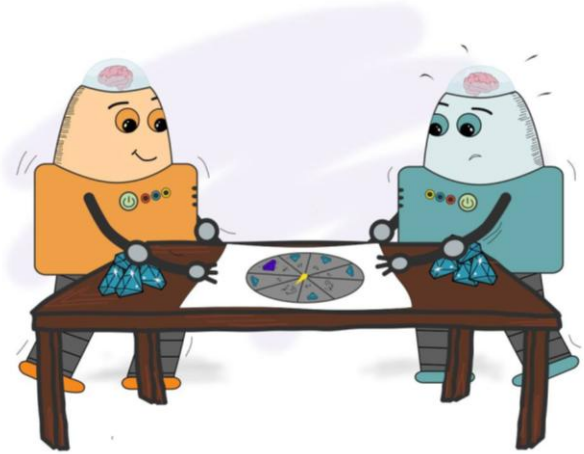
مبانی و کاربردهای هوش مصنوعی

جستجوی خصمانه 2 – عدم قطعیت (فصل 5.2 الی 5.5)

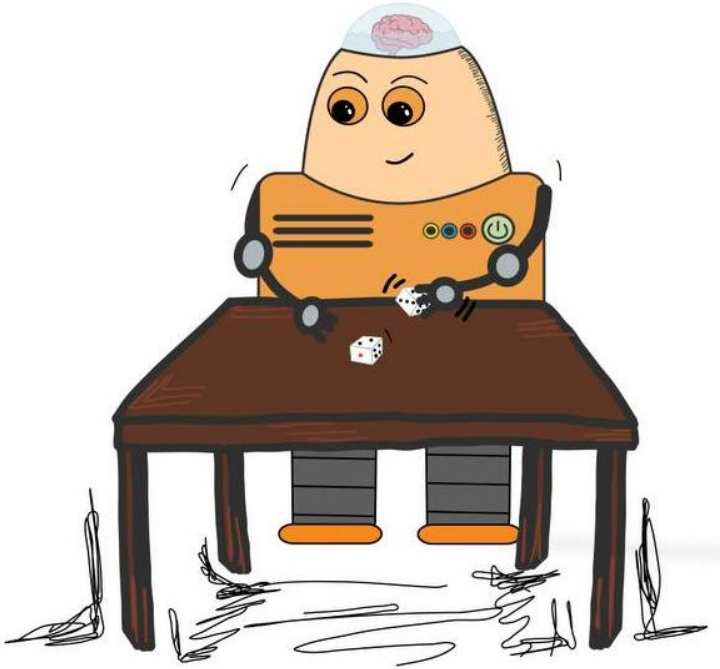


مدرس: مهدی جوانمردی

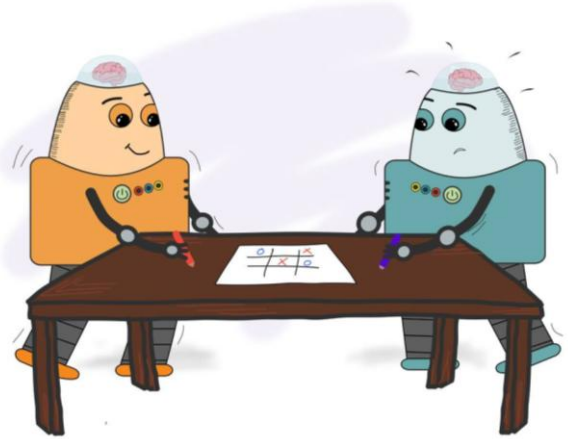
دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر

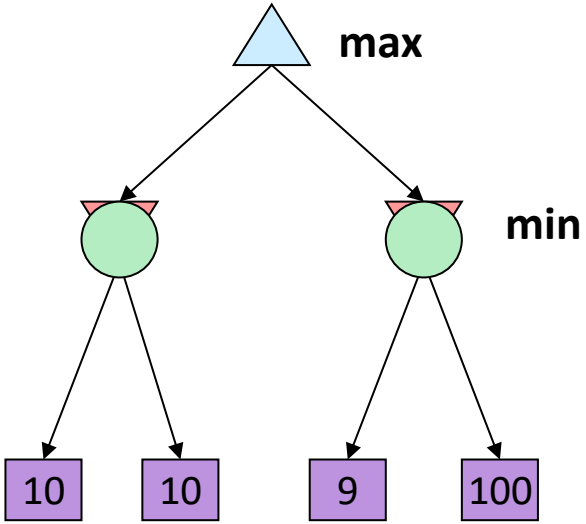
(الهام گرفته از محتوای درس هوش مصنوعی دانشگاه برکلی)

