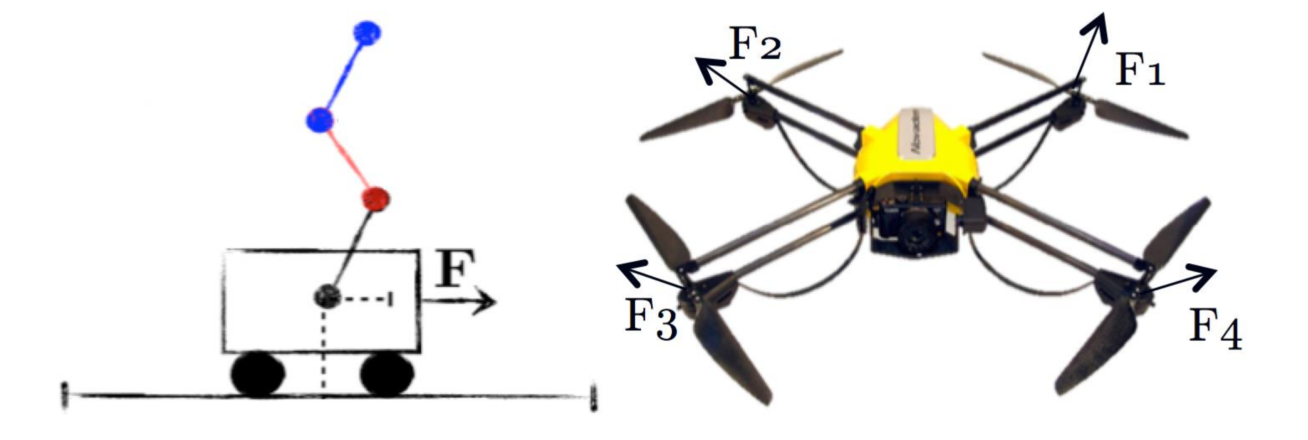
Chapter 06. 스스로 전략을 짜는 강화학습 (Reinforcement Learning)

Policy Gradient



Multi-Task Policy Gradient Methods for Control

Value Based RL

http://www.modulabs.co.kr/RL_library/3305

* **Value-based reinforcement learning** vs Policy-based reinforcement learning

- In the last lecture we approximated the value or action-value function using parameters θ,

$$\boxed{\begin{aligned} V_\theta(s) &\approx V^\pi(s) \\ Q_\theta(s,a) &\approx Q^\pi(s,a) \end{aligned}}$$

- A policy was generated directly from the value function
  - e.g. using ϵ-greedy

Policy Based RL

http://www.modulabs.co.kr/RL_library/3305

* Value-based reinforcement learning vs **Policy-based reinforcement learning**

- In this lecture we will directly parametrise the policy

$$\pi_\theta(s, a) = \mathbb{P}\left[a \mid s, \theta\right]$$

- We will focus again on model-free reinforcement learning

Policy Gradient

http://www.modulabs.co.kr/RL_library/3305

Advantages:

- Better convergence properties
- Effective in high-dimensional or continuous action spaces
- Can learn stochastic policies

Disadvantages:

- Typically converge to a local rather than global optimum
- Evaluating a policy is typically inefficient and high variance

Value Based RL

http://www.modulabs.co.kr/RL_library/3305

Unstable Policy

- **Greedy** updates

$$\theta_{\pi'} = arg\max_{\theta} \mathbb{E}_{\pi_\theta}[Q^\pi(s,a)]$$

- $V^{\pi_0} \xrightarrow{\text{small change}} \pi_1 \xrightarrow{\text{large change}} V^{\pi_1} \xrightarrow{\text{large change}} \pi_2 \xrightarrow{\text{large change}}$
- Potentially **unstable** learning process with **large policy jumps**
- **Policy Gradient** updates

$$\theta_{\pi'} = \theta_\pi + \alpha \left.\frac{dJ(\theta)}{d\theta}\right|_{\theta=\theta^\pi}$$

- $V^{\pi_0} \xrightarrow{\text{small change}} \pi_1 \xrightarrow{\text{small change}} V^{\pi_1} \xrightarrow{\text{small change}} \pi_2 \xrightarrow{\text{small change}}$
- **Stable** learning process with **smooth policy improvement**

fast campus

Value Based RL

http://www.modulabs.co.kr/RL_library/3305

Stochastic Policy

Example: Rock-Paper-Scissors

Sometimes you need stochastic policy!!!

