Policy Gradient

CISC 7404 - Decision Making

Steven Morad

University of Macau

Admin

Admin

Homework 1 was due yesterday 23:59

Admin

Homework 1 was due yesterday 23:59

How was the homework?

Admin

Homework 1 was due yesterday 23:59

How was the homework?

We are about 50% finished with the course

Admin

Homework 1 was due yesterday 23:59

How was the homework?

We are about 50% finished with the course

Any opinions/feedback on the course?

Admin

Homework 1 was due yesterday 23:59

How was the homework?

We are about 50% finished with the course

Any opinions/feedback on the course?

- Can also talk after class

Admin

Homework 1 was due yesterday 23:59

How was the homework?

We are about 50% finished with the course

Any opinions/feedback on the course?
- Can also talk after class
- Or email smorad at um.edu.mo

Admin

If you want full participation marks, you must participate in lecture

Admin

If you want full participation marks, you must participate in lecture

Right now, the following students have full participation marks:

Admin

If you want full participation marks, you must participate in lecture

Right now, the following students have full participation marks:

- LIU KEJIA
- LIU HUANRONG
- HOI HOU HONG
- CHEN ZELAI
- WANG ZEKANG
- HE ZHE
- WANG MENGQI
- ZHANG BORONG
- HE ENHAO
- QIAO YULIN
- YAO CHENYU
- KAM KA HOU

Admin

Some names might be missing!

Admin

Some names might be missing!

I am bad with names, but I remember faces

Admin

Some names might be missing!

I am bad with names, but I remember faces

I have a list of university pictures linked to names

Admin

Some names might be missing!

I am bad with names, but I remember faces

I have a list of university pictures linked to names

Some of you look very different in your university pictures

Admin

Some names might be missing!

I am bad with names, but I remember faces

I have a list of university pictures linked to names

Some of you look very different in your university pictures

If you often answer questions but are missing from the list, I will remember you

Admin

Some names might be missing!

I am bad with names, but I remember faces

I have a list of university pictures linked to names

Some of you look very different in your university pictures

If you often answer questions but are missing from the list, I will remember you

Come talk to me during the break or after class

Admin

Some names might be missing!

I am bad with names, but I remember faces

I have a list of university pictures linked to names

Some of you look very different in your university pictures

If you often answer questions but are missing from the list, I will remember you

Come talk to me during the break or after class

For everyone else, there is still time to participate

Ask/answer lecture questions

Admin

Some names might be missing!

I am bad with names, but I remember faces

I have a list of university pictures linked to names

Some of you look very different in your university pictures

If you often answer questions but are missing from the list, I will remember you

Come talk to me during the break or after class

For everyone else, there is still time to participate

Ask/answer lecture questions

Ask questions at office hours

Admin

A cute video of trajectory optimization

Admin

A cute video of trajectory optimization

https://www.youtube.com/watch?v=tudxHzZ5_ls

Admin

Richard Sutton and Andrew Barto (authors of RL textbook) recently won the Turing award ("Nobel prize of computing")

Admin

Richard Sutton and Andrew Barto (authors of RL textbook) recently won the Turing award ("Nobel prize of computing")

Admin

They won the award "For developing the conceptual and algorithmic foundations of reinforcement learning"

Admin

They won the award “For developing the conceptual and algorithmic foundations of reinforcement learning”

Richard Sutton’s *Bitter Lesson*

Admin

They won the award "For developing the conceptual and algorithmic foundations of reinforcement learning"

Richard Sutton's *Bitter Lesson*

http://www.incompleteideas.net/IncIdeas/BitterLesson.html

Parameterized Policies

Parameterized Policies

There are two types of algorithms

Parameterized Policies

There are two types of algorithms

- Value based methods (Q learning, trajectory optimization)

Parameterized Policies

There are two types of algorithms

- Value based methods (Q learning, trajectory optimization)
- Policy gradient methods (today's material)

Parameterized Policies

There are two types of algorithms

- Value based methods (Q learning, trajectory optimization)
- Policy gradient methods (today's material)

All other algorithms are some combination of these two methods

Parameterized Policies

There are two types of algorithms

- Value based methods (Q learning, trajectory optimization)
- Policy gradient methods (today's material)

All other algorithms are some combination of these two methods

Once you know policy gradient, you can understand any decision making algorithm

Parameterized Policies

There are two types of algorithms

- Value based methods (Q learning, trajectory optimization)
- Policy gradient methods (today's material)

All other algorithms are some combination of these two methods

Once you know policy gradient, you can understand any decision making algorithm

Note: LLM finetuning almost always uses policy gradient methods

Parameterized Policies

There are two types of algorithms

- Value based methods (Q learning, trajectory optimization)
- Policy gradient methods (today's material)

All other algorithms are some combination of these two methods

Once you know policy gradient, you can understand any decision making algorithm

Note: LLM finetuning almost always uses policy gradient methods

- Direct Preference Optimization (DPO)

Parameterized Policies

There are two types of algorithms

- Value based methods (Q learning, trajectory optimization)
- Policy gradient methods (today's material)

All other algorithms are some combination of these two methods

Once you know policy gradient, you can understand any decision making algorithm

Note: LLM finetuning almost always uses policy gradient methods

- Direct Preference Optimization (DPO)
- Group Relative Policy Optimization (GRPO)

Parameterized Policies

There are two types of algorithms

- Value based methods (Q learning, trajectory optimization)
- Policy gradient methods (today's material)

All other algorithms are some combination of these two methods

Once you know policy gradient, you can understand any decision making algorithm

Note: LLM finetuning almost always uses policy gradient methods

- Direct Preference Optimization (DPO)
- Group Relative Policy Optimization (GRPO)

Policy gradient can change pretrained model parameters

Parameterized Policies

Q learning is perfect, why do we need a new algorithm?

Parameterized Policies

Q learning is perfect, why do we need a new algorithm?

With Q learning, we learn a Q function

Parameterized Policies

Q learning is perfect, why do we need a new algorithm?

With Q learning, we learn a Q function

From the Q function, we derive an optimal policy

Parameterized Policies

Q learning is perfect, why do we need a new algorithm?