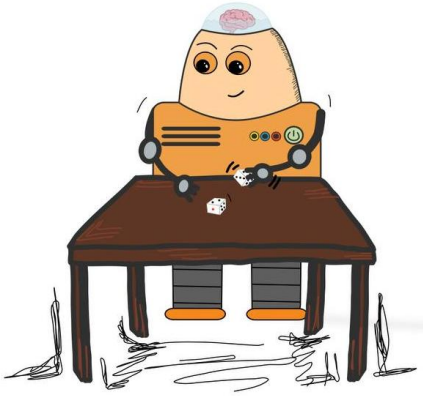
عدم قطعیت در خروجی



بدترین حالت در برابر حالت میانگین







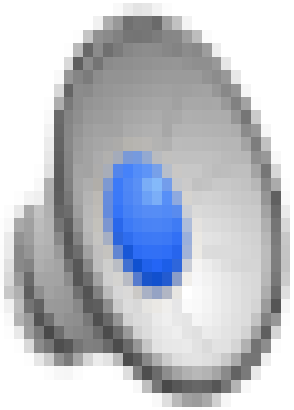
ایده: نتایج غیرقطعی توسط شانس کنترل می‌شوند، نه خصومت!

جستجوی Expectimax

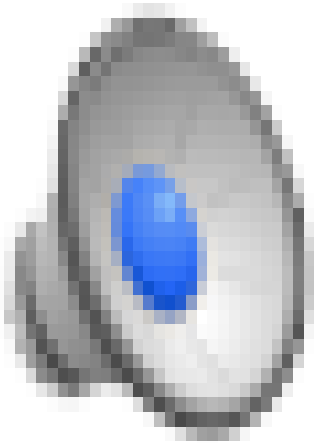
- چرا ما نتیجه یک عمل را نمی‌دانیم؟
  - تصادفی بودن محض: پرتاب تاس
  - رقبای غیر قابل پیش‌بینی: روح‌ها تصادفی پاسخ می‌دهند
  - اعمال ممکن است شکست بخورند:
    موقع حرکت ربات ممکن است چرخ‌ها لیز بخوردند
- مقادیر باید نمایانگر نتایج میانگین باشند (Expectimax)، نه بدترین حالت نتایج (Minimax)

- جستجوی Expectimax: محاسبه امتیاز میانگین در صورت بازی بهینه
  - گره‌های بیشینه مشابه جستجوی کمینه-بیشینه
  - گره‌های شانس شبیه به گره‌های کمینه ولی نتیجه غیرقطعی
  - محاسبه سودمندی مورد انتظار
  - یعنی میانگین وزنی (مورد انتظار) فرزندان را می‌گیریم
- در ادامه، یاد خواهیم گرفت که چگونه مسائل نتایج نامعلوم را به صورت فرآیندهای تصمیم‌گیری مارکوف رسمی‌سازی کنیم.

دموی Minimax در برابر Expectimax (Minimax)



دموی Minimax در برابر Expectimax (Expectimax)



شبه کد Expectimax

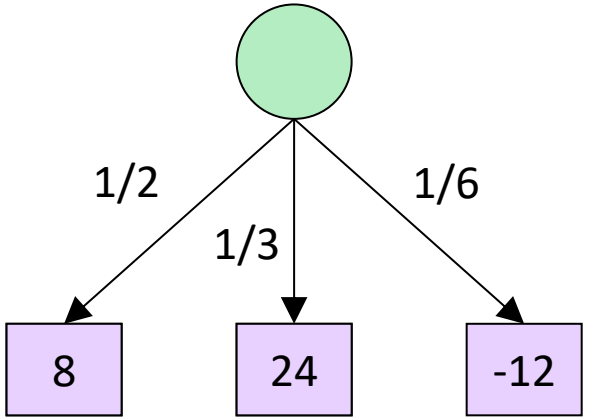
```
def value(state):
        if the state is a terminal state: return the state's utility
        if the next agent is MAX: return max-value(state)
        if the next agent is EXP: return exp-value(state)
```

```
def max-value(state):
        initialize v = -∞
        for each successor of state:
                v = max(v, value(successor))
        return v
```

```
def exp-value(state):
        initialize v = 0
        for each successor of state:
                p = probability(successor)
                v += p * value(successor)
        return v
```

