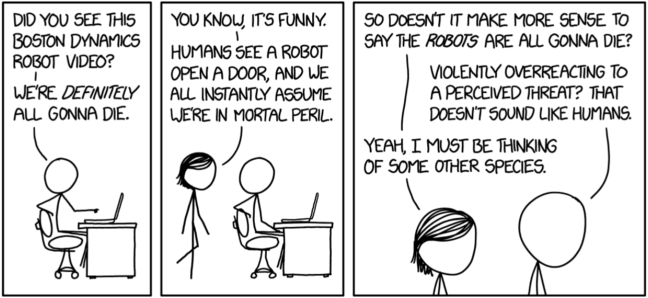


image source: xkcd.com

References I

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.03741.

Thomas Degris, Martha White, and Richard S. Sutton. Off-policy actor-critic. In *Proceedings of the 29th International Coference on International Conference on Machine Learning (ICML)*, pages 179–186, 2012. URL https://arxiv.org/abs/1205.4839.

Ivo Grondman, Lucian Busoniu, Gabriel A. D. Lopes, and Robert Babuska. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1291–1307, 2012. doi: 10.1109/TSMCC.2012.2218595. URL https://hal.archives-ouvertes.fr/hal-00756747/document.

Sham M Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 14, 2002. URL https://proceedings.neurips.cc/paper/2001/file/4b86abe48d358ecf194c56c69108433e-Paper.pdf.

Vijay R. Konda and John N. Tsitsiklis. On actor-critic algorithms. *SIAM Journal on Control and Optimization*, 42(4): 1143–1166, 2003. URL http://web.mit.edu/people/jnt/Papers/J094-03-kon-actors.pdf.

Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *7th International Conference on Learning Representations, ICLR*, 2019. URL https://arxiv.org/abs/1803.08475.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937. PMLR, 2016. URL https://proceedings.mlr.press/v48/mniha16.html.

OpenAI. Gpt-4 technical report, 2023. URL https://arxiv.org/abs/2303.08774.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of Machine Learning Research (ICML)*, volume 37, pages 1889–1897, 2015. URL http://proceedings.mlr.press/v37/schulman15.html.

References II

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Álvaro Serra-Gómez, Eduardo Montijano, Wendelin Böhmer, and Javier Alonso-Mora. Active classification of moving targets with learned control policies. *IEEE Robotics and Automation Letters*, 8(6):3717–3724, 2023. ISSN 2377-3766. doi: 10.1109/LRA.2023.3271508. URL https://arxiv.org/abs/2212.03068.

Richard S. Sutton, David McAllester, and Satinder Singh anf Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 12, pages 1057–1063. MIT Press, 1999. URL https://papers.nips.cc/paper/1999/hash/464d828b85b0bed98e80ade0a5c43b0f-Abstract.html.

Richard S. Sutton, A. Rupam Mahmood, and Martha White. An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of Machine Learning Research*, 17(1):2603–2631, jan 2016. ISSN 1532-4435. URL https://arxiv.org/abs/1503.04269.

Ronald J. Williams. Simple statistical gradient algorithms for connectionist reinforcement learning. *Machine Learning*, 8: 229–256, 1992. URL https://link.springer.com/article/10.1007/BF00992696.

Shangtong Zhang, Wendelin Böhmer, and Shimon Whiteson. Generalized off-policy actor-critic. In *Advances in Neural Information Processing Systems (NeurIPS) 32*, pages 2001–2011. 2019. URL http://papers.nips.cc/paper/8474-generalized-off-policy-actor-critic.pdf.