Mesterséges intelligencia előadássorozat

Az előadás diái az AIMA könyvre épülve (http://aima.cs.berkeley.edu) készültek a University of California, Berkeley mesterséges intelligencia kurzusának anyagainak felhasználásával (http://ai.berkeley.edu).

These slides are based on the AIMA book (http://aima.cs.berkeley.edu) and were adapted from the AI course material of University of California, Berkeley (http://ai.berkeley.edu).

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Mesterséges Intelligencia és Rendszertervezés Tanszék

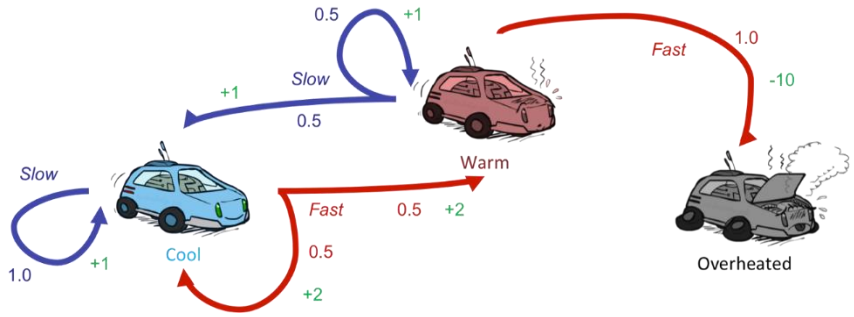
VIMIAC16 2025/26/I.

Megerősítéses tanulás

Előadó: Dr. Hullám Gábor

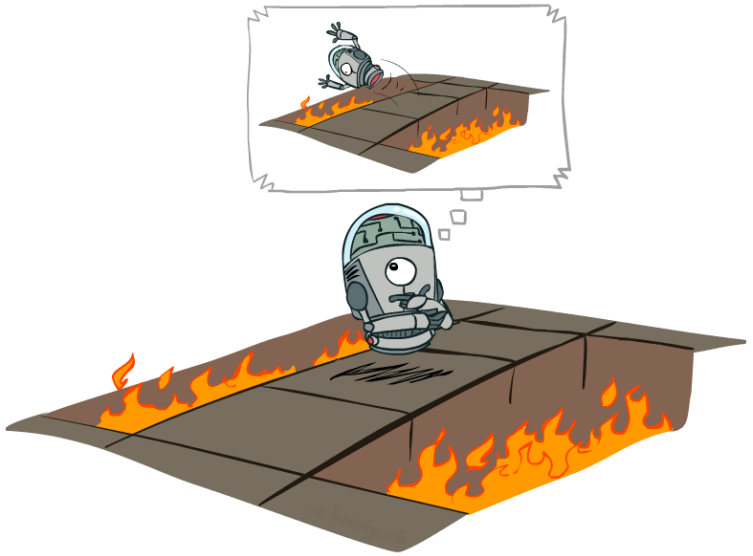
Megerősítéses tanulás (reinforcement learning)

- Feltételezzünk egy Markov-döntési folyamatot (MDF):
  - állapothalmaz $s \in S$
  - cselekvések halmaza (állapotonként) A
  - állapotátmenet-modell T(s,a,s')
  - jutalomfüggvény R(s,a,s')
- $\pi(s)$ eljárásmódot keressük



- Új fordulat: nem ismerjük T-t vagy az R-et
  - Nem tudjuk, hogy mely állapotok jók, vagy hogy mit tesznek a cselekvések.
  - A cselekvéseket és állapotokat ténylegesen ki kell próbálni a tanuláshoz