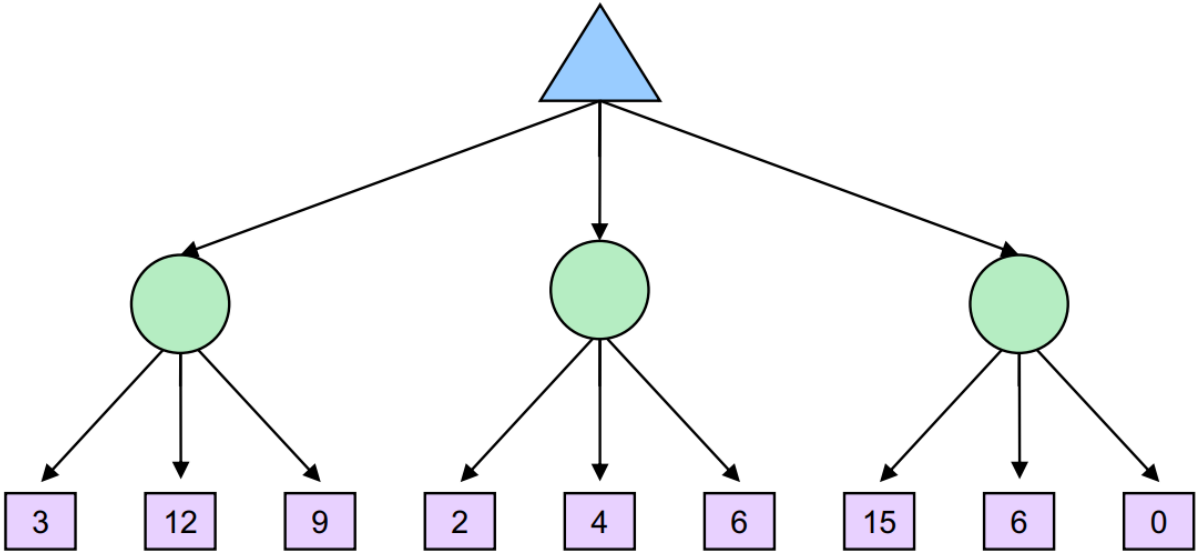
شبه کد Expectimax

```
def exp-value(state):
    initialize v = 0
    for each successor of state:
        p = probability(successor)
        v += p * value(successor)
    return v
```

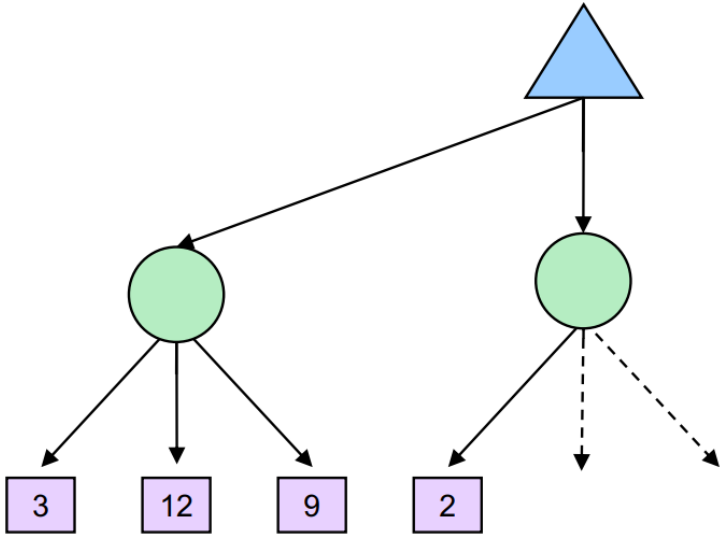


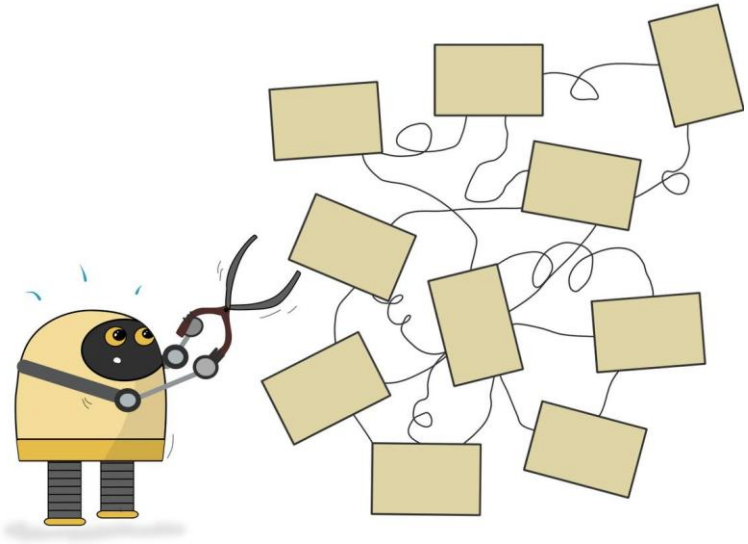
$$v = (1/2)\,(8) + (1/3)\,(24) + (1/6)\,(-12) = 10$$

