بسم الله الرحمن الرحیم

هوش مصنوعی پیشرفته

فصل ۲۱

یادگیری تقویتی

Reinforcement Learning

کاظم فولادی
دانشکده مهندسی برق و کامپیوتر
دانشگاه تهران

http://courses.fouladi.ir/aai

هوش مصنوعی پیشرفته

یادگیری تقویتی

۱

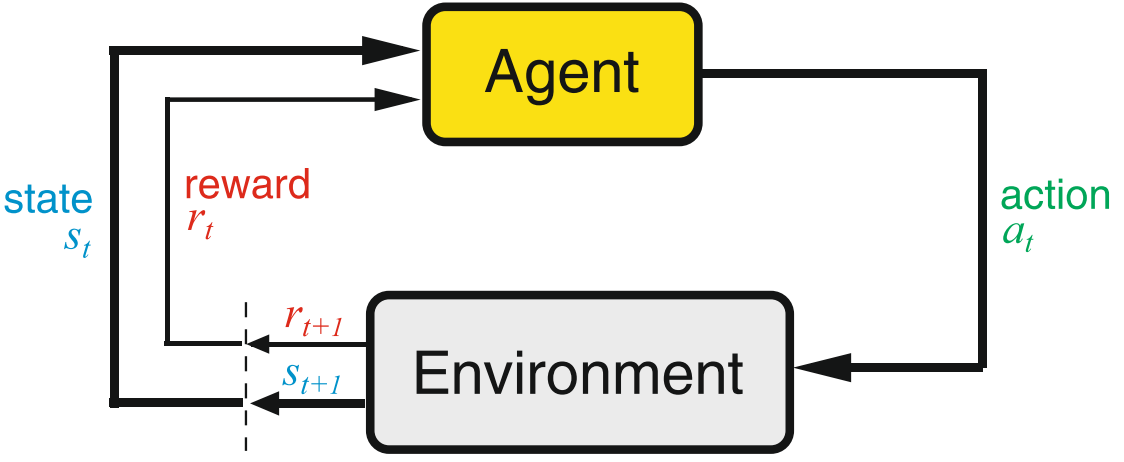
مقدمه

یادگیری تقویتی

عامل و محیط

REINFORCEMENT LEARNING

- فیدبک اصلی دریافت شده توسط عامل: در قالب **پاداش‌ها**
- سودمندی عامل توسط تابع پاداش تعریف می‌شود.
- هدف: یادگیری چگونگی کنش به منظور ماکزیمم کردن امید پاداش‌ها



یادگیری تقویتی

مثال: یادگیری در حیوانات

EXAMPLE: ANIMAL LEARNING

یادگیری تقویتی (RL) بیش از ۷۰ سال به طور تجربی در روان‌شناسی مطالعه شده است.
برای یادگیری در حیوانات

پاداش‌ها: غذا، درد، گرسنگی، دارو، ...

مثال: زنبور عسل
زنبورهای عسل در میدان گل‌های مصنوعی با منابع شهد کنترل شده
Plan نزدیک به بهینه را یاد می‌گیرند.

(زنبورها یک اتصال عصبی مستقیم از اندازه‌گیری میزان شهد به حوزه‌ی طرح‌ریزی موتوری دارند.)

یادگیری تقویتی

REINFORCEMENT LEARNING

عامل در یک محیط MDP یا POMDP قرار دارد.
تنها فیدبک برای یادگیری: ادراک‌ها + پاداش‌ها

عامل باید یک **سیاست** را به یکی از شکل‌های زیر یاد بگیرد:

۱. **مدل گذر** *Transition Model* $T(s, a, s')$ + **تابع ارزش** *Value Function* $U(s)$

۲. **امید سودمندی** *Expected Utility* $Q(a, s)$
مقدار مورد انتظار سودمندی اگر کنش a را در حالت s انجام دهیم و سپس به طور بهینه کنش کنیم.

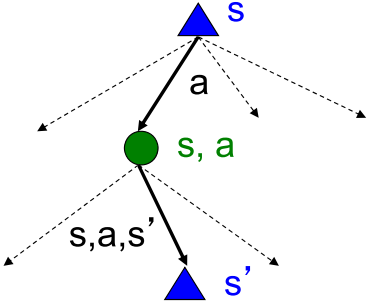
۳. **سیاست** *Policy* $\pi(s)$
کنشی که در حالت s باید انجام شود.

یادگیری تقویتی

مسائل تصمیم‌گیری مارکوف

MDPs

مؤلفه‌های تعریف یک MDP				
کنش‌های عامل *Actions of Agent*	**حالت‌های محیط** *States of Environment*	**حالت آغازین** *Initial State*	**مدل گذر** *Transition Model*	**تابع پاداش** *Reward Function*
States $s \in S$, actions $a \in A$		s_0	$T(s,a,s')$	$R(s)$ $R(s,a)$ $R(s,a,s')$



روش‌های راه‌حل		
تکرار ارزش *Value Iteration (VI)*	**تکرار سیاست** *Policy Iteration (PI)*	...

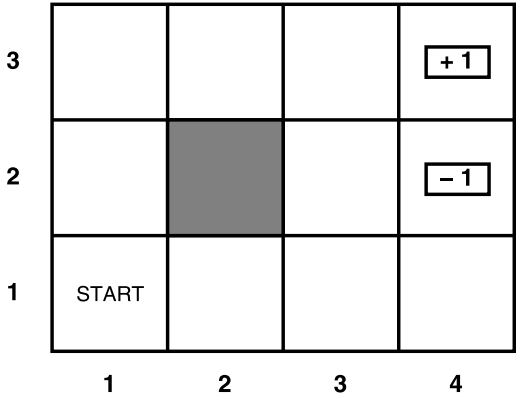
محدودیت‌ها:

* فضای حالت نباید زیاد بزرگ باشد.
* فرض شده است T و R (مدل محیط) معلوم است.