With Q learning, we learn a Q function

From the Q function, we derive an optimal policy

$$\pi(a_t \mid s_t; \theta_\pi) = \begin{cases} 1 \text{ if } a_t = \arg\max_{a \in A} Q(s_t, a, \theta_\pi) \\ 0 \text{ otherwise} \end{cases}$$

Parameterized Policies

Q learning is perfect, why do we need a new algorithm?

With Q learning, we learn a Q function

From the Q function, we derive an optimal policy

$$\pi(a_t \mid s_t; \theta_\pi) = \begin{cases} 1 \text{ if } a_t = \arg\max_{a \in A} Q(s_t, a, \theta_\pi) \\ 0 \text{ otherwise} \end{cases}$$

So why do we need a new algorithm?

Parameterized Policies

Example: Consider a Unitree BenBen, with 12 joints

Parameterized Policies

Example: Consider a Unitree BenBen, with 12 joints

Parameterized Policies

Example: Consider a Unitree BenBen, with 12 joints

Parameterized Policies

Example: Consider a Unitree BenBen, with 12 joints

To learn to motion, we must learn actions for all joints $A \in [0, 2\pi]^{12}$

Parameterized Policies

Parameterized Policies

$$A \in [0, 2\pi]^{12}$$

Parameterized Policies

$$A \in [0, 2\pi]^{12}$$

Question: Can we use our greedy max Q policy for BenBen?

Parameterized Policies

$$A \in [0, 2\pi]^{12}$$

Question: Can we use our greedy max Q policy for BenBen?

$$\pi(a_t \mid s_t; \theta_\pi) = \begin{cases} 1 \text{ if } a_t = \arg\max_{a \in A} Q(s_t, a, \theta_\pi) \\ 0 \text{ otherwise} \end{cases}$$

Parameterized Policies

$$A \in [0, 2\pi]^{12}$$

Question: Can we use our greedy max Q policy for BenBen?

$$\pi(a_t \mid s_t; \theta_\pi) = \begin{cases} 1 \text{ if } a_t = \arg\max_{a \in A} Q(s_t, a, \theta_\pi) \\ 0 \text{ otherwise} \end{cases}$$

Answer: No, $\arg\max_{a \in A}$, but A is infinite. How can we take arg max over an infinite set?

Parameterized Policies

Can discretize action space (similar to last time, discretized state space)

Parameterized Policies

Can discretize action space (similar to last time, discretized state space)

Discrete actions lead to clunky/clumsy robots

Parameterized Policies

Can discretize action space (similar to last time, discretized state space)

Discrete actions lead to clunky/clumsy robots

We want our BenBen to dance naturally

Parameterized Policies

Can discretize action space (similar to last time, discretized state space)

Discrete actions lead to clunky/clumsy robots

We want our BenBen to dance naturally

More flexible algorithm, policy for discrete or **continuous** action spaces

Parameterized Policies

Can discretize action space (similar to last time, discretized state space)

Discrete actions lead to clunky/clumsy robots

We want our BenBen to dance naturally

More flexible algorithm, policy for discrete or **continuous** action spaces

In continuous action spaces, there are infinitely many actions

Parameterized Policies

Can discretize action space (similar to last time, discretized state space)

Discrete actions lead to clunky/clumsy robots

We want our BenBen to dance naturally

More flexible algorithm, policy for discrete or **continuous** action spaces

In continuous action spaces, there are infinitely many actions

Policy gradient can learn over this infinite action space

Parameterized Policies

Can discretize action space (similar to last time, discretized state space)

Discrete actions lead to clunky/clumsy robots

We want our BenBen to dance naturally

More flexible algorithm, policy for discrete or **continuous** action spaces

In continuous action spaces, there are infinitely many actions

Policy gradient can learn over this infinite action space

Does that sound impossible?

Parameterized Policies

The secret is directly learning the policy (action distribution)

Parameterized Policies

The secret is directly learning the policy (action distribution)

$$\pi(a \mid s; \theta_\pi)$$

Parameterized Policies

The secret is directly learning the policy (action distribution)

$$\pi(a \mid s; \theta_\pi)$$

This is a distribution over the action space

Parameterized Policies

The secret is directly learning the policy (action distribution)

$$\pi(a \mid s; \theta_\pi)$$

This is a distribution over the action space

Some distributions (like Gaussian) have infinite support

Parameterized Policies

The secret is directly learning the policy (action distribution)

$$\pi(a \mid s; \theta_\pi)$$

This is a distribution over the action space

Some distributions (like Gaussian) have infinite support

If $\pi(a \mid s; \theta_\pi)$ is Gaussian, every action has nonzero probability

Parameterized Policies

The secret is directly learning the policy (action distribution)

$$\pi(a \mid s; \theta_\pi)$$

This is a distribution over the action space

Some distributions (like Gaussian) have infinite support

If $\pi(a \mid s; \theta_\pi)$ is Gaussian, every action has nonzero probability

We can improve the action distribution over time

Policy Gradient

Policy Gradient

Definition: General form of policy-conditioned discounted return

Policy Gradient

Definition: General form of policy-conditioned discounted return

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

Policy Gradient

Definition: General form of policy-conditioned discounted return

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

Where the state distribution is

Policy Gradient

Definition: General form of policy-conditioned discounted return

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

Where the state distribution is

$$\Pr(s_{n+1} \mid s_0; \theta_\pi) = \sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right)$$

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

Question: What can we change here to change the return?

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

Question: What can we change here to change the return?

Answer: The policy parameters θ_π

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

Question: What can we change here to change the return?

Answer: The policy parameters θ_π

$$\Pr(s_{n+1} \mid s_n) = \sum_{a_n \in A} \mathrm{Tr}(s_{n+1} \mid s_n, a_n) \cdot \pi(a_n \mid s_n; \theta_\pi)$$

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

Question: What can we change here to change the return?

Answer: The policy parameters θ_π

$$\Pr(s_{n+1} \mid s_n) = \sum_{a_n \in A} \mathrm{Tr}(s_{n+1} \mid s_n, a_n) \cdot \pi(a_n \mid s_n; \theta_\pi)$$

Question: How should we change θ_π?