مثال Expectimax

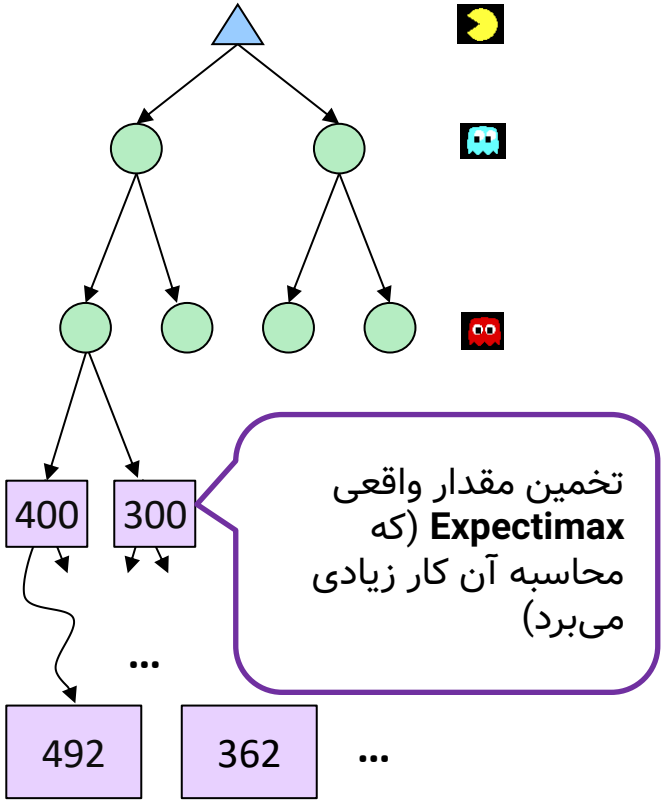


هرس Expectimax ؟

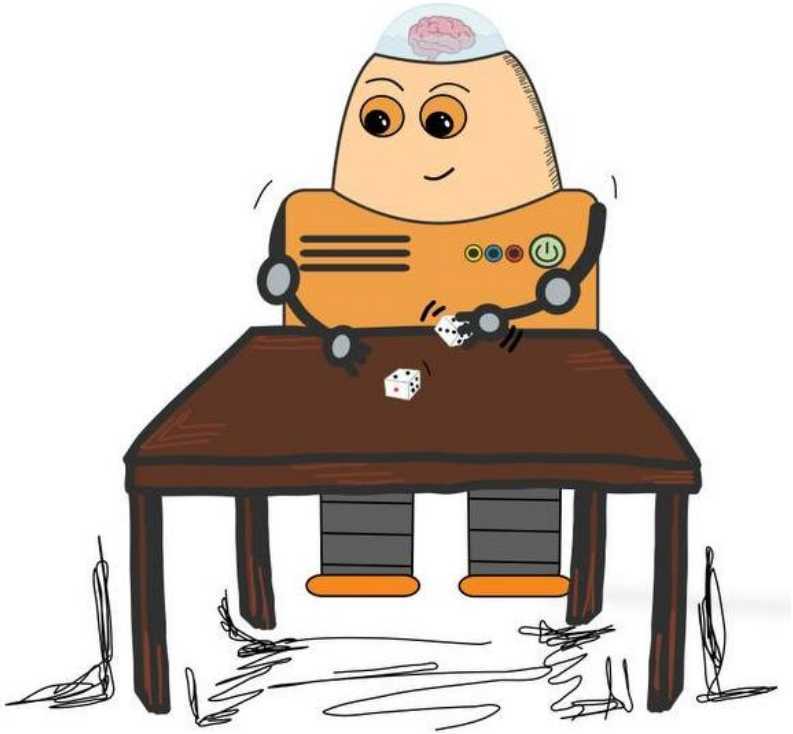




Expectimax با عمق محدود (Depth-Limited)



احتمالات (Probabilities)



یادآوری: احتمالات

- یک متغیر تصادفی نمایانگر یک اتفاق است که نتیجه‌ی آن نامعلوم است.
- یک توزیع احتمال یک تخصیص وزن به نتایج است.
- تعدادی از قوانین احتمال (بعدا، تعدادی بیشتر)
  - احتمالات همیشه غیرمنفی هستند.
  - مجموع احتمالات همه‌ی نتایج، یک است.
- همینطور که شواهد بیشتر می‌شود، احتمالات ممکن است تغییر کند.
  - P(T=heavy) = 0.25, P(T=heavy | Hour=8am) = 0.60
  - بعدا در مورد روش‌های استدلال و به روزرسانی احتمالات بیشتر صحبت خواهیم کرد.

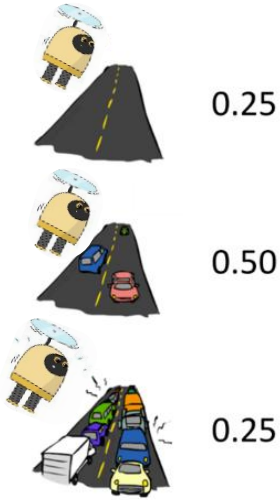
مثال: ترافیک در آزادراه

- متغیر تصادفی: T = آیا ترافیک است؟
- نتایج: T در {heavy، light، none}
- توزیع:

P(T=none) = 0.25

P(T=light) = 0.50

P(T=heavy) = 0.25

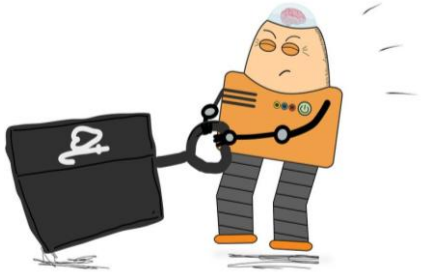


0.25

0.50

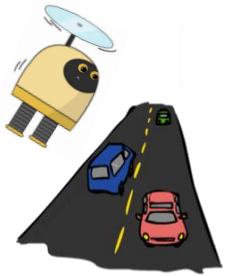
0.25

یادآوری: امید ریاضی



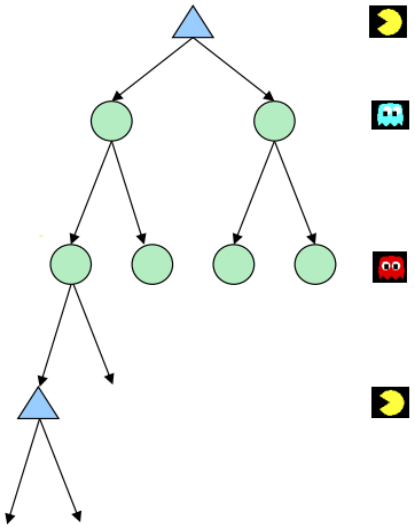
- امید ریاضی یک متغیر تصادفی، میانگین نتایج با وزن‌های توزیع احتمال است.
- مثال: چقدر طول می‌کشد تا به فرودگاه برسیم؟

زمان:	20 min		30 min		60 min		
	x	+	x	+	x		35 min
احتمال:	0.25		0.50		0.25		



از چه احتمالاتی استفاده شود؟

- در جستجوی Expectimax ما یک مدل احتمالاتی از نحوه رفتار رقیب (یا محیط) در هر حالت داریم.
  - مدل می‌تواند یک توزیع ساده یکسان باشد (پرتاب تاس)
  - مدل می‌تواند پیچیده باشد و به مقدار زیادی محاسبات نیاز داشته باشد
  - ما از گره شانس (Chance Node) برای هر نتیجه‌ی خارج از کنترلمان استفاده میکنیم: رقیب یا محیط
  - ممکن است مدل بگوید که اعمال خصمانه با احتمالی ممکن است <- expectimax

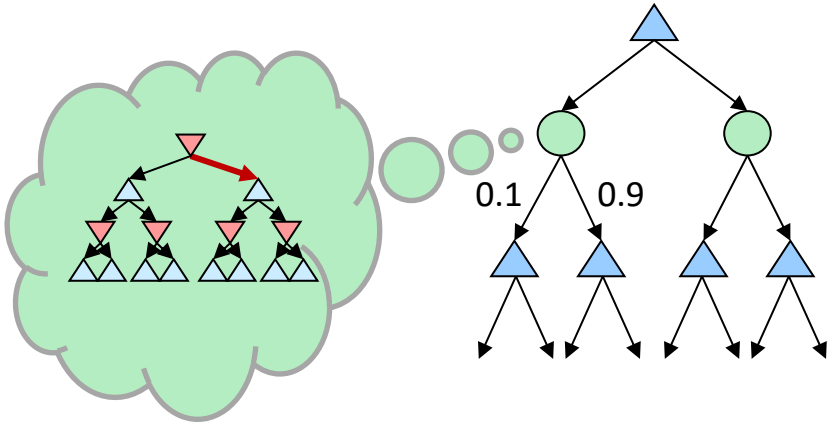


- فعلا، فرض کنید هر گره شانس به صورت جادویی توزیع احتمالات نتایج را دارد

داشتن یک باور احتمالاتی در مورد اعمال یک عامل دیگر به معنای این نیست که عامل سکه می‌اندازد!

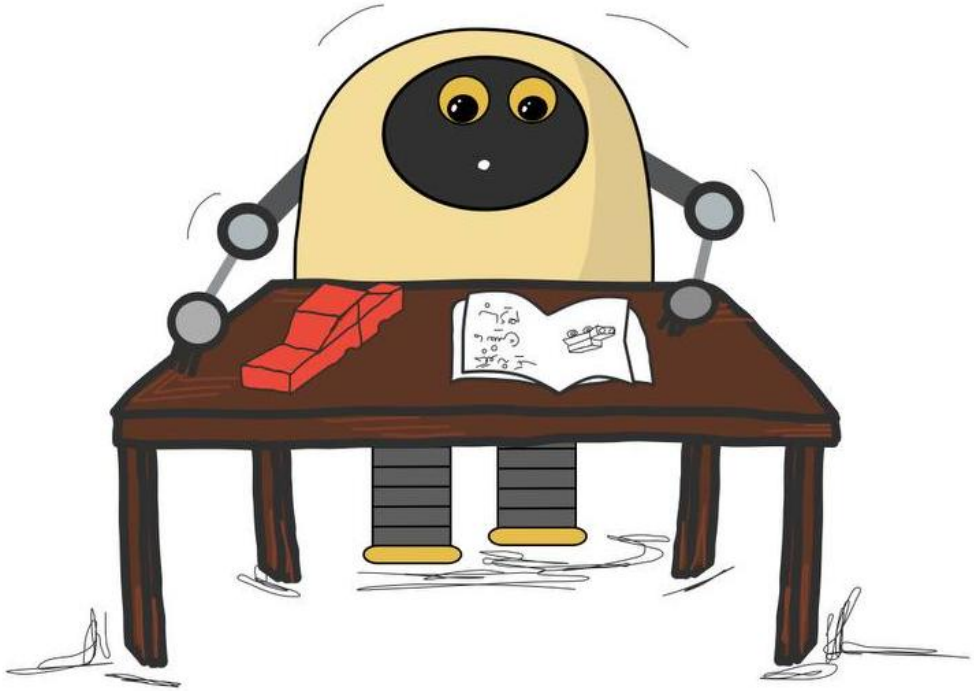
آزمونک: احتمالات آگاهانه (Informed Probabilities)

