Fraunhofer IIS

Fraunhofer-Institut für Integrierte Schaltungen IIS

Reinforcement Learning

—

Exercise 6: Policy Approximation

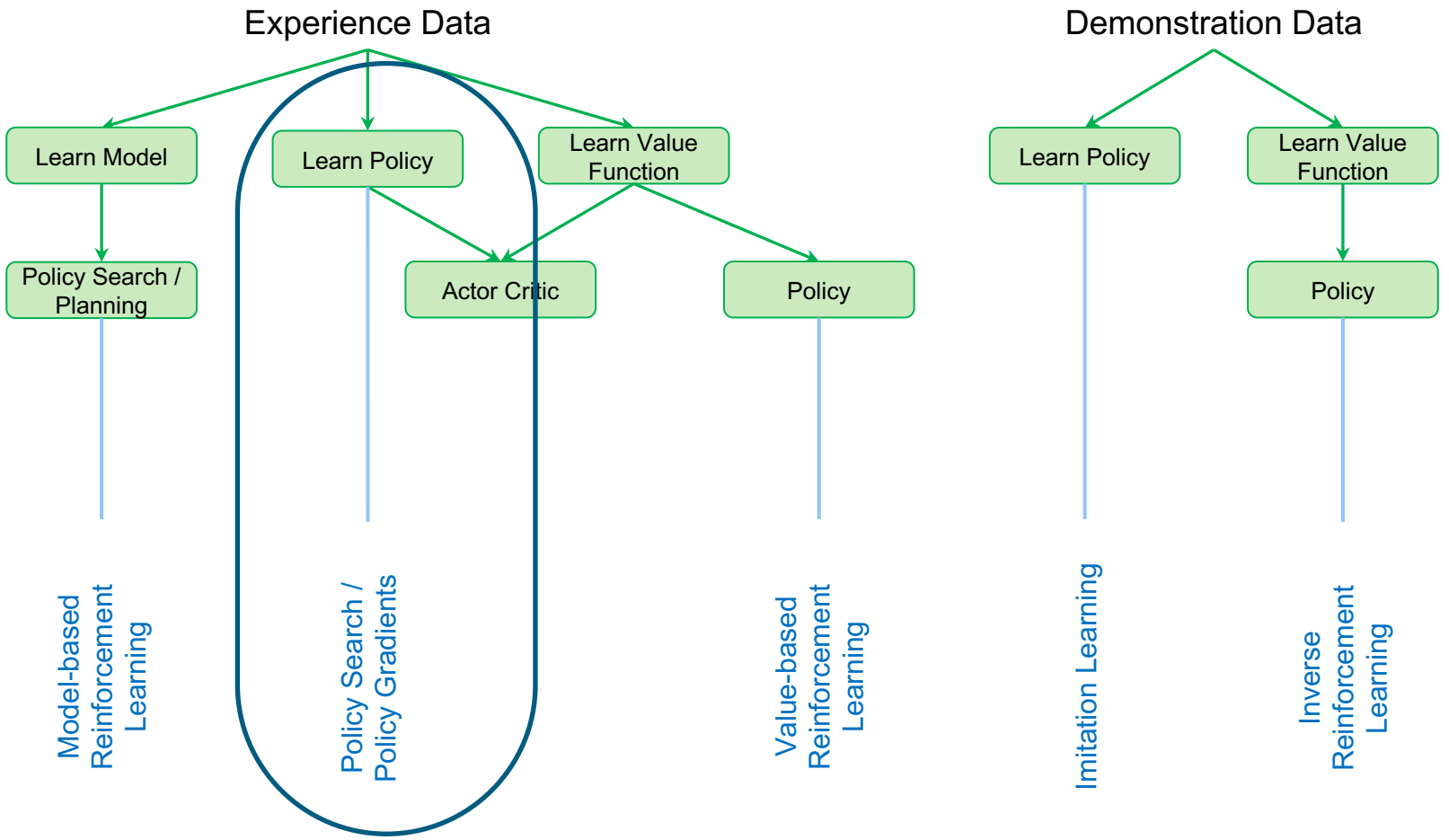
02.06.2023

Nico Meyer

Exercise Sheet 6

Value Function Approximation

Policy-based Reinforcement Learning

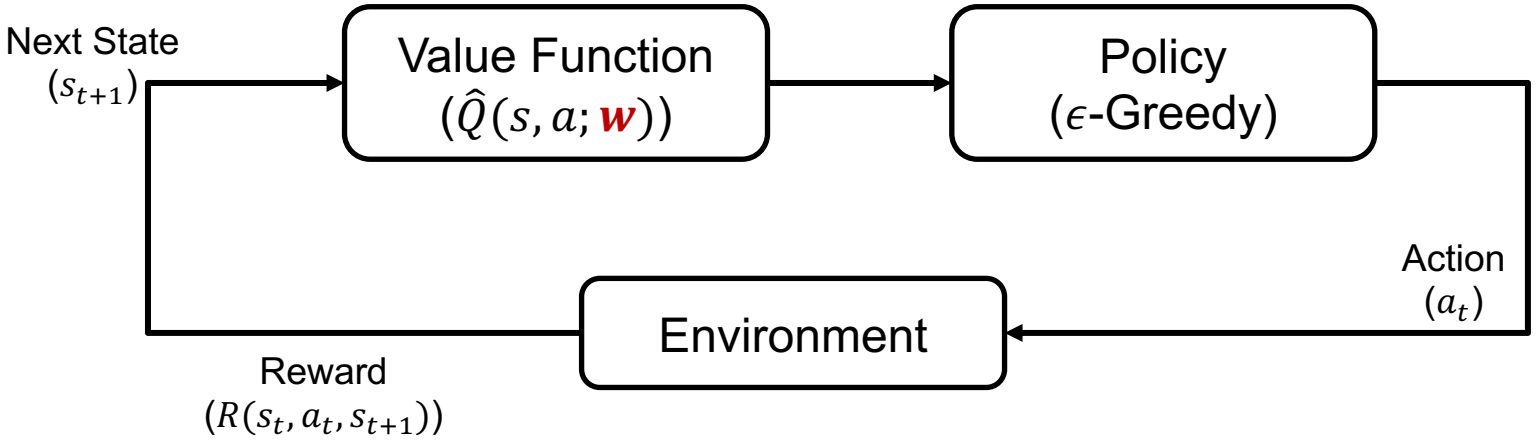


Fraunhofer IIS

Policy-based Reinforcement Learning

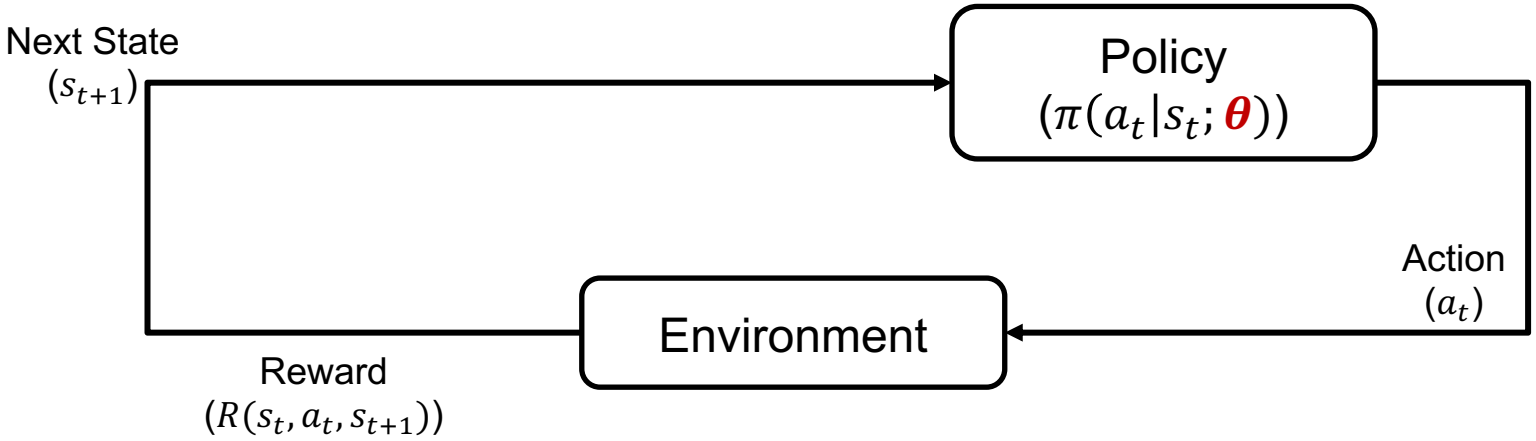
Change of pipeline

So far:



Goal: find w that approximates the true Q-function

Now:



Goal: find θ that maximizes long term reward

Policy-based Reinforcement Learning

Advantages and Disadvantages

Advantages:

- **Good convergence properties**
- Easily extended to high-dimensional or continuous state and action spaces
- Can learn ***stochastic*** **policies**
- Sometimes policies are simple while values and models are complex
  - e.g., rich domain, but optimal is always to go left

Disadvantages:

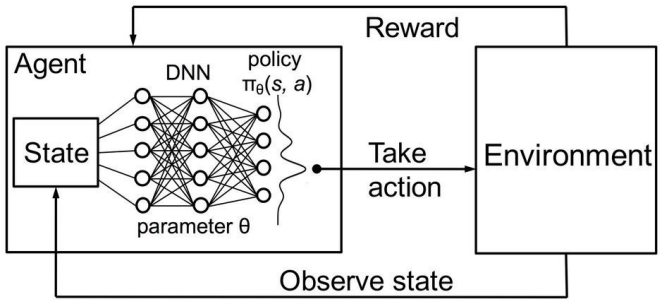
- Susceptible to local optima (especially with non-linear FA)
- Obtained knowledge is specific, does not always generalize well
- Ignores a lot of information in the data (when used in isolation)

Policy-based Reinforcement Learning

Stochastic policies

We have seen deterministic policies like this:

State gives $Q(s, a; w)$ and we selected $\pi(a|s)$ by $\text{argmax}_a\, Q(s, a; w)$

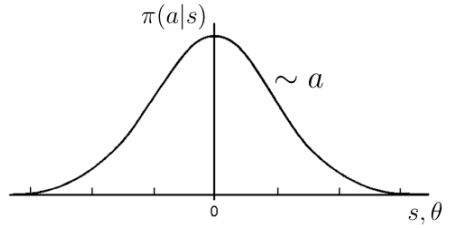


https://towardsdatascience.com/self-learning-ai-agents-iv-stochastic-policy-gradients-b53f088fce20

Instead, stochastic policies do something like this:

$$\pi(a|s) = \mathbb{P}[a|s; \theta]$$

(policy is represented as a probability distribution)



- optimal policy might be stochastic

Policy-based Reinforcement Learning

Optimizing via gradient ascent

- Our goal is to maximize the expected reward:

$$G(\tau) := \sum_{t=0}^{T-1} \gamma^t R(s_t, a_t)$$

$$\max_{\theta} \mathbb{E}_{\pi_\theta} G(\tau)$$

(where π_θ is a parameterized policy, e.g., a neural network)

- But how do we maximize this?

  → Gradient Ascent! *Suppose* we know how to calculate the gradient w.r.t. the parameters:

$$\nabla_\theta \mathbb{E}_{\tau \sim \pi_\theta} G(\tau)$$

Policy Gradient

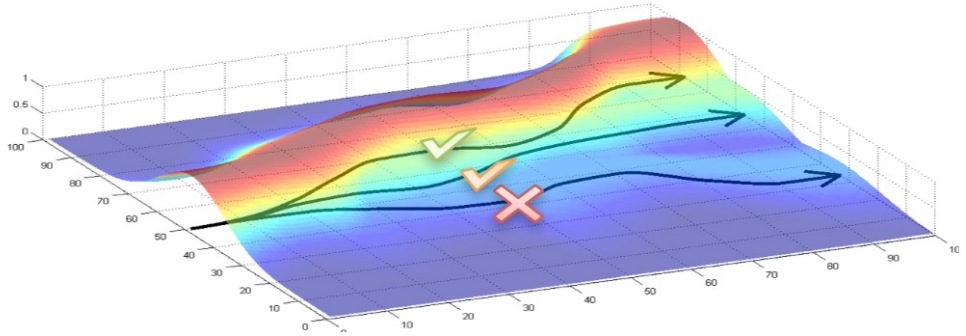
often in literature referred to as $\nabla_\theta J(\pi_\theta)$

- Then we can update our parameters θ in the direction of the gradient:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \mathbb{E}_{\tau \sim \pi_\theta} G(\tau)$$

Policy-based Reinforcement Learning

REINFORCE



Pieter Abbeel. DeepRL Bootcamp 4A Policy Gradients.

REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s,\boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot,\boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ (G_t)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t,\boldsymbol{\theta})$

Intuition: The gradient tries to

- Increase probability of paths with positive G
- Decrease probability of paths with negative G

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Exercise Sheet 7.1

Vanilla Policy Gradient (VPG)

Actor Critic Approaches

Introduce critic that estimates Q

- The policy gradient we used so far (without baseline to begin with):

$$\nabla_\theta \mathbb{E}_{\pi_\theta} G(\tau) \approx \frac{1}{L} \sum_{\tau} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)\, G(\tau)$$

$$\approx \frac{1}{L} \sum_{\tau} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \underbrace{\sum_{t'=t}^{T} \gamma^{t'-t} R(s_{t'}, a_{t'})}_{= Q^\pi(s_t, a_t)}$$

➢ Use e.g. a neural network to approximate Q!

- In practice: estimate $v^\pi(s_t; \phi)$ explicitly, and then sample

$$q^\pi(s_t, a_t) \approx G_t^{(n)}$$

i.e. $\hat{G}_t^{(1)} = R_t + \gamma v^\pi(s_{t+1}; \phi)$

Actor Critic Approaches

Advantage Actor Critic (A2C)

- Introduce a baseline:

$$\nabla_\theta \mathbb{E}_{\pi_\theta} G(\tau) \approx \frac{1}{L} \sum_{\tau} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \underbrace{\sum_{t'=t}^{T} \gamma^{t'-t} R(s_{t'}, a_{t'})}_{= Q^\pi(s_t, a_t)} - b(s_t)$$

$$= \frac{1}{L} \sum_{\tau} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \underbrace{(\hat{G}_t - b(s_t))}_{:= A^\pi(s_t, a_t)}$$

- Calculate via TD error:

$$A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t)$$
$$= r + \gamma \cdot v^\pi - v^\pi(s_t)$$

- Or multi-step TD error: "Generalized Advantage Estimation (GAE)"

Exercise Sheet 7.2

Advantage Actor Critic (A2C)

Fraunhofer
IIS

Fraunhofer-Institut für Integrierte Schaltungen IIS

Thank you for your attention!