راه‌حل: روش‌های یادگیری تقویتی (Reinforcement Learning)

یادگیری تقویتی

مثال: دنیای ۴ در ۳



کنش‌ها = {Right, Left, Down, Up}

تابع پاداش

Reward function $R(s)$ (or $R(s,a)$, $R(s,a,s')$)

$$= \begin{cases} -0.04 & \text{(جریمه/پنالتی کوچک) برای حالت‌های ناپایانی} \\ \pm 1 & \text{برای حالت‌های پایانی} \end{cases}$$

$$(1,1)_{-.04} \rightarrow (1,2)_{-.04} \rightarrow (1,3)_{-.04} \rightarrow (1,2)_{-.04} \rightarrow (1,3)_{-.04} \rightarrow \cdots (4,3)_{+1}$$
$$(1,1)_{-.04} \rightarrow (1,2)_{-.04} \rightarrow (1,3)_{-.04} \rightarrow (2,3)_{-.04} \rightarrow (3,3)_{-.04} \rightarrow \cdots (4,3)_{+1}$$
$$(1,1)_{-.04} \rightarrow (2,1)_{-.04} \rightarrow (3,1)_{-.04} \rightarrow (3,2)_{-.04} \rightarrow (4,2)_{-1}.$$

یادگیری مبتنی بر مدل/ رها از مدل

مثال

MODEL-BASED VS. MODEL-FREE LEARNING

هدف: محاسبه‌ی امید سن (مقدار مورد انتظار سن) دانشجویان

Known P(A)

$$E[A] = \sum_a P(a) \cdot a = 0.35 \times 20 + \cdots$$

بدون داشتن $P(A)$ باید نمونه گردآوری کنیم: $[a_1, a_2, \ldots, a_N]$

Unknown P(A): "Model Based"

$$\hat{P}(a) = \frac{\text{num}(a)}{N}$$

$$E[A] \approx \sum_a \hat{P}(a) \cdot a$$

Unknown P(A): "Model Free"

$$E[A] \approx \frac{1}{N} \sum_i a_i$$

یادگیری تقویتی مبتنی بر مدل/ رها از مدل

مثال

MODEL-BASED VS. MODEL-FREE REINFORCEMENT LEARNING

یادگیری تقویتی	
یادگیری تقویتی مبتنی بر مدل *Model-Based RL*	**یادگیری تقویتی رها از مدل** *Model-Free RL*
○ ابتدا در MDP کنش می‌کنیم و T و R را یاد می‌گیریم. ○ سپس «تکرار ارزش» یا «تکرار سیاست» را با T و R یاد گرفته شده اجرا می‌کنیم.	○ به یادگیری T و R نیاز نداریم. ○ از روش‌های ارزیابی V^{π} (تابع ارزش برای یک سیاست ثابت π) بدون دانستن T و R استفاده می‌کنیم: ○ ارزیابی مستقیم ○ یادگیری تفاضل زمانی ○ یا از روش‌های یادگیری π^*, Q^*, V^* بدون دانستن T و R استفاده می‌کنیم: ○ یادگیری Q
○ **مزیت:** استفاده‌ی کارآمد از داده‌ها ○ **عیب:** نیاز به ساخت یک مدل برای T و R	

یادگیری تقویتی منفعل / فعال

PASSIVE VS. ACTIVE REINFORCEMENT LEARNING

یادگیری تقویتی	
یادگیری تقویتی منفعل *Passive RL*	**یادگیری تقویتی فعال** *Active RL*
○ سیاست عامل ثابت است. ○ هدف، یادگیری سودمندی حالت‌ها یا سودمندی زوج (حالت، کنش) است.	○ عامل باید سیاست را نیز یاد بگیرد: یادگیری آنچه باید انجام دهد (نیازمند **اکتشاف** در مقابل بهره‌برداری)

هوش مصنوعی پیشرفته

یادگیری تقویتی

۲

یادگیری تقویتی مبتنی بر مدل

یادگیری تقویتی مبتنی بر مدل

مثال

MODEL-BASED REINFORCEMENT LEARNING

یادگیری تقویتی

یادگیری تقویتی مبتنی بر مدل
Model-Based RL

- ابتدا در MDP کنش می‌کنیم و T و R را یاد می‌گیریم.
- سپس «تکرار ارزش» یا «تکرار سیاست» را با T و R یاد گرفته شده اجرا می‌کنیم.

- **مزیت**: استفاده‌ی کارآمد از داده‌ها
- **عیب**: نیاز به ساخت یک مدل برای T و R

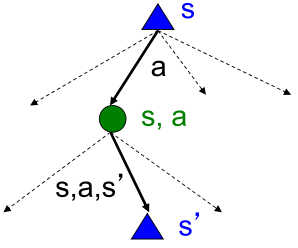
یادگیری تقویتی مبتنی بر مدل

MODEL-BASED REINFORCEMENT LEARNING

۱ یادگیری مدل

یادگیری مدل تجربی از طریق انجام آزمایش

- برآمدها را برای هر (s,a) شمارش کنید.
- حاصل را نرمال‌سازی کنید تا $T(s,a,s')$ محاسبه شود.
- پاداش $R(s,a,s')$ را هنگام آزمایش (s,a,s') کشف کنید.



۲ حل کردن MDP

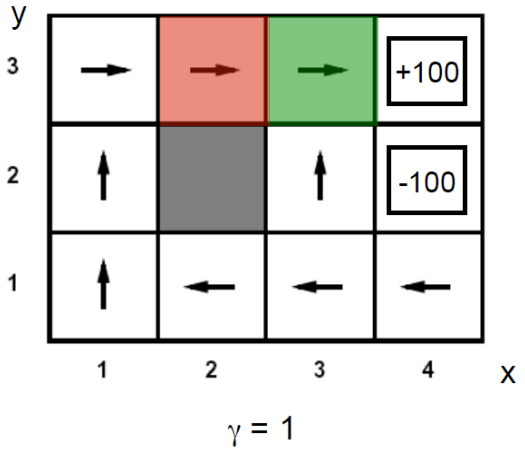
حل کردن MDP بر اساس مدل یادگیری شده

- مثلاً با استفاده از «تکرار ارزش» یا «تکرار سیاست»

یادگیری تقویتی مبتنی بر مدل

مثال: یادگیری مدل در یادگیری مبتنی بر مدل

LEARNING THE MODEL IN MODEL-BASED REINFORCEMENT LEARNING



Episode 1

```
----------------
(1,1) up -1
(1,2) up -1
(1,2) up -1
(1,3) right -1
(2,3) right -1
(3,3) right -1
(3,2) up -1
(3,3) right -1
(4,3) exit +100
(done)
```

Episode 2

```
----------------
(1,1) up -1
(1,2) up -1
(1,3) right -1
(2,3) right -1
(3,3) right -1
(3,2) up -1
(4,2) exit -100
(done)
```