- فرض کنید می‌دانید که حریف شما در اصل یک کمینه-بیشینه با عمق دو اجرا می‌کند، 80% مواقع از نتایج آن استفاده می‌کند، در غیر اینصورت تصادفی حرکت می‌کند
- سوال: از کدام جستجوی درختی استفاده می‌کنید؟
- جواب: Expectimax



- برای پیدا کردن احتمال **هر** گره شانس، شما باید یک شبیه‌سازی از رقیب اجرا کنید
- این تیپ شبیه‌سازی‌ها به سرعت کند می‌شود
- حتی بدتر اگر شبیه‌سازی رقیب از خودتان را شبیه‌سازی کنید...

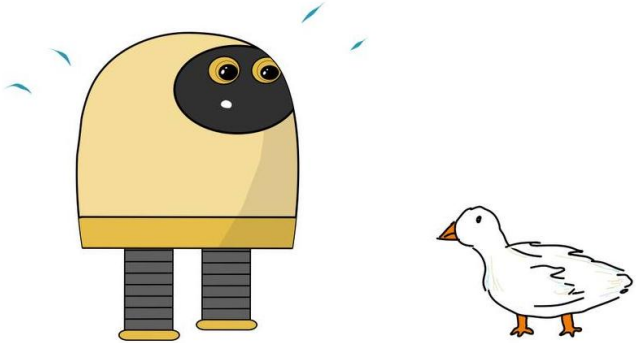
مفروضات مدل‌سازی
(Modeling Assumptions)



خطرات خوش‌بینی و بدبینی

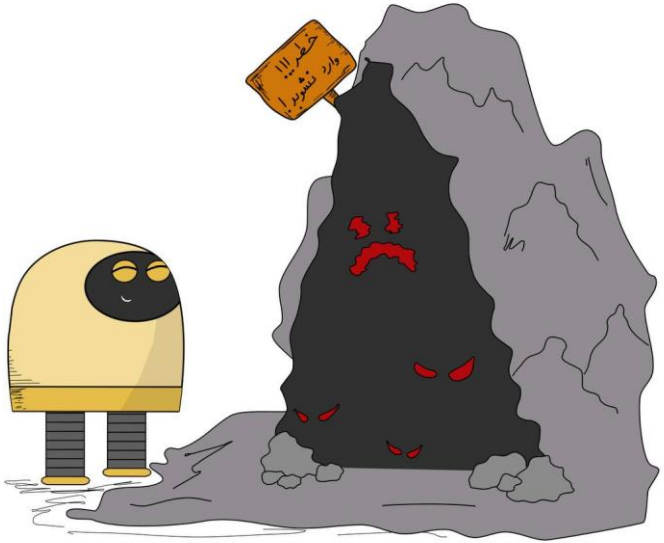
بدبینی خطرناک

بدترین حالت را در نظر گرفتن وقتی که محتمل نیست

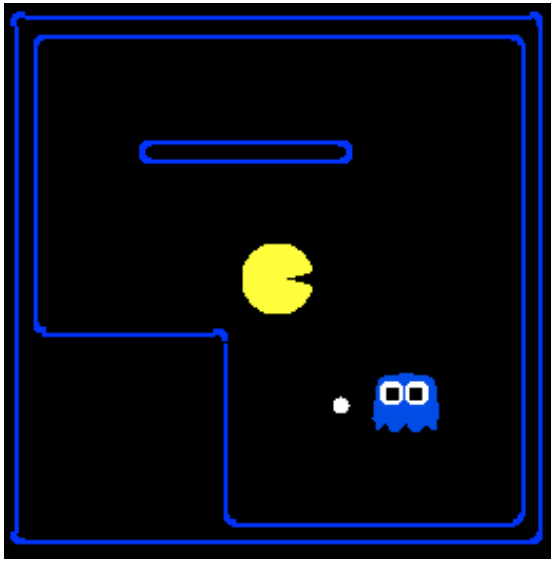


خوش‌بینی خطرناک

شانس را در نظر گرفتن وقتی که جهان خصمانه است



مفروضات در برابر واقعیت



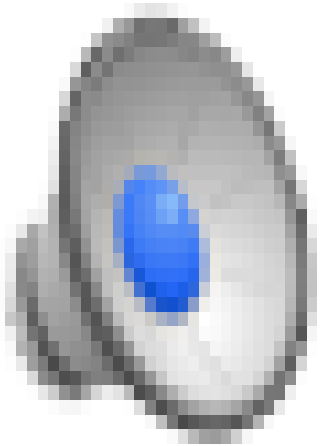
	Adversarial Ghost	Random Ghost
Minimax Pacman		
Expectimax Pacman		