Policy Gradient

http://www.modulabs.co.kr/RL_library/3305

Policy Objective Functi

- Goal: given policy $\pi_\theta(s, a)$ with parameters θ, find best θ
- But how do we measure the quality of a policy π_θ?
- In episodic environments we can use the start value

$$\boxed{J_1(\theta) = V^{\pi_\theta}(s_1) = \mathbb{E}_{\pi_\theta}[v_1]}$$

- In continuing environments we can use the average value

$$\boxed{J_{avV}(\theta) = \sum_s d^{\pi_\theta}(s) V^{\pi_\theta}(s)}$$

- Or the average reward per time-step

$$\boxed{J_{avR}(\theta) = \sum_s d^{\pi_\theta}(s) \sum_a \pi_\theta(s, a) \mathcal{R}_s^a}$$

$$d^\pi(s) = \sum_{t=0}^{\infty} \gamma^t Pr\{s_t = s | s_0, \pi\}$$

- where $d^{\pi_\theta}(s)$ is stationary distribution of Markov chain for π_θ

Policy Gradient

http://www.modulabs.co.kr/RL_library/3305

Objective Function의 Gradient를 구하는 방법

1. Finite Difference Policy Gradient
2. Monte-Carlo Policy Gradient
3. Actor-Critic Policy Gradient

Policy Gradient

http://www.modulabs.co.kr/RL_library/3305

Finite Difference Policy Gradient

- To evaluate policy gradient of $\pi_\theta(s, a)$
- For each dimension $k \in [1, n]$
  - Estimate kth partial derivative of objective function w.r.t. θ
  - By perturbing θ by small amount ϵ in kth dimension

$$\frac{\partial J(\theta)}{\partial \theta_k} \approx \frac{J(\theta + \epsilon u_k) - J(\theta)}{\epsilon}$$

where u_k is unit vector with 1 in kth component, 0 elsewhere

- Uses n evaluations to compute policy gradient in n dimensions
- Simple, noisy, inefficient - but sometimes effective
- Works for arbitrary policies, even if policy is not differentiable

Policy Gradient

http://www.modulabs.co.kr/RL_library/3305

Monte-Carlo Policy Gradient

- We now compute the policy gradient *analytically*
- Assume policy π_θ is differentiable whenever it is non-zero
- and we know the gradient $\nabla_\theta \pi_\theta(s, a)$
- Likelihood ratios exploit the following identity

$$\begin{aligned}\nabla_\theta \pi_\theta(s, a) &= \pi_\theta(s, a)\frac{\nabla_\theta \pi_\theta(s, a)}{\pi_\theta(s, a)} \\ &= \pi_\theta(s, a)\nabla_\theta \log \pi_\theta(s, a)\end{aligned}$$

- The $\boxed{\text{score function is } \nabla_\theta \log \pi_\theta(s, a)}$

fast campus

Policy Gradient

http://www.modulabs.co.kr/RL_library/3305

Monte-Carlo Policy Gradient

$$
\begin{aligned}
J(\theta) &= \mathbb{E}_{\pi_\theta}[r] \\
&= \sum_{s \in \mathcal{S}} d(s) \sum_{a \in \mathcal{A}} \pi_\theta(s, a) \mathcal{R}_{s,a} \\
\nabla_\theta J(\theta) &= \sum_{s \in \mathcal{S}} d(s) \sum_{a \in \mathcal{A}} \pi_\theta(s, a) \nabla_\theta \log \pi_\theta(s, a) \mathcal{R}_{s,a} \\
&= \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s, a) r]
\end{aligned}
$$

fast campus

Policy Gradient

http://www.modulabs.co.kr/RL_library/3305

Monte-Carlo Policy Gradient

Policy Gradient Theorem

- The policy gradient theorem generalises the likelihood ratio approach to multi-step MDPs
- Replaces instantaneous reward r with long-term value $Q^{\pi}(s, a)$
- Policy gradient theorem applies to start state objective, average reward and average value objective