$T(\langle 3,3\rangle, \text{right}, \langle 4,3\rangle) = 1 / 3$

$T(\langle 2,3\rangle, \text{right}, \langle 3,3\rangle) = 2 / 2$

یادگیری مدل در یادگیری مبتنی بر مدل

تخمین احتمالات

LEARNING THE MODEL IN MODEL-BASED LEARNING

می‌خواهیم $P(x)$ را از روی نمونه‌ها تخمین بزنیم:

نمونه‌ها $\quad x_i \sim P(x)$

تخمین $\quad \hat{P}(x) = \text{count}(x)/k$

می‌خواهیم $P(s'|s,a)$ را از روی نمونه‌ها تخمین بزنیم:

نمونه‌ها $\quad s_0, a_0, s_1, a_1, s_2, \ldots$

تخمین $\quad \hat{P}(s'|s,a) = \dfrac{\text{count}(s_{t+1} = s', a_t = a, s_t = s)}{\text{count}(a_t = a, s_t = s)}$

هوش مصنوعی پیشرفته

یادگیری تقویتی

۳

یادگیری تقویتی رها از مدل

یادگیری تقویتی رها از مدل

مثال

MODEL-FREE REINFORCEMENT LEARNING

یادگیری تقویتی

یادگیری تقویتی رها از مدل
Model-Free RL

- به یادگیری T و R نیاز نداریم.
- از روش‌های ارزیابی V^{π}
  (تابع ارزش برای یک سیاست ثابت π)
  بدون دانستن T و R استفاده می‌کنیم:
  - ارزیابی مستقیم
  - یادگیری تفاضل زمانی
- یا از روش‌های یادگیری π^*, Q^*, V^*
  بدون دانستن T و R استفاده می‌کنیم:
  - یادگیری Q

یادگیری تقویتی رها از مدل

روش ارزیابی مستقیم

DIRECT EVALUATION

روش ارزیابی مستقیم

اجرای سیاست

به صورت تکراری، سیاست π را اجرا کنید.

تخمین ارزش

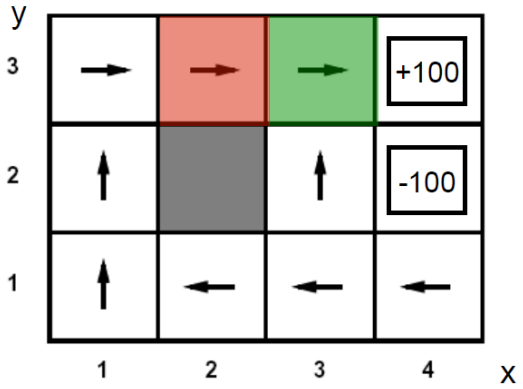
ارزش حالت S را تخمین بزنید.

متوسط گیری بر مجموع پاداش‌های تخفیف‌یافته انباشته از حالت S به بعد، بر روی همه‌ی زمان‌هایی که حالت S ملاقات شده است.

یادگیری تقویتی رها از مدل

روش ارزیابی مستقیم: مثال

DIRECT EVALUATION



$\gamma = 1, R = -1$

Episode 1

```
----------------
(1,1) up -1
(1,2) up -1
(1,2) up -1
(1,3) right -1
(2,3) right -1
(3,3) right -1
(3,2) up -1
(3,3) right -1
(4,3) exit +100
(done)
```

Episode 2

```
----------------
(1,1) up -1
(1,2) up -1
(1,3) right -1
(2,3) right -1
(3,3) right -1
(3,2) up -1
(4,2) exit -100
(done)
```

$V(2,3) \sim (96 + -103) / 2 = -3.5$
$V(3,3) \sim (99 + 97 + -102) / 3 = 31.3$

یادگیری رها از مدل

تخمین امید ریاضی

MODEL-FREE LEARNING

می‌خواهیم امید $f(x)$ را با توزیع $P(x)$ محاسبه کنیم:

$$E[f(x)] = \sum_x P(x) f(x)$$

تخمین مبتنی بر مدل
Model-Based Estimation

تخمین مستقیم امید ریاضی
از روی نمونه‌ها

$$E[f(x)] \approx \frac{1}{N} \sum_i f(x_i)$$

تخمین رها از مدل
Model-Free Estimation

ابتدا تخمین $P(x)$ و
سپس محاسبه‌ی امید ریاضی

$$x_i \sim P(x)$$

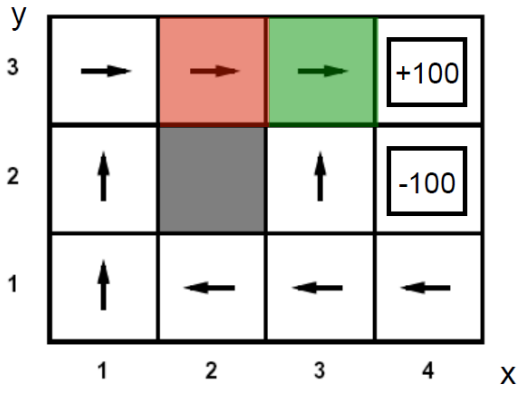
$$\hat{P}(x) = \text{count}(x)/N$$

$$E[f(x)] \approx \sum_x \hat{P}(x) f(x)$$

یادگیری تقویتی رها از مدل

روش ارزیابی مستقیم: محدودیت‌ها

DIRECT EVALUATION



- فرض می‌کنیم حالت آغازین، تصادفی باشد.
- فرض می‌کنیم ارزش حالت $(1,2)$ بر اساس سلسله اجراهای قبلی کاملاً معلوم باشد.
- حال، برای اولین بار با $(1,1)$ مواجه می‌شویم:
  آیا می‌توانیم برای تخمین ارزش $V(1,1)$
  بهتر از برآمدهای پاداش آن اجرا عمل کنیم؟

ارزیابی سیاست مبتنی بر نمونه

SAMPLE-BASED POLICY EVALUATION

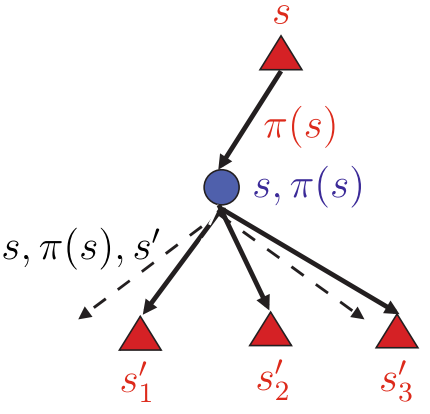
$$V_0^\pi(s) = 0$$

$$V_{i+1}^\pi(s) = \sum_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V_i^\pi(s')]$$

به مدل T و R نیاز نداریم: امید را با نمونه‌های s' (که از T بیرون کشیده شده‌اند) تقریب می‌زنیم.

$$sample_1 = R(s, \pi(s), s'_1) + \gamma V_i^\pi(s'_1)$$
$$sample_2 = R(s, \pi(s), s'_2) + \gamma V_i^\pi(s'_2)$$
$$\dots$$
$$sample_k = R(s, \pi(s), s'_k) + \gamma V_i^\pi(s'_k)$$

$$V_{i+1}^\pi \leftarrow \frac{1}{k} \sum_i sample_i$$



یادگیری تفاضل زمانی

TEMPORAL DIFFERENCE LEARNING

یک سیاست ثابت π را در نظر بگیرید؛ آن را اجرا کنید؛ $U^{\pi}(s)$ را یاد بگیرید.

