CS342 - Section 11

- Homework 10
  - Due today @11:59pm
  - How was your performance?
- Homework 11
  - On gradient free optimization
- Homework 9
  - Grade will be posted soon

Agenda

- RL from last week
- Evolution Strategy (Gradient free RL)
- QA for Homework 10

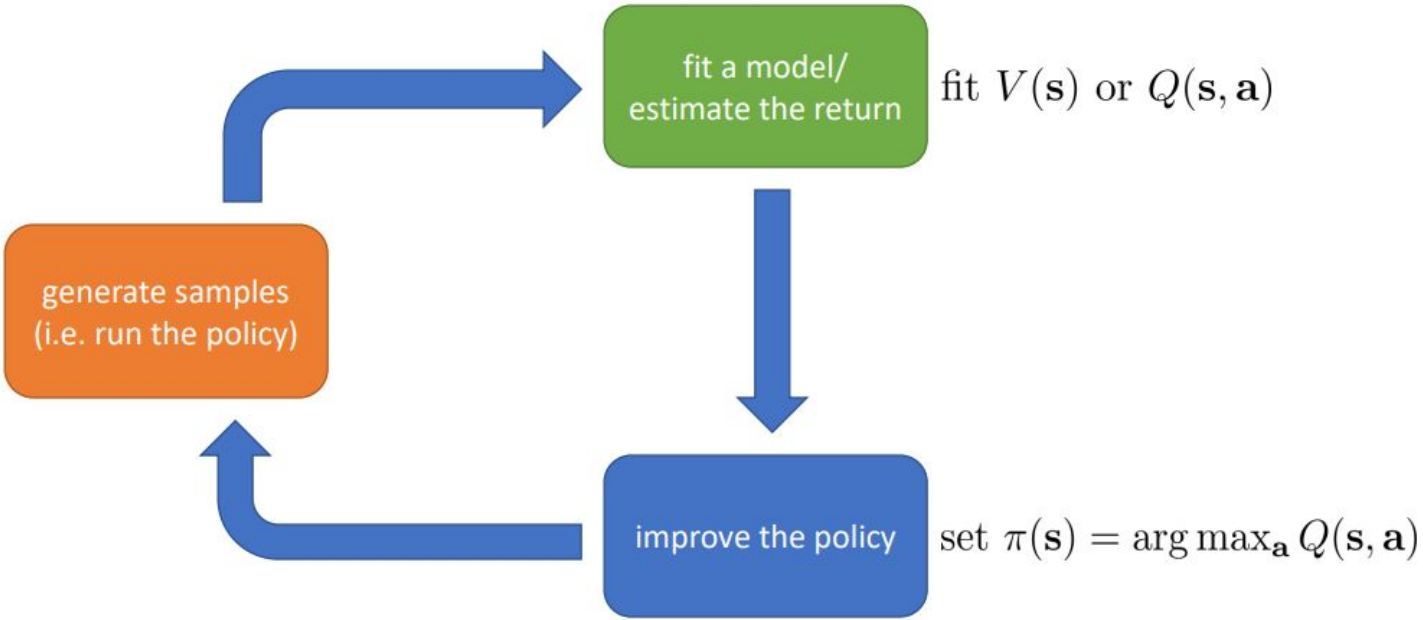
News: RL for everyone

- OpenAI released a detailed RL tutorial yesterday
  - https://blog.openai.com/spinning-up-in-deep-rl/
  - With detailed explanation, clean tutorial code
  - Great learning resource

Recap of RL from lecture

- How to find the policy?
  - Model-based : Explicitly model the environment
  - Model-free : Do not explicitly model the environment
    - On-policy
      - Policy Gradient, Actor-Critic
    - Off-policy
      - Value function approximation

Value functions/Q Learning



Slides from CS294-112

Value functions/Q-Learning

- We improve the policy by maximizing Q(s,a)

$$\text{Set } \pi(s) \text{ s.t. } \operatorname{argmax}_a Q(s,a)$$

$$Q(s,a) = r(s,a) + \operatorname{argmax}_{a'} Q(s',a')$$

Value functions/Q-Learning

- We improve the policy by maximizing Q(s,a)

$$\text{Set } \pi(s) \text{ s.t. } \text{argmax}_a\ Q(s,a)$$

$$Q(s,a) = r(s,a) + \text{argmax}_{a'} Q(s',a')$$

- How do we train Q?
  - Bellman Update:

Given (s,a,s',r)

$$y = r(s,a) + \max_{a'} Q_\Phi(s',a')$$

$$\Phi \leftarrow \text{argmin}_\Phi ||y - Q_\Phi(s,a)||_2$$

Value functions/Q-Learning

- We improve the policy by maximizing Q(s,a)

$$\text{Set } \pi(s) \text{ s.t. } \operatorname{argmax}_a Q(s,a)$$

$$Q(s,a) = r(s,a) + \operatorname{argmax}_{a'} Q(s',a')$$

- How do we train policy?
    - $\pi(s) = \operatorname{argmax}_a Q(s,a)$
        - Discrete case: directly argmax a
        - Continuous case : argmax of a is intractable

Value functions/Q-Learning

- We improve the policy by maximizing Q(s,a)

$$\text{Set } \pi(s) \text{ s.t. } \operatorname{argmax}_a Q(s,a)$$

$$Q(s,a) = r(s,a) + \operatorname{argmax}_{a'} Q(s',a')$$

- How do we train policy?
  - $\pi_\theta(s) = \operatorname{argmax}_a Q_\theta(s,a)$
    - Continuous case : $\theta \leftarrow \theta + \nabla_\theta Q(s,a)|_{s=s',\ a=a'}$

Recap of RL from lecture

- How to find the policy?
  - Model-free : On-policy
    - Policy Gradient
    - Value function/Q-Learning
    - Actor-Critic
- Problem
  - Model-free methods, especially the on-policy ones, requires too many samples!
  - Model-free methods have worse generalization potentials than model-based ones

Gradient Free RL

- Reward Functions might not be smooth
- Credit Assignment in delayed rewards
- Less expensive computations

Evolution Strategy: Key Idea

1. Randomly Sample a set of θ
2. Compute reward for each θ
3. Update θ
4. Repeat

Evolution Strategy: Problem Setup

- Policy: $\pi_\theta(s_t|a_t)$
- Reward / Objective value for complete episode: $F(\theta)$
- θ sampled from $p_\psi = N(\Psi, \sigma^2 I)$
- No requirement for $F(\theta)$ to be smooth

[1] Salimans, Tim, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. "Evolution strategies as a scalable alternative to reinforcement learning." *arXiv preprint arXiv:1703.03864*(2017).

Evolution Strategy

- Average Objective Value: $E_{\theta \sim p\Psi} F(\theta)$
- Gradient Step

$$\nabla_\Psi E_{\theta \sim p\Psi} F(\theta) = E_{\theta \sim p\Psi} F(\theta) \nabla_\Psi \log p_\Psi$$

$$= (1/\sigma^2) E_{\theta \sim p\Psi} F(\theta) (\Psi - \theta)$$

$$= (1/\sigma) E_{\epsilon \sim N(0, I)} F(\theta + \sigma\epsilon) (\epsilon)$$

[1] Salimans, Tim, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. "Evolution strategies as a scalable alternative to reinforcement learning." *arXiv preprint arXiv:1703.03864*(2017).

Evolution Strategy: Algorithm

Algorithm 1 Evolution Strategies

1: **Input:** Learning rate α, noise standard deviation σ, initial policy parameters θ_0
2: **for** $t = 0, 1, 2, \ldots$ **do**
3: Sample $\epsilon_1, \ldots, \epsilon_n \sim \mathcal{N}(0, I)$
4: Compute returns $F_i = F(\theta_t + \sigma\epsilon_i)$ for $i = 1, \ldots, n$
5: Set $\theta_{t+1} \leftarrow \theta_t + \alpha \frac{1}{n\sigma} \sum_{i=1}^{n} F_i \epsilon_i$
6: **end for**

Can Compute in Parallel

[1] Salimans, Tim, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. "Evolution strategies as a scalable alternative to reinforcement learning." *arXiv preprint arXiv:1703.03864*(2017).

Advantages

- Requires Less time to train
  - Benefit of no backpropagation step
- Better exploration behavior
- Operates on complete episodes. Thus, invariant to,
  - Action Frequency
  - Delayed Rewards
  - Long Horizons

Disadvantages

- Requires 3-10 times more data
- Needs F such that at least some better values exist after perturbation

Resources

1. Open AI blog: https://blog.openai.com/evolution-strategies/
2. http://blog.otoro.net/2017/10/29/visual-evolution-strategies/
3. Understanding Derivations: https://davidbarber.github.io/blog/2017/04/03/variational-optimisation/

Questions?