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

Question: What can we change here to change the return?

Answer: The policy parameters θ_π

$$\Pr(s_{n+1} \mid s_n) = \sum_{a_n \in A} \mathrm{Tr}(s_{n+1} \mid s_n, a_n) \cdot \pi(a_n \mid s_n; \theta_\pi)$$

Question: How should we change θ_π?

Answer: Change θ_π so we reach good $s \in S$, making the return larger

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

We want to make $\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi]$ larger

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

We want to make $\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi]$ larger

Question: How to change the policy parameters to improve the return?

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

We want to make $\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi]$ larger

Question: How to change the policy parameters to improve the return?

HINT: Calculus and optimization

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

We want to make $\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi]$ larger

Question: How to change the policy parameters to improve the return?

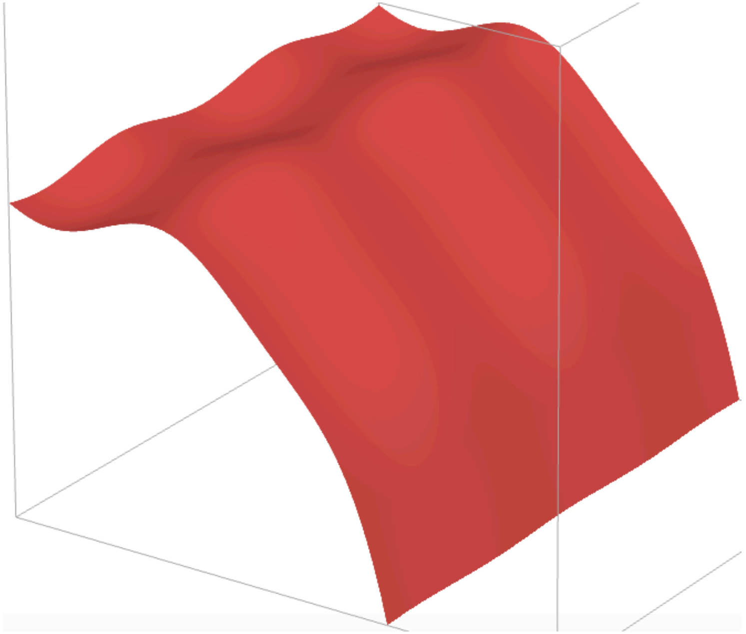
HINT: Calculus and optimization

Policy Gradient

Answer: Gradient ascent, find the greatest slope and move that way

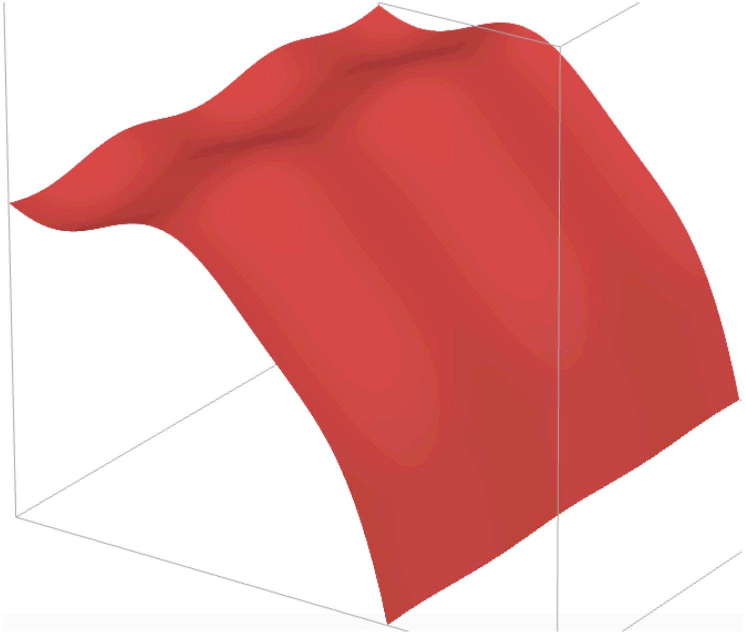
Policy Gradient

Answer: Gradient ascent, find the greatest slope and move that way



Policy Gradient

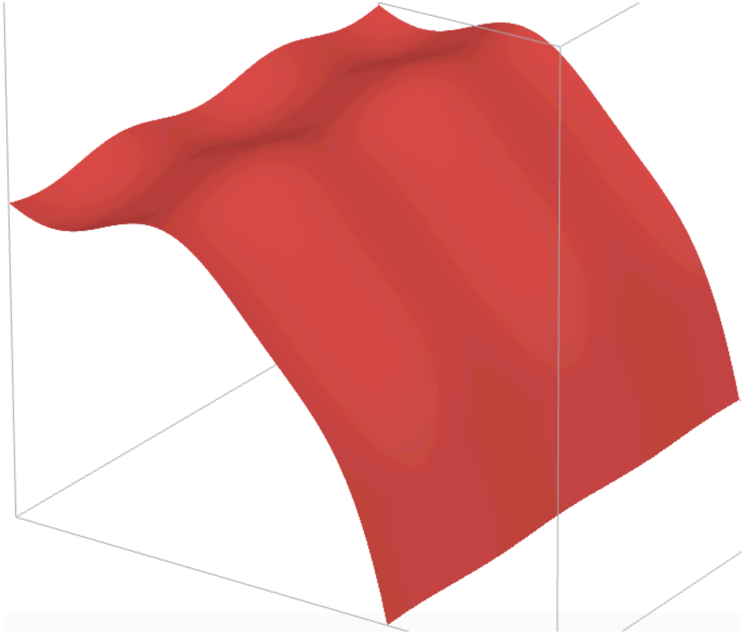
Answer: Gradient ascent, find the greatest slope and move that way



$$\theta_{\pi, i+1} = \theta_{\pi, i} + \alpha \cdot \nabla_{\theta_{\pi, i}} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi, i}]$$