$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} T(s, \pi(s); s') U^{\pi}(s')$$

معادله‌ی بلمن
Bellman Equation

به‌هنگام‌سازی TD تخمین سودمندی را به‌گونه‌ای تنظیم می‌کند که با معادله‌ی بلمن هماهنگ باشند:

$$U^{\pi}(s) \leftarrow U^{\pi}(s) + \alpha(R(s) + \gamma U^{\pi}(s') - U^{\pi}(s))$$

در اساس: استفاده از نمونه‌برداری از محیط به‌جای محاسبه‌ی دقیق مجموع

یادگیری تفاضل زمانی

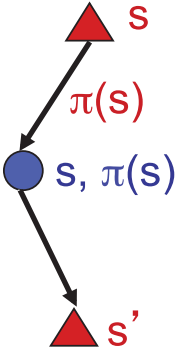
TEMPORAL DIFFERENCE LEARNING

یک سیاست ثابت π را در نظر بگیرید:

نمونه‌ی $V(s)$ $$sample = R(s, \pi(s), s') + \gamma V_i^{\pi}(s')$$

به‌روزرسانی $V(s)$ $$V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + \alpha(sample)$$

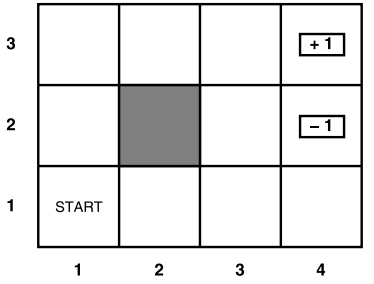
به‌روزرسانی $V(s)$ $$V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha(\underbrace{sample - V^{\pi}(s)}_{\text{تفاضل زمانی}})$$

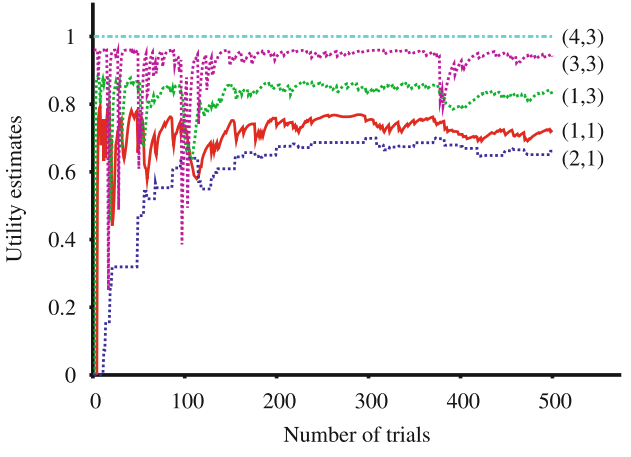


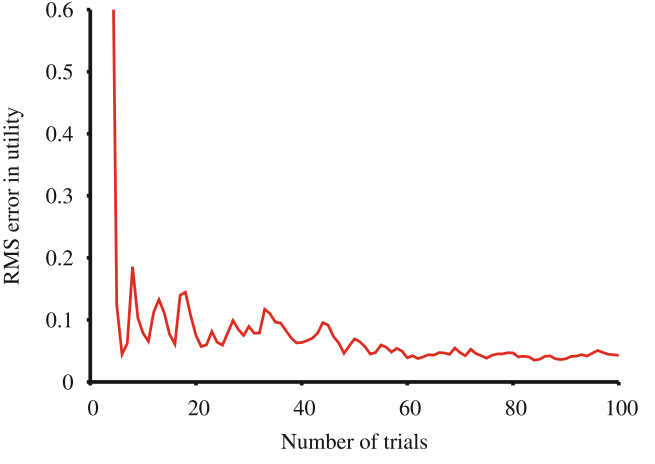
یادگیری تفاضل زمانی

کاربی: مثال دنیای ۴ در ۳

TEMPORAL DIFFERENCE LEARNING







یادگیری تفاضل زمانی

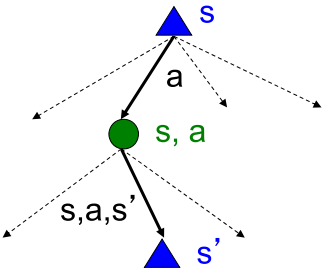
TEMPORAL DIFFERENCE LEARNING

یادگیری ارزش TD، یک روش رها از مدل برای ارزیابی سیاست است.

با این وجود، اگر بخواهیم ارزش‌ها را به یک سیاست (جدید) تبدیل کنیم، دوباره به مشکل می‌خوریم!:

$$\pi(s) = \arg\max_a Q^*(s,a)$$

$$Q^*(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^*(s')]$$



ایده: یادگیری مقادیر Q به‌طور مستقیم
(باعث می‌شود انتخاب کنش هم رها از مدل شود)

یادگیری Q

Q-LEARNING

یک نقطه ضعف یادگیری $U(s)$: هنوز به مدل گذر $T(s,a,s')$ برای تصمیم‌گیری نیاز داریم.

$Q(a,s)$ مقدار مورد انتظار سودمندی اگر کنش a را در حالت s انجام دهیم و سپس به طور بهینه کنش کنیم.

Bellman equation:

$$Q(a,s) = R(s) + \gamma \sum_{s'} T(s,\pi(s),s,) \max_{a'} Q(a',s')$$

Q-learning update:

$$Q(a;s) \leftarrow Q(a,s) + \alpha(R(s) + \max_{a'} Q(a',s') - Q(a,s))$$

Q-learning یک روش رها از مدل برای یادگیری و تصمیم‌گیری است

(این ویژگی خوب است، اما از این جهت که نمی‌توان از مدل برای محدود کردن مقادیر Q و ... استفاده کرد، بد است.)