Theorem

For any differentiable policy $\pi_\theta(s, a)$,
for any of the policy objective functions $J = J_1, J_{avR},$ or $\frac{1}{1-\gamma} J_{avV}$,
the policy gradient is

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s, a)\ Q^{\pi_\theta}(s, a)\right]$$

Policy Gradient

http://www.modulabs.co.kr/RL_library/3305

Monte-Carlo Policy Gradient

Stochastic Policy Model

*How can we express stochastic policy?

1. Softmax Policy

$$\pi(s,a) = \frac{e^{\theta^T \phi_{sa}}}{\sum_b e^{\theta^T \phi_{sb}}}, \qquad \forall s \in \mathcal{S}, s \in \mathcal{A},$$

2. Gaussian Policy

$$\pi_\theta(s,a) = \frac{1}{\sigma\sqrt{2\pi}} e(-\frac{(a-\mu(s))^2}{2\sigma^2})$$

Monte-Carlo Policy Gradient (REINFORCE)

- Update parameters by stochastic gradient ascent
- Using policy gradient theorem
- Using return v_t as an unbiased sample of $Q^{\pi_\theta}(s_t, a_t)$

$$\Delta\theta_t = \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) v_t$$

function REINFORCE
    Initialise θ arbitrarily
    for each episode $\{s_1, a_1, r_2, ..., s_{T-1}, a_{T-1}, r_T\} \sim \pi_\theta$ **do**
        for $t = 1$ to $T - 1$ **do**
            $\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) v_t$
        end for
    end for
    return θ
end function

Policy Gradient

http://www.modulabs.co.kr/RL_library/3305

Actor-Critic Policy Gradient

- Monte-Carlo policy gradient still has high variance
- We use a critic to estimate the action-value function,

$$Q_w(s,a) \approx Q^{\pi_\theta}(s,a)$$

- Actor-critic algorithms maintain *two* sets of parameters

  Critic Updates action-value function parameters w

  Actor Updates policy parameters θ, in direction suggested by critic

- Actor-critic algorithms follow an *approximate* policy gradient

$$\nabla_\theta J(\theta) \approx \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\ Q_w(s,a)\right]$$
$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(s,a)\ Q_w(s,a)$$

fast campus

Policy Gradient

http://www.modulabs.co.kr/RL_library/3305

Actor-Critic Policy Gradient

Reducing Variance Using a Baseline

- We subtract a baseline function $B(s)$ from the policy gradient
- This can reduce variance, without changing expectation

$$\begin{aligned}\mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a) B(s)\right] &= \sum_{s \in \mathcal{S}} d^{\pi_\theta}(s) \sum_a \nabla_\theta \pi_\theta(s,a) B(s) \\ &= \sum_{s \in \mathcal{S}} d^{\pi_\theta} B(s) \nabla_\theta \sum_{a \in \mathcal{A}} \pi_\theta(s,a) \\ &= 0\end{aligned}$$

- A good baseline is the state value function $B(s) = V^{\pi_\theta}(s)$
- So we can rewrite the policy gradient using the advantage function $A^{\pi_\theta}(s,a)$

$$A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)$$
$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a) \ A^{\pi_\theta}(s,a)\right]$$

Policy Gradient

http://jmlr.org/proceedings/papers/v32/silver14.pdf

Actor-Critic Policy Gradient

2.4. Off-Policy Actor-Critic

It is often useful to estimate the policy gradient *off-policy* from trajectories sampled from a distinct behaviour policy $\beta(a|s) \neq \pi_\theta(a|s)$. In an off-policy setting, the performance objective is typically modified to be the value function of the target policy, averaged over the state distribution of the behaviour policy (Degris et al., 2012b),

$$\begin{aligned} J_\beta(\pi_\theta) &= \int_S \rho^\beta(s) V^\pi(s) \mathrm{d}s \\ &= \int_S \int_A \rho^\beta(s) \pi_\theta(a|s) Q^\pi(s,a) \mathrm{d}a \mathrm{d}s \end{aligned}$$