Policy Gradient

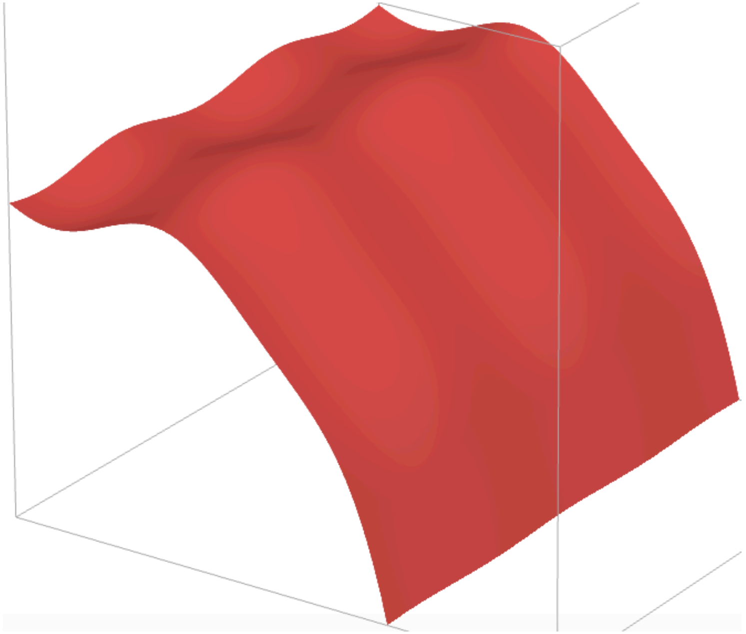
Answer: Gradient ascent, find the greatest slope and move that way



$$\theta_{\pi, i+1} = \underbrace{\theta_{\pi, i}}_{\text{Current policy}} + \alpha \cdot \nabla_{\theta_{\pi, i}} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi, i}]$$

Policy Gradient

Answer: Gradient ascent, find the greatest slope and move that way



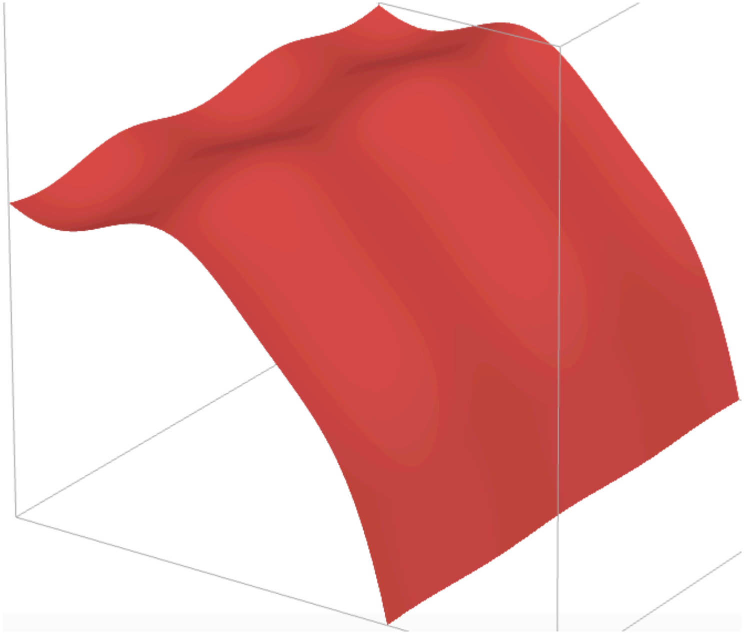
Current policy

$$\theta_{\pi,i+1} = \boxed{\theta_{\pi,i}} + \alpha \cdot \boxed{\nabla_{\theta_{\pi,i}} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi,i}\right]}$$

θ direction that maximizes return

Policy Gradient

Answer: Gradient ascent, find the greatest slope and move that way



$$\underbrace{\theta_{\pi,i+1}}_{\text{New policy}} = \overbrace{\theta_{\pi,i}}^{\text{Current policy}} + \alpha \cdot \underbrace{\nabla_{\theta_{\pi,i}} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi,i}]}_{\theta \text{ direction that maximizes return}}$$

Policy Gradient

Current policy

$$\theta_{\pi, i+1} = \theta_{\pi, i} + \alpha \cdot \nabla_{\theta_{\pi, i}} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi, i}]$$

New policy

θ direction that maximizes return

Policy Gradient

Current policy

$$\boxed{\theta_{\pi,i+1}} = \boxed{\theta_{\pi,i}} + \alpha \cdot \boxed{\nabla_{\theta_{\pi,i}} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi,i}]}$$

New policy

θ direction that maximizes return

If find $\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi]$, we can improve the policy and return