یادگیری Q

Q-LEARNING

یادگیری Q: تکرار ارزش Q مبتنی بر نمونه

برای یادگیری ارزش‌های $Q^*(s,a)$:

- یک نمونه (s,a,s',r) را دریافت کنید.
- تخمین قبلی خود را در نظر بگیرید: $Q(s,a)$
- تخمین جدید نمونه‌ی خود را در نظر بگیرید:

$$sample = R(s,a,s') + \gamma \max_{a'} Q(s',a')$$

$$Q^*(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \max_{a'} Q^*(s',a')]$$

- تخمین جدید را در یک متوسط (در حال اجرا) وارد می‌کنیم:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha \cdot (sample)$$

یادگیری Q

خصوصیات

Q-LEARNING

یادگیری Q به سیاست بهینه همگرا می‌شود

- اگر عامل به‌اندازه‌ی کافی **اکتشاف** کند.
- اگر نرخ یادگیری به اندازه‌ی کافی کوچک شود.
- ... کوچک کردن نرخ یادگیری خیلی سریع نباشد.

❊ اساساً مهم نیست که کنش‌ها را چگونه انتخاب کنیم.

یادگیری برون–سیاست (off-policy)

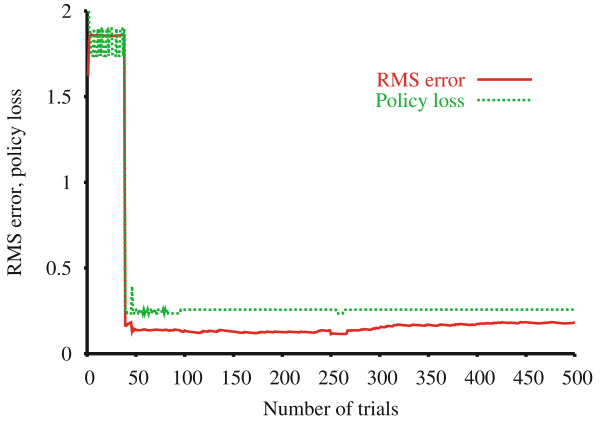
- یادگیری سیاست بهینه بدون دنبال کردن آن

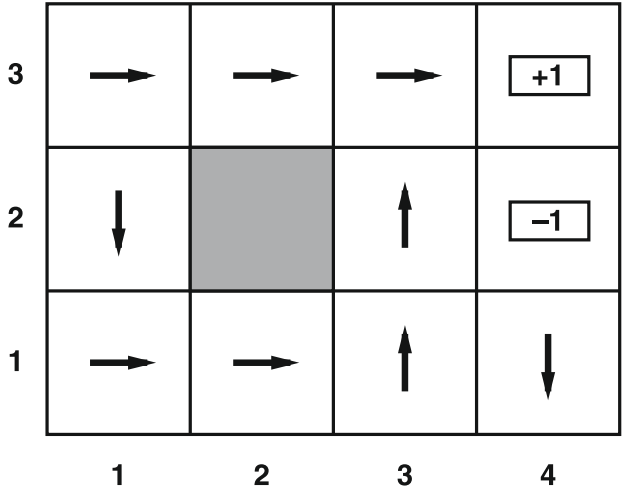
اکتشاف / بهره‌برداری

EXPLORATION / EXPLOITATION

عامل چگونه باید رفتار کند؟

آیا همیشه باید کنشی با بالاترین سودمندی مورد انتظار را انتخاب کند **(بهره‌برداری)**؟





Exploration vs. exploitation: occasionally try "suboptimal" actions!!

عامل باید گاه‌گاهی کنش‌های «زیر بهینه» را انتخاب کند **(اکتشاف)**.

اکتشاف / بهره‌برداری

چگونگی اجبار عامل به اکتشاف

EXPLORATION / EXPLOITATION

روش اول

کنش‌های تصادفی (ε–حریصانه)
Random Actions (ε-Greedy)

در هر گام زمانی، یک سکه می‌اندازیم:
- با احتمال ϵ تصادفی کنش می‌کنیم.
- با احتمال $1 - \epsilon$ بر اساس سیاست فعلی کنش می‌کنیم.

روش دوم

استفاده از تابع‌های اکتشاف
Using Exploration Functions

در طول زمان ϵ را کاهش می‌دهیم.

اکتشاف / بهره‌برداری

چگونگی اجبار عامل به اکتشاف: تابع اکتشاف: مثال

EXPLORATION / EXPLOITATION

یک تخمین ارزش و یک شمارش را در نظر می‌گیریم
و یک سودمندی خوش‌بینانه را بر می‌گردانیم: مثلاً:

$$f(u,n) = u + k/n$$

این تابع، بین میزان حریصانگی (اولویت به مقادیر بالای u)
و کنجکاوی (اولویت به مقادیر پایین n یعنی کنش‌هایی که قبلاً آزمایش نشده‌اند) توازن بر قرار می‌دهد.
* تابع اکتشاف در جهت افزایش u و کاهش n است.

$$Q_{i+1}(s,a) \leftarrow (1-\alpha)Q_i(s,a) + \alpha\left(R(s,a,s') + \gamma \max_a Q_i(s',a')\right)$$

تبدیل می‌شود به:

$$Q_{i+1}(s,a) \leftarrow (1-\alpha)Q_i(s,a) + \alpha\left(R(s,a,s') + \gamma \max_a f(Q_i(s',a'), N(s',a'))\right)$$

$N(s,a)$ تعداد دفعات آزمایش کنش a در حالت s

هوش مصنوعی پیشرفته

یادگیری تقویتی

۴

تعمیم در یادگیری تقویتی

تقریب تابع

FUNCTION APPROXIMATION

برای مسائل واقعی، نمی‌توانیم U یا Q را به‌صورت یک جدول بازنمایی کنیم.

معمولاً از تقریب تابعی خطی استفاده می‌کنیم:

$$\hat{U}_\theta(s) = \theta_1 f_1(s) + \theta_2 f_2(s) + \cdots + \theta_n f_n(s)$$

از یک جستجوی گام به گام گرادیانی برای تغییر پارامترهای θ استفاده می‌شود:

$$\theta_i \leftarrow \theta_i + \alpha[R(s) + \gamma \hat{U}_\theta(s') - \hat{U}_\theta(s)] \frac{\partial \hat{U}_\theta(s)}{\partial \theta_i}$$

$$\theta_i \leftarrow \theta_i + \alpha[R(s) + \gamma \max_{a'} \hat{Q}_\theta(a', s') - \hat{Q}_\theta(a, s)] \frac{\partial \hat{Q}_\theta(a, s)_\theta(s)}{\partial \theta_i}$$

اغلب در عمل بسیار کارآمد است، اما همگرایی آن تضمین‌شده نیست!