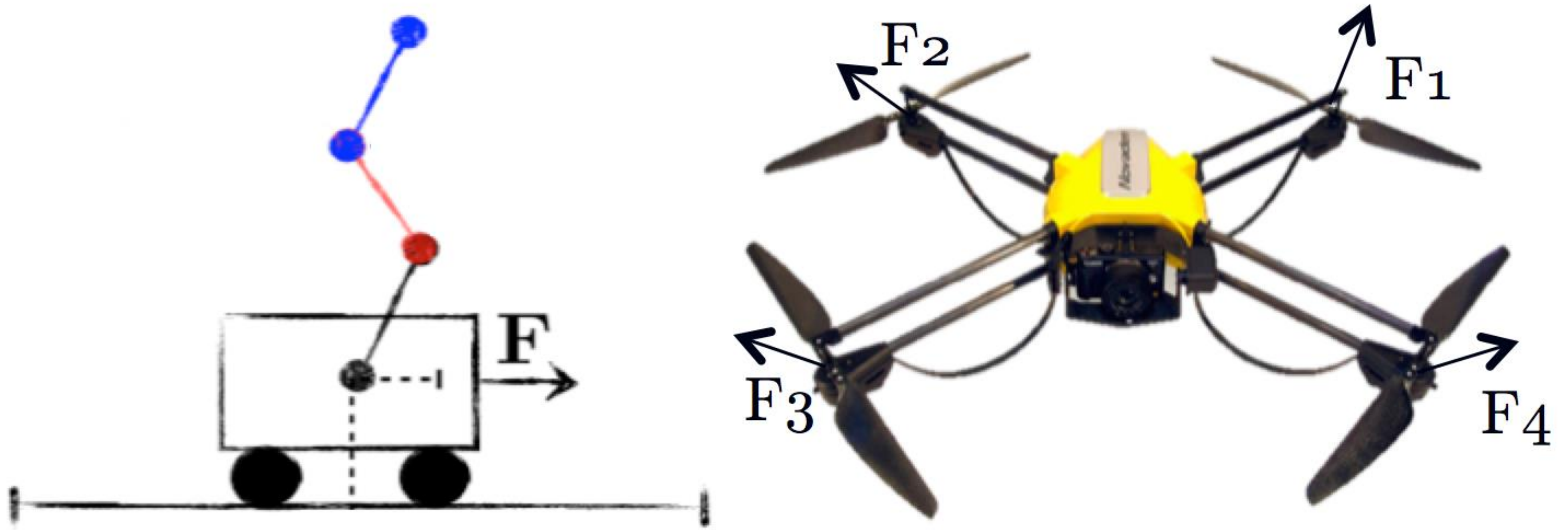
Differentiating the performance objective and applying an approximation gives the *off-policy policy-gradient* (Degris et al., 2012b)

$$\nabla_\theta J_\beta(\pi_\theta) \approx \int_S \int_A \rho^\beta(s) \nabla_\theta \pi_\theta(a|s) Q^\pi(s,a) \mathrm{d}a \mathrm{d}s \quad (4)$$

$$= \mathbb{E}_{s\sim\rho^\beta, a\sim\beta}\left[\frac{\pi_\theta(a|s)}{\beta_\theta(a|s)} \nabla_\theta \log \pi_\theta(a|s) Q^\pi(s,a)\right] \quad (5)$$

Policy Gradient

http://www.modulabs.co.kr/RL_library/3305



Multi-Task Policy Gradient Methods for Control

https://talkingaboutme.tistory.com/entry/RL-Policy-Gradient-Algorithms

fast campus

- *Thank you*