Policy Gradient

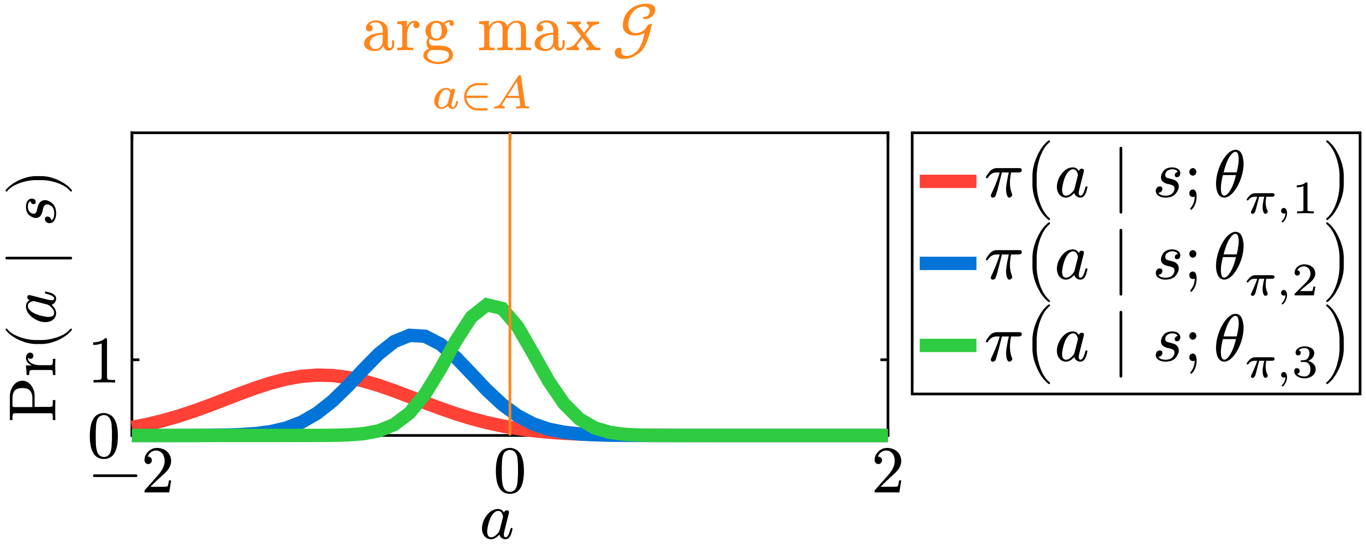
Current policy

$$\theta_{\pi,i+1} = \theta_{\pi,i} + \alpha \cdot \nabla_{\theta_{\pi,i}} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi,i}\right]$$

New policy

θ direction that maximizes return

If find $\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi]$, we can improve the policy and return



Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

$$\Pr(s_{n+1} \mid s_0; \theta_\pi) = \sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right)$$

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

$$\Pr(s_{n+1} \mid s_0; \theta_\pi) = \sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right)$$

We want

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi]$$

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

$$\Pr(s_{n+1} \mid s_0; \theta_\pi) = \sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right)$$

We want

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi]$$

First, combine top two equations so we have more space

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

$$\Pr(s_{n+1} \mid s_0; \theta_\pi) = \sum_{s_1, \dots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right)$$

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

$$\Pr(s_{n+1} \mid s_0; \theta_\pi) = \sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right)$$

Plug line 2 into line 1

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \Pr(s_{n+1} \mid s_0; \theta_\pi)$$

$$\Pr(s_{n+1} \mid s_0; \theta_\pi) = \sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right)$$

Plug line 2 into line 1

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot$$

$$\left(\sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \right)$$

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot$$

$$\left(\sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \right)$$

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot$$

$$\left(\sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \right)$$

Take the gradient with respect to θ_π of both sides

Policy Gradient

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot$$

$$\left(\sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \right)$$

Take the gradient with respect to θ_π of both sides

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \nabla_{\theta_\pi} \left[\sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \right.$$

$$\left. \left(\sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \right) \right]$$

Policy Gradient

$$= \nabla_{\theta_\pi} \left[\sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \right.$$

$$\left. \left(\sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \right) \right]$$

Policy Gradient

$$= \nabla_{\theta_\pi} \left[\sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \right.$$

$$\left. \left(\sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \right) \right]$$

Move gradient inside sums

Policy Gradient

$$= \nabla_{\theta_\pi} \left[\sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \right.$$

$$\left. \left(\sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \right) \right]$$

Move gradient inside sums

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot$$

$$\nabla_{\theta_\pi} \left[\sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot$$

$$\nabla_{\theta_\pi} \left[\sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot$$

$$\nabla_{\theta_\pi} \left[\sum_{s_1, \ldots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \right]$$

Rewrite $\Pr(s_{n+1} \mid s_0; \theta_\pi)$ by pulling action sum outside

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot$$

$$\nabla_{\theta_\pi} \left[\sum_{s_1, \dots, s_n \in S} \prod_{t=0}^{n} \left(\sum_{a_t \in A} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \right]$$

Rewrite $\Pr(s_{n+1} \mid s_0; \theta_\pi)$ by pulling action sum outside

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot$$

$$\nabla_{\theta_\pi} \left[\sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A} \prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot$$

$$\nabla_{\theta_\pi} \left[\sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A} \prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot$$

$$\nabla_{\theta_\pi} \left[\sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A} \prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right]$$

Move the gradient operator further inside the sum

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot$$

$$\nabla_{\theta_\pi} \left[\sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A} \prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right]$$

Move the gradient operator further inside the sum

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\nabla_{\theta_\pi} \left[\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\nabla_{\theta_\pi} \left[\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\nabla_{\theta_\pi} \left[\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right]$$

Want to move ∇ inside, split product

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\nabla_{\theta_\pi} \left[\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right]$$

Want to move ∇ inside, split product

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\nabla_{\theta_\pi} \left[\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \left(\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\nabla_{\theta_\pi} \left[\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \left(\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\nabla_{\theta_\pi} \left[\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \left(\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right) \right]$$

First term is a constant factor, pull out constant

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\nabla_{\theta_\pi} \left[\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \left(\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right) \right]$$

First term is a constant factor, pull out constant

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \nabla_{\theta_\pi} \left[\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \nabla_{\theta_\pi} \left[\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \nabla_{\theta_\pi} \left[\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right]$$

Can use chain and product rule, but will create a mess of terms

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \nabla_{\theta_\pi} \left[\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right]$$

Can use chain and product rule, but will create a mess of terms

$$\nabla_{\theta_\pi}[\pi(a_n \mid s_n; \theta_\pi) \ldots \cdot \pi(a_0 \mid s_0; \theta_\pi)] =$$

$$\nabla_{\theta_\pi}[\pi(a_n \mid s_n; \theta_\pi)] \cdot \pi(a_{n-1} \mid s_{n-1}; \theta_\pi) \ldots$$

It will be very expensive/intractable to compute all terms!