یادگیری تقویتی

مثال: بازی Pacman

PACMAN GAME

می‌خواهیم از طریق تجربه کشف کنیم که این حالت بد است.

در Q-learning ساده، چیزی در مورد این حالت و حالت‌های Q آن نمی‌دانیم.

حتی در مورد این!

یادگیری تقویتی

مثال: بازی Pacman (بازنمایی مبتنی بر ویژگی)

PACMAN GAME: FEATURE-BASED REPRESENTATION

یک حالت را توسط برداری از ویژگی‌ها توصیف می‌کنیم:
(ویژگی‌ها توابعی از حالت‌ها به اعداد حقیقی هستند که خواص مهم حالت را نشان می‌دهند.)

ویژگی‌های نمونه:

- فاصله تا نزدیک‌ترین شبح (ghost)
- فاصله تا نزدیک‌ترین نقطه
- تعداد شبح‌ها
- معکوس مجذور فاصله تا نقطه‌ها
- آیا پک‌من در تونل است (ویژگی دودویی 0/1)
- …

حتی می‌توان یک حالت Q (یعنی (s,a)) را با ویژگی‌ها توصیف کرد.
(مثلاً کنشی که پک‌من را به غذا نزدیک‌تر می‌کند)

تقریب تابع

توابع ویژگی خطی

LINEAR FEATURE FUNCTIONS

با استفاده از یک بازنمایی ویژگی،
می‌توانیم یک تابع Q (یا تابع ارزش V) را برای هرحالت بنویسیم (با تعدادی وزن):

$$V(s) = w_1 f_1(s) + w_2 f_2(s) + \cdots + w_n f_n(s)$$

$$Q(s,a) = w_1 f_1(s) + w_2 f_2(s) + \cdots + w_n f_n(s)$$

تقریب تابع

توابع ویژگی خطی

LINEAR FEATURE FUNCTIONS

$$Q(s,a) = w_1 f_1(s) + w_2 f_2(s) + \cdots + w_n f_n(s)$$

یادگیری Q با توابع ارزش Q خطی:

$$transition = (s, a, r, s')$$

$$difference = [r + \gamma \max_{a'} Q(s', a')] - Q(s, a)$$

$$Q(s,a) \leftarrow Q(s,a) + \alpha \cdot (difference)$$

$$w_i \leftarrow w_i + \alpha \cdot (difference) \cdot f_i(s,a)$$

تعبیر شهودی: وزن‌های ویژگی‌های فعال را تنظیم می‌کنیم

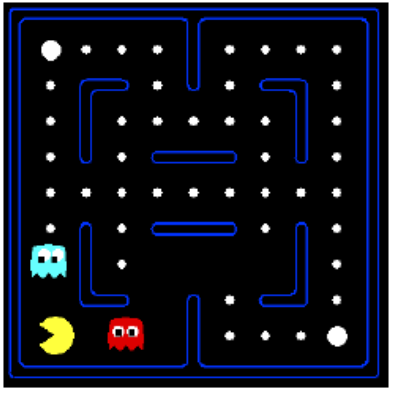
(مثلاً: اگر اتفاق بد غیرمنتظره‌ای بیفتد، همه‌ی حالت‌ها با آن ویژگی‌های حالت نامرجح می‌شوند.)

توجیه صوری: **کمترین مربعات برخط**

یادگیری تقویتی

مثال: بازی Pacman (توابع ویژگی خطی و تقریب تابع)

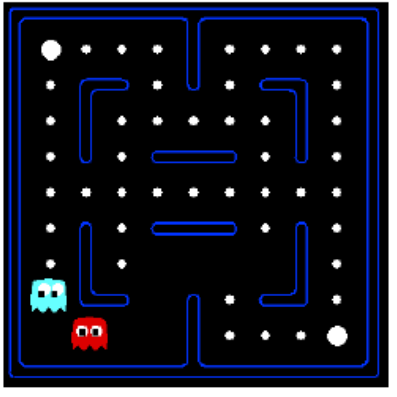
PACMAN GAME



$$Q(s,a) = w_{DOT} f_{DOT}(s,a) + w_{GST} f_{GST}(s,a)$$

$$w_{DOT} = 4.0 \quad w_{GST} = -1.0$$
$$f_{DOT}(s, \text{NORTH}) = 0.5$$
$$f_{GST}(s, \text{NORTH}) = 1.0$$

$$Q(s,a) = +1$$
$$R(s,a,s') = -500$$
$$error = -501$$

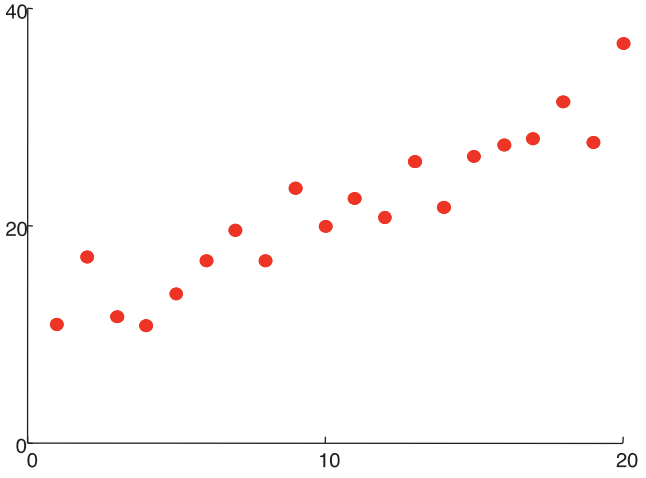


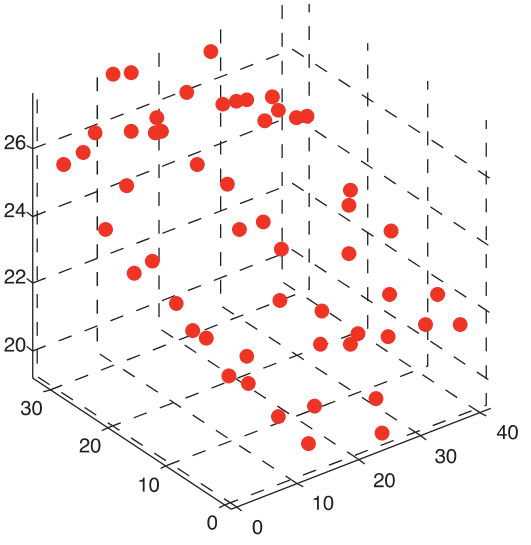
$$w_{DOT} \leftarrow 4.0 + \alpha \times (-501) \times 0.5 = 3.0$$
$$w_{GST} \leftarrow -1.0 + \alpha \times (-501) \times 1.0 = -3.0$$