نتایج بدست آمده از 5 بازی

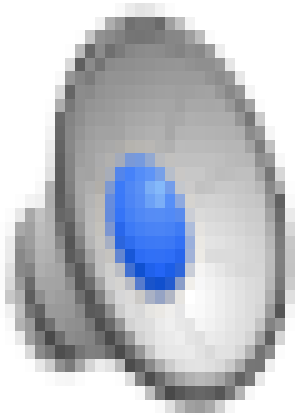
استفاده پکمن از جستجوی عمق 4 با تابع ارزیابی‌ای که از مشکلات پرهیز کند
استفاده روح از جستجوی عمق 2 با تابع ارزیابی‌ای که به دنبال پکمن است

[Demos: world assumptions (L7D3,4,5,6)]

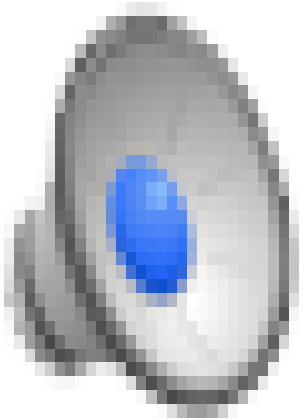
World Assumptions Random Ghost – Expectimax Pacman دموی



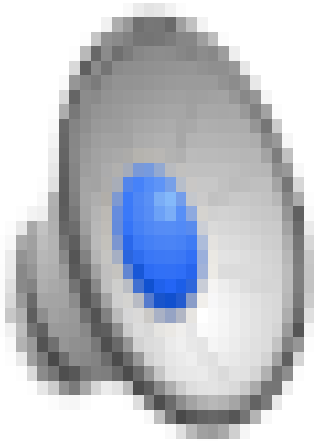
World Assumptions Adversarial Ghost – Minimax Pacman دموی



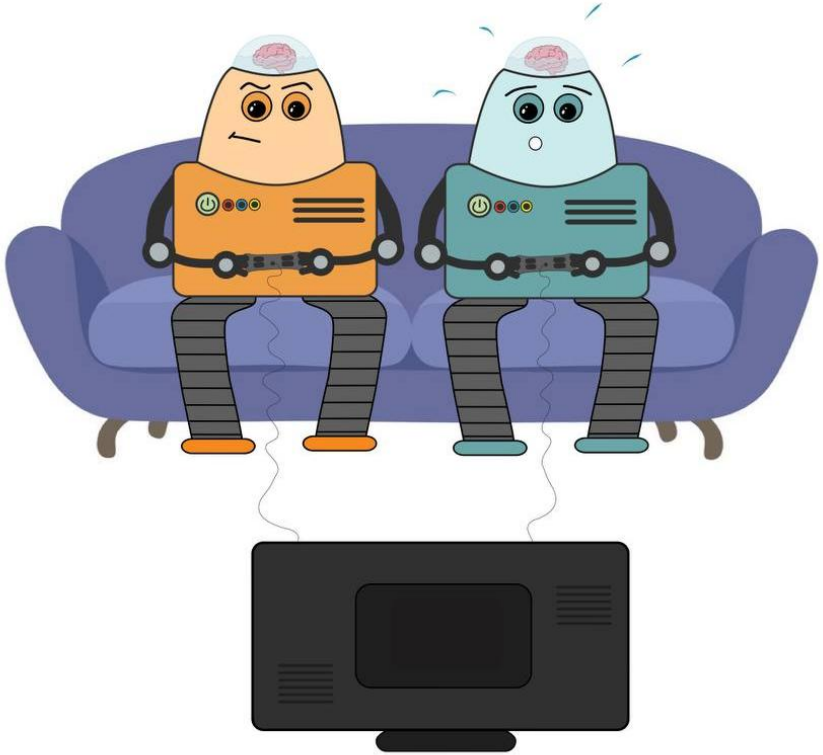
World Assumptions Adversarial Ghost – Expectimax Pacman ویدیوی نمایشی



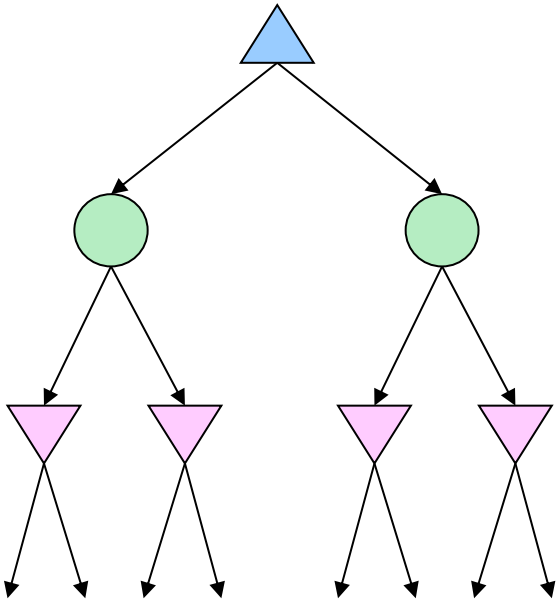
World Assumptions Random Ghost – Minimax Pacman ویدیوی نمایشی