Policy Gradient

Log-derivative trick:

Policy Gradient

Log-derivative trick:

Question: What is

$$\nabla_x \log(f(x))$$

Policy Gradient

Log-derivative trick:

Question: What is

$$\nabla_x \log(f(x))$$

Answer:

$$\nabla_x \log(f(x)) = \frac{1}{f(x)} \cdot \nabla_x f(x)$$

Policy Gradient

Log-derivative trick:

Question: What is

$$\nabla_x \log(f(x))$$

Answer:

$$\nabla_x \log(f(x)) = \frac{1}{f(x)} \cdot \nabla_x f(x)$$

$$f(x) \nabla_x \log f(x) = \nabla_x f(x)$$

Policy Gradient

Log-derivative trick:

Question: What is

$$\nabla_x \log(f(x))$$

Answer:

$$\begin{aligned}\nabla_x \log(f(x)) &= \frac{1}{f(x)} \cdot \nabla_x f(x) \\ f(x) \nabla_x \log f(x) &= \nabla_x f(x) \\ \nabla_x f(x) &= f(x) \nabla_x \log f(x)\end{aligned}$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \nabla_{\theta_\pi} \left[\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \nabla_{\theta_\pi} \left[\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right]$$

$$\nabla_x f(x) = f(x) \nabla_x \log f(x)$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \nabla_{\theta_\pi} \left[\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right]$$

$$\nabla_x f(x) = f(x) \nabla_x \log f(x)$$

Apply log-derivative trick to $\nabla \prod$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \nabla_{\theta_\pi} \left[\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right]$$

$$\nabla_x f(x) = f(x) \nabla_x \log f(x)$$

Apply log-derivative trick to $\nabla \prod$

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \cdot \nabla_{\theta_\pi} \left[\log \left(\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \nabla_{\theta_\pi} \left[\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right]$$

$$\nabla_x f(x) = f(x) \nabla_x \log f(x)$$

Apply log-derivative trick to $\nabla \prod$

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \cdot \nabla_{\theta_\pi} \left[\log \left(\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \cdot \nabla_{\theta_\pi} \left[\log \left(\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \cdot \nabla_{\theta_\pi} \left[\log \left(\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right) \right]$$

The log of products is the sum of logs: $\log(ab) = \log(a) + \log(b)$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \cdot \nabla_{\theta_\pi} \left[\log \left(\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right) \right]$$

The log of products is the sum of logs: $\log(ab) = \log(a) + \log(b)$

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \cdot \nabla_{\theta_\pi} \left[\sum_{t=0}^{n} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \cdot \nabla_{\theta_\pi} \left[\log \left(\prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \right) \right]$$

The log of products is the sum of logs: $\log(ab) = \log(a) + \log(b)$

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \cdot \nabla_{\theta_\pi} \left[\sum_{t=0}^{n} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \cdot \nabla_{\theta_\pi} \left[\sum_{t=0}^{n} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \cdot \nabla_{\theta_\pi} \left[\sum_{t=0}^{n} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Gradient of sum is sum of gradients

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \cdot \nabla_{\theta_\pi} \left[\sum_{t=0}^{n} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Gradient of sum is sum of gradients

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \cdot \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \cdot \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \cdot \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Last step, recombine product

Policy Gradient

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \right) \cdot \prod_{t=0}^{n} \pi(a_t \mid s_t; \theta_\pi) \cdot \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Last step, recombine product

$$= \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \cdot \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

This is the **policy gradient**

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \cdot \sum_{s_1, \dots, s_n \in S} \sum_{a_0, \dots, a_n \in A}$$

$$\left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

This is the **policy gradient**

Rewrote the gradient of the return in terms of the gradient of policy

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1})$$

$$\cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A} \left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1})$$

$$\cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A} \left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Question: Is this familiar?

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1})$$

$$\cdot \sum_{s_1, \ldots, s_n \in S} \sum_{a_0, \ldots, a_n \in A} \left(\prod_{t=0}^{n} \mathrm{Tr}(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t; \theta_\pi) \right) \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Question: Is this familiar?

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] =$$

$$\sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \Pr(s_{n+1} \mid s_0; \theta_\pi) \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] =$$

$$\sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \Pr(s_{n+1} \mid s_0; \theta_\pi) \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] =$$

$$\sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \Pr(s_{n+1} \mid s_0; \theta_\pi) \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Question: Is this familiar?

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] =$$

$$\sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \Pr(s_{n+1} \mid s_0; \theta_\pi) \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Question: Is this familiar?

$$\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi]$$

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] =$$

$$\sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \Pr(s_{n+1} \mid s_0; \theta_\pi) \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] =$$

$$\sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \Pr(s_{n+1} \mid s_0; \theta_\pi) \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Careful rewriting as return because π relies on n

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] =$$

$$\sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \mathcal{R}(s_{n+1}) \Pr(s_{n+1} \mid s_0; \theta_\pi) \left[\sum_{t=0}^{n} \nabla_{\theta_\pi} \log \pi(a_t \mid s_t; \theta_\pi) \right]$$

Careful rewriting as return because π relies on n

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s, a \in \boldsymbol{\tau}} \nabla_{\theta_\pi} \log \pi(a \mid s; \theta_\pi)$$

We can write the gradient of the return in terms of the policy gradient

Policy Gradient

Definition: The policy gradient family of algorithms

Policy Gradient

Definition: The policy gradient family of algorithms

Update the parameters iteratively until convergence

Policy Gradient

Definition: The policy gradient family of algorithms

Update the parameters iteratively until convergence

$$\theta_{\pi, i+1} = \theta_{\pi, i} + \alpha \cdot \nabla_{\theta_{\pi, i}} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi, i}]$$

Policy Gradient

Definition: The policy gradient family of algorithms

Update the parameters iteratively until convergence

$$\theta_{\pi, i+1} = \theta_{\pi, i} + \alpha \cdot \nabla_{\theta_{\pi, i}} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi, i}\right]$$

Using the policy gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s, a \in \boldsymbol{\tau}} \nabla_{\theta_\pi} \log \pi(a \mid s; \theta_\pi)$$

Policy Gradient

$$\theta_{\pi,i+1} = \theta_{\pi,i} + \alpha \cdot \nabla_{\theta_{\pi,i}} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi,i}\right]$$

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s,a \in \boldsymbol{\tau}} \nabla_{\theta_\pi} \log \pi(a \mid s; \theta_\pi)$$

Policy Gradient

$$\theta_{\pi,i+1} = \theta_{\pi,i} + \alpha \cdot \nabla_{\theta_{\pi,i}} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi,i}]$$

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s,a \in \boldsymbol{\tau}} \nabla_{\theta_\pi} \log \pi(a \mid s; \theta_\pi)$$

Question: How to find this?