تقریب تابع

رگرسیون خطی

LINEAR REGRESSION





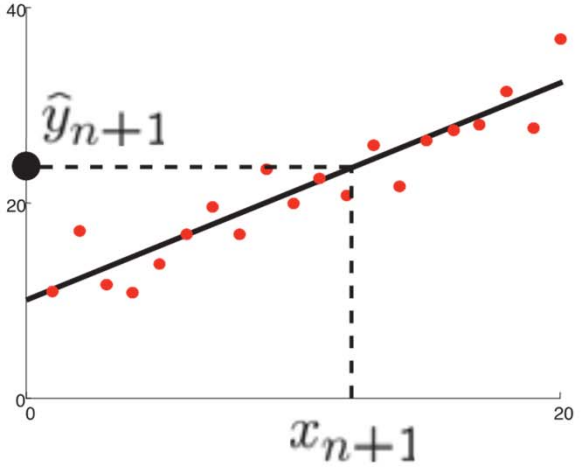
Given examples $(x_i, y_i)_{i=1...n}$

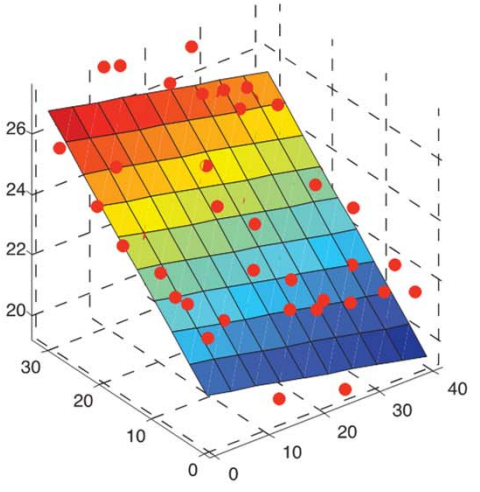
Predict y_{n+1} given a new point x_{n+1}

تقریب تابع

رگرسیون خطی

LINEAR REGRESSION





Prediction

$$\widehat{y}_i = w_0 + w_1 x_i$$

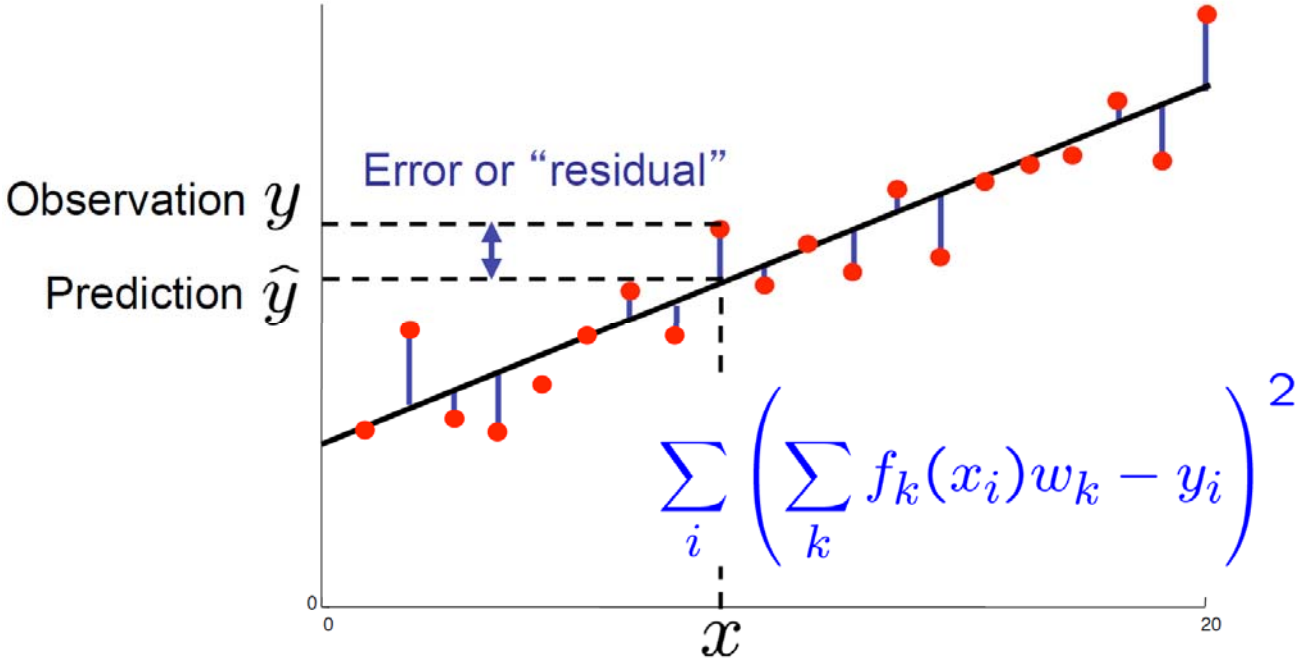
Prediction

$$\widehat{y}_i = w_0 + w_1 x_{i,1} + w_2 x_{i,2}$$

تقریب تابع

رگرسیون خطی: روش کمترین مربعات معمولی

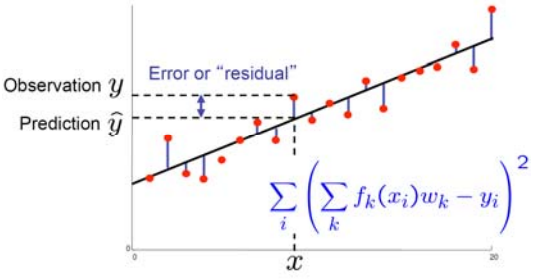
LINEAR REGRESSION: ORDINARY LEAST SQUARES (OLS)