انواع دیگر بازی‌ها



گونه‌هایی با لایه‌های مختلط
(Mixed Layer Types)



- مثل تخته نرد (Backgammon)
- Expectiminimax
  - محیط، یک بازیکن "عامل تصادفی" اضافه است که بعد از هر عامل کمینه-بیشینه حرکت می‌کند
  - هر گره ترکیب مناسب فرزندانش را محاسبه می‌کند

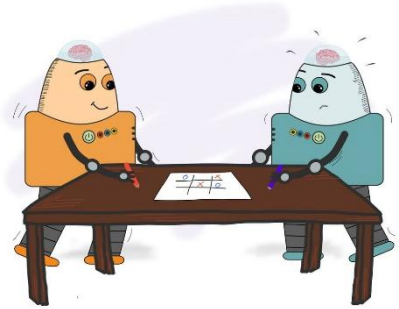
مثال: تخته نرد

- پرتاب‌های تاس اندازه b را افزایش می‌دهد: b=21 برای دو عدد تاس
  - تخته نرد ≈ 20 حرکت قانونی
  - عمق 2 = $20 \times (21 \times 20)^3 = 1.2 \times 10^9$
- همینطور که عمق افزایش می‌یابد، احتمال رسیدن به یک گره جستجوی داده شده کم می‌شود
  - بنابراین مفید بودن جستجو کاهش می‌یابد
  - بنابراین کم کردن عمق کمتر آسیب می‌زند

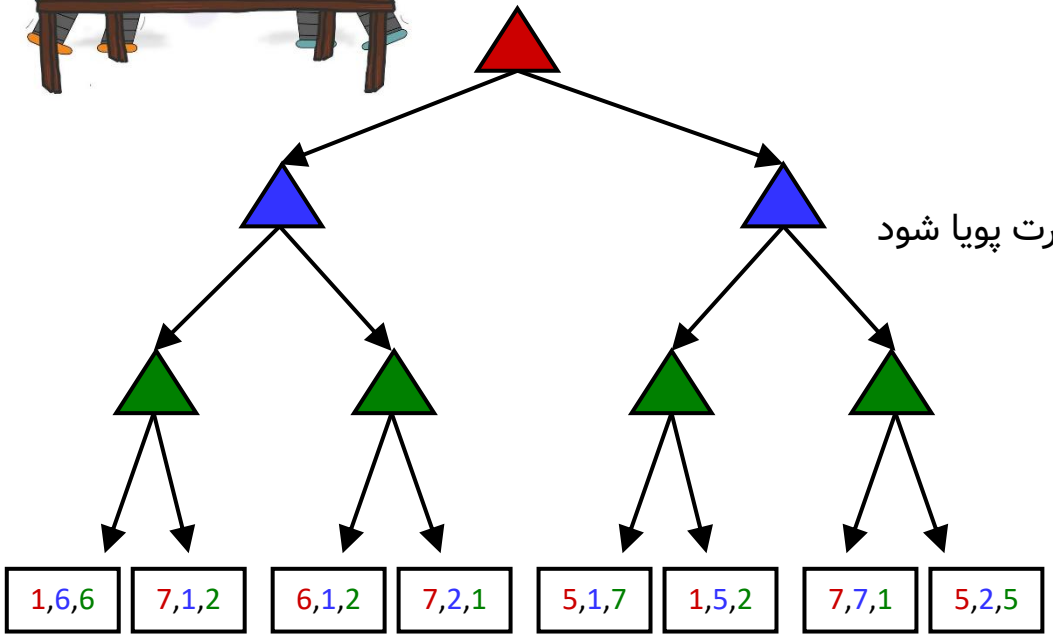
هوش مصنوعی تاریخی: TDGammon از جستجوی با عمق 2 + تابع ارزیابی خیلی خوب + یادگیری تقویتی استفاده می‌کند (by IBM):
بازی در سطح قهرمانی جهانی

IBM 1992

سودمندی‌های چند عامله



- اگر بازی مجموع صفر نبود یا چند بازیکن داشت چی؟
- تعمیم کمینه-بیشینه:
  - خروجی‌ها ردیف‌های سودمندی دارند
  - مقادیر گره‌ها نیز ردیف‌های سودمندی دارند
  - هر بازیکنی بخش خودش را بیشینه می‌کند
  - می‌تواند باعث رشد همکاری و رقابت به صورت پویا شود



دفعه بعد: MDPها!