Policy Gradient

$$\theta_{\pi,i+1} = \theta_{\pi,i} + \alpha \cdot \nabla_{\theta_{\pi,i}} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi,i}\right]$$

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s,a \in \boldsymbol{\tau}} \nabla_{\theta_\pi} \log \pi(a \mid s; \theta_\pi)$$

Question: How to find this?

Answer: Estimate expectation empirically

Policy Gradient

$$\theta_{\pi,i+1} = \theta_{\pi,i} + \alpha \cdot \nabla_{\theta_{\pi,i}} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi,i}\right]$$

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s,a \in \boldsymbol{\tau}} \nabla_{\theta_\pi} \log \pi(a \mid s; \theta_\pi)$$

Question: How to find this?

Answer: Estimate expectation empirically

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \approx \hat{\mathbb{E}}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s,a \in \boldsymbol{\tau}} \nabla_{\theta_\pi} \log \pi(a \mid s; \theta_\pi)$$

Policy Gradient

Definition: The REINFORCE algorithm updates θ_π with empirical returns

Policy Gradient

Definition: The REINFORCE algorithm updates θ_π with empirical returns

Monte-Carlo variant of policy gradient

Policy Gradient

Definition: The REINFORCE algorithm updates θ_π with empirical returns

Monte-Carlo variant of policy gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \approx \hat{\mathbb{E}}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s, a \in \boldsymbol{\tau}} \nabla_{\theta_\pi} \log \pi(a \mid s; \theta_\pi)$$

Policy Gradient

Definition: The REINFORCE algorithm updates θ_π with empirical returns

Monte-Carlo variant of policy gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \approx \hat{\mathbb{E}}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s, a \in \boldsymbol{\tau}} \nabla_{\theta_\pi} \log \pi(a \mid s; \theta_\pi)$$

$$\theta_{\pi, i+1} = \theta_{\pi, i} + \alpha \cdot \nabla_{\theta_{\pi, i}} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi, i}\right]$$

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \approx \hat{\mathbb{E}}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s, a \in \boldsymbol{\tau}} \nabla_{\theta_\pi} \log \pi(a \mid s; \theta_\pi)$$

$$\theta_{\pi, i+1} = \theta_{\pi, i} + \alpha \cdot \nabla_{\theta_{\pi, i}} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi, i}\right]$$

Question: Is REINFORCE on-policy or off-policy?

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \approx \hat{\mathbb{E}}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s, a \in \boldsymbol{\tau}} \nabla_{\theta_\pi} \log \pi(a \mid s; \theta_\pi)$$

$$\theta_{\pi, i+1} = \theta_{\pi, i} + \alpha \cdot \nabla_{\theta_{\pi, i}} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi, i}\right]$$

Question: Is REINFORCE on-policy or off-policy?

HINT:

- On-policy algorithms require data collected with θ_π
- Off-policy algorithms can use data collected with any policy

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \approx \hat{\mathbb{E}}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s,a \in \boldsymbol{\tau}} \nabla_{\theta_\pi} \log \pi(a \mid s; \theta_\pi)$$

$$\theta_{\pi,i+1} = \theta_{\pi,i} + \alpha \cdot \nabla_{\theta_{\pi,i}} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi,i}\right]$$

Question: Is REINFORCE on-policy or off-policy?

HINT:

- On-policy algorithms require data collected with θ_π
- Off-policy algorithms can use data collected with any policy

Answer: On-policy, empirical return based on θ_π

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s, a \in \boldsymbol{\tau}} \log \pi(a \mid s; \theta_\pi)$$

$$\theta_{\pi, i+1} = \theta_{\pi, i} + \alpha \cdot \nabla_{\theta_{\pi, i}} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi, i}\right]$$

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s,a \in \boldsymbol{\tau}} \log \pi(a \mid s; \theta_\pi)$$

$$\theta_{\pi,i+1} = \theta_{\pi,i} + \alpha \cdot \nabla_{\theta_{\pi,i}} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi,i}\right]$$

Question: Any other ways to express $\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi]$?

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s, a \in \boldsymbol{\tau}} \log \pi(a \mid s; \theta_\pi)$$

$$\theta_{\pi, i+1} = \theta_{\pi, i} + \alpha \cdot \nabla_{\theta_{\pi, i}} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi, i}\right]$$

Question: Any other ways to express $\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi]$?

Answer: Value function or Q function

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s,a \in \boldsymbol{\tau}} \log \pi(a \mid s; \theta_\pi)$$

$$\theta_{\pi,i+1} = \theta_{\pi,i} + \alpha \cdot \nabla_{\theta_{\pi,i}} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi,i}\right]$$

Question: Any other ways to express $\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi]$?

Answer: Value function or Q function

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = V(s_0; \theta_\pi, \theta_V) \cdot \sum_{s,a \in \boldsymbol{\tau}} \log \pi(a \mid s; \theta_\pi)$$

Policy Gradient

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] \cdot \sum_{s,a \in \boldsymbol{\tau}} \log \pi(a \mid s; \theta_\pi)$$

$$\theta_{\pi,i+1} = \theta_{\pi,i} + \alpha \cdot \nabla_{\theta_{\pi,i}} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_{\pi,i}\right]$$

Question: Any other ways to express $\mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi]$?

Answer: Value function or Q function

$$\nabla_{\theta_\pi} \mathbb{E}[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\pi] = V(s_0; \theta_\pi, \theta_V) \cdot \sum_{s,a \in \boldsymbol{\tau}} \log \pi(a \mid s; \theta_\pi)$$

We call this **actor-critic**, more discussion next time

Coding

Coding

We can implement our policy using all sorts of action distributions

[1] "Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution." International conference on machine learning. PMLR, 2017.

Coding

We can implement our policy using all sorts of action distributions

For discrete tasks, we often use **categorical** distributions

[1] "Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution." International conference on machine learning. PMLR, 2017.

Coding

We can implement our policy using all sorts of action distributions

For discrete tasks, we often use **categorical** distributions

For continuous tasks, we usually use **normal** distributions

---

[1] "Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution." International conference on machine learning. PMLR, 2017.

Coding

We can implement our policy using all sorts of action distributions

For discrete tasks, we often use **categorical** distributions

For continuous tasks, we usually use **normal** distributions

However, some people say beta distributions work better![1]

---

[1] "Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution." International conference on machine learning. PMLR, 2017.

Coding

The coding looks a little bit different than the math

Coding

The coding looks a little bit different than the math

We often separate the model, policy distribution, and sampled action

```
# Compute "logits", unnormalized probabilities
z = model(x)
# Create distribution pi(a | s; theta_pi)
p_a = dist(z) # For loss function
# Sample action that we use in environment
a = sample(p_a) # For env step
```

Coding

Create a model for discrete action spaces

Coding

Create a model for discrete action spaces

```
discrete_action_model = nn.Sequential([
    nn.Linear(state_size, hidden_size),
    nn.Lambda(leaky_relu),
    nn.Linear(hidden_size, hidden_size),
    nn.Lambda(leaky_relu)
    # Output logits for possible actions
    nn.Linear(hidden_size, action_size),
])
```

Coding

Create a model for discrete action spaces

```
discrete_action_model = nn.Sequential([
    nn.Linear(state_size, hidden_size),
    nn.Lambda(leaky_relu),
    nn.Linear(hidden_size, hidden_size),
    nn.Lambda(leaky_relu)
    # Output logits for possible actions
    nn.Linear(hidden_size, action_size),
])
```

Coding

Need a function to get actions for our environment

```
def sample_action(model, state, key):
    z = model(state)
    # BE VERY CAREFUL, always read documentation
    # Sometimes takes UNNORMALIZED logits, sometimes probs
    action_probs = softmax(model, state)
    a = categorical(key, action_probs)
    a = categorical(key, z) # Does not even use pi
    return a
```

Coding

```
def REINFORCE_loss(model, episode):
    """REINFORCE for discrete actions"""
    G = compute_return(rewards) # empirical return
    # We need log(pi(a | s)), softmax => probs
    # log_softmax more stable than log(softmax(x))
    log_probs = log_softmax(model(episode.states))
    # We only update the policy for the actions we took
    # Discrete/categorical actions
    # Can use sum or mean
    policy_gradient = mean(G * log_probs[episode.actions])
    # Want gradient ascent, most library do gradient descent
    return -policy_gradient
```

Coding

What about continuous action spaces?

Coding

What about continuous action spaces?

```
continuous_action_model = nn.Sequential([
    nn.Linear(state_size, hidden_size),
    nn.Lambda(leaky_relu),
    nn.Linear(hidden_size, hidden_size),
    nn.Lambda(leaky_relu)
    nn.Linear(hidden_size, 2 * action_size),
    # Like to use a diagonal multivariate Gaussian
    # Assumes independence between actions (approximation)
    nn.Lambda(lambda x: split(x, 2))
])
```

Coding

```
def sample_action(model, state, key):
    # Log(sigma) because neural network outputs +/-
    # sigma only + but log_sigma can be +/-
    mu, log_sigma = model(state)
    a = normal(key, mu, exp(sigma))
    return a
```

Coding

```
def REINFORCE_loss(model, episode):
    """REINFORCE for continuous actions using Gaussian pi"""
    G = compute_return(rewards) # empirical return
    # Policy outputs mean and log(std dev)
    mus, log_sigmas = model(episode.states)
    # Log probability from equation of Gaussian
    log_probs = -(
        (episode.actions - mus) ** 2
        / (2 * exp(log_sigmas) ** 2)
        + log_sigmas
    )
    policy_gradient = mean(G * log_probs)
    # Want gradient ascent, library does gradient descent
    return -policy_gradient
```

Homework

Homework

Homework 2 is the final homework, and it is a little special

Homework

Homework 2 is the final homework, and it is a little special

Choose only one assignment

Homework

Homework 2 is the final homework, and it is a little special

Choose only one assignment

- Policy gradient (easier, max 80/100)

Homework

Homework 2 is the final homework, and it is a little special

Choose only one assignment

- Policy gradient (easier, max 80/100)
- Deep Q learning (harder, max 100/100)

Homework

Homework 2 is the final homework, and it is a little special

Choose only one assignment
- Policy gradient (easier, max 80/100)
- Deep Q learning (harder, max 100/100)

I will only grade one assignment

Homework

Homework 2 is the final homework, and it is a little special

Choose only one assignment
- Policy gradient (easier, max 80/100)
- Deep Q learning (harder, max 100/100)

I will only grade one assignment

You can try and estimate the return for completing each assignment

Homework

Homework 2 is the final homework, and it is a little special

Choose only one assignment

- Policy gradient (easier, max 80/100)
- Deep Q learning (harder, max 100/100)

I will only grade one assignment

You can try and estimate the return for completing each assignment

My suggestion: start with policy gradient

Homework

Homework 2 is the final homework, and it is a little special

Choose only one assignment

- Policy gradient (easier, max 80/100)
- Deep Q learning (harder, max 100/100)

I will only grade one assignment

You can try and estimate the return for completing each assignment

My suggestion: start with policy gradient

If you solve policy gradient early, then try deep Q learning

Homework

Homework 2 is the final homework, and it is a little special

Choose only one assignment
- Policy gradient (easier, max 80/100)
- Deep Q learning (harder, max 100/100)

I will only grade one assignment

You can try and estimate the return for completing each assignment

My suggestion: start with policy gradient

If you solve policy gradient early, then try deep Q learning

Deep Q learning requires more hyperparameter tuning

Homework

Start early, one training run can take up to an hour

Homework

Start early, one training run can take up to an hour

If you have bugs, you will need to restart training

Homework

Start early, one training run can take up to an hour

If you have bugs, you will need to restart training

If you wait until the day before, you will not succeed

Homework

Start early, one training run can take up to an hour

If you have bugs, you will need to restart training

If you wait until the day before, you will not succeed

Due in 3 weeks (04.09)

Homework

https://colab.research.google.com/drive/1JWfMYviwt7tgU08QDeIZV82MuzVQZbX1?usp=sharing

https://colab.research.google.com/drive/1qKXsaOpT27paCmPA-Hbh_-PtbQnrrkla?usp=sharing