تقریب تابع

رگرسیون خطی: روش کمترین مربعات معمولی: می‌نیمم‌سازی خطی

LINEAR REGRESSION: ORDINARY LEAST SQUARES (OLS)



$$E(w) = \frac{1}{2} \sum_i \left(\sum_k f_k(x_i) w_k - y_i \right)^2$$

$$\frac{\partial E}{\partial w_m} = \sum_i \left(\sum_k f_k(x_i) w_k - y_i \right) f_m(x_i)$$

$$E \leftarrow E + \alpha \sum_i \left(\sum_k f_k(x_i) w_k - y_i \right) f_m(x_i)$$

به‌روزرسانی ارزش توضیح داده شده:

$$w_i \leftarrow w_i + \alpha \, [error] \, f_i(s, a)$$

هوش مصنوعی پیشرفته

یادگیری تقویتی

۵

جستجوی سیاست

جستجوی سیاست

POLICY SEARCH

جستجوی سیاست *Policy Search*	یادگیری سیاست بهینه با جستجو روی زیرمجموعه‌ای از همه‌ی سیاست‌ها

مشکل: سیاست‌های مبتنی بر ویژگی که خوب کار می‌کنند،
اغلب آنهایی نیستند که بهترین تقریب‌ها را برای ارزش‌های V و Q ارائه می‌دهند.

راه‌حل: یادگیری سیاستی که پاداش‌ها را ماکزیمم می‌کند،
به‌جای یادگیری ارزشی که پاداش‌ها را پیش‌بینی می‌کند.

جستجوی سیاست

POLICY SEARCH

جستجوی سیاست ساده
Simple Policy Search

با یک تابع ارزش V یا Q خطی آغازین شروع می‌کنیم .
وزن‌ها را بالا و پایین می‌کنیم
و نگاه می‌کنیم که آیا سیاست نسبت به قبل بهتر شده است یا خیر.

مشکلات:

- چگونه متوجه شویم که سیاست بهتر شده است؟
- چند اپیزود از نمونه‌ها باید اجرا شود؟ (بسیار)
- اگر تعداد ویژگی‌ها زیاد باشد، عملی نیست!

جستجوی سیاست پیشرفته
Advanced Policy Search

یک سیاست اتفاقی (نرم) را می‌نویسیم:

$$\pi_w(s) \propto e^{\sum_i w_i f_i(s,a)}$$

مشتق دریافتی‌ها نسبت به پارامتر وزن w را می‌توان به صورت کارآمدی تخمین زد.

هوش مصنوعی پیشرفته

یادگیری تقویتی

۶

کاربردهای یادگیری تقویتی

یادگیری تقویتی

جستجوی سیاست (مثال: هلی‌کوپتر خودمختار)

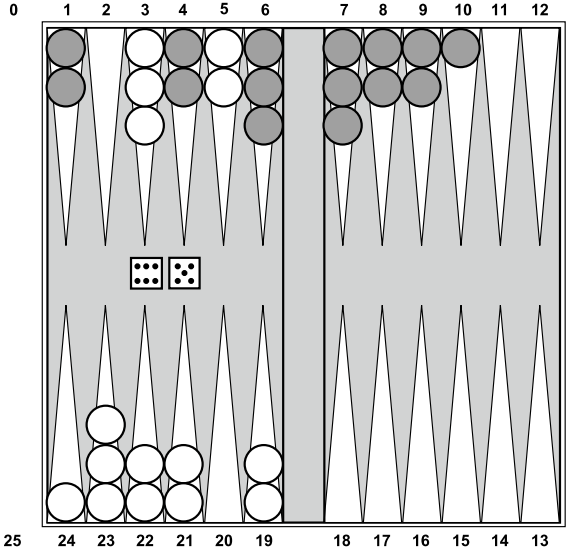
EXAMPLE: AUTONOMOUS HELICOPTER

پاداش: منفی مربع انحراف از حالت مطلوب

یادگیری تقویتی

مثال: بازی تخته‌نرد

EXAMPLE: BACKGAMMON



پاداش برای برد/باخت تنها درحالت‌های پایانی؛ برای سایر موارد پاداش صفر است.

TDGammon تابع تخمین ارزش $\hat{U}(s)$ را یاد می‌گیرد که به صورت یک شبکه‌ی عصبی سه‌لایه بازنمایی شده است.
(در ترکیب با یک جستجوی EXPECTIMINIMAX تا عمق ۲ یا ۳ تبدیل به یکی از سه بازیکن برتر جهان می‌شود!)

مسائل تصمیم مارکوف و یادگیری تقویتی

جمع‌بندی

MDPs AND RL

MDP را می‌دانیم

- امکان محاسبه‌ی دقیق V^*, Q^*, π^*
- امکان ارزیابی یک سیاست ثابت π

تکنیک

برنامه‌ریزی پویا مبتنی بر مدل
Model-Based Dynamic Programming

- تکرار ارزش
- ارزیابی سیاست

MDP را نمی‌دانیم

۱) تخمین MDP و سپس حل آن

تکنیک

یادگیری تقویتی مبتنی بر مدل
Model-Based RL

۲) تخمین ارزش یا سیاست

تکنیک

یادگیری تقویتی رها از مدل
Model-Free RL

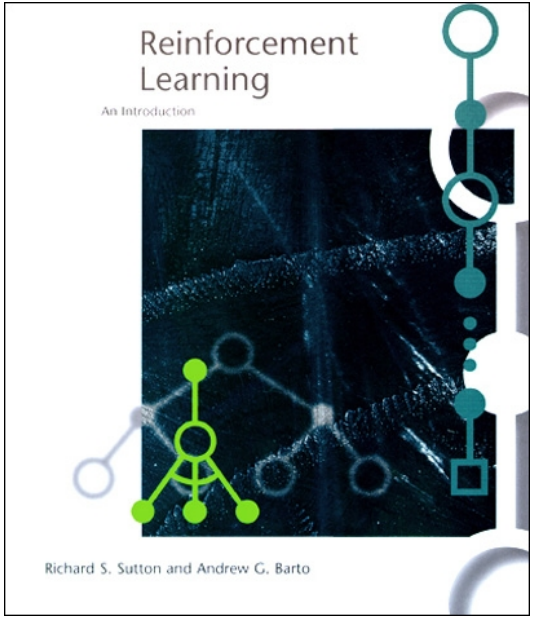
- یادگیری V
- یادگیری Q

هوش مصنوعی پیشرفته

یادگیری تقویتی

منابع، مطالعه، تکلیف

منبع کمکی



Richard S. Sutton, Andrew G. Barto,
Reinforcement Learning: An Introduction,
MIT Press, 1998.

Draft of unpublished new edition (2017):
https://webdocs.cs.ualberta.ca/~sutton/book/the-book-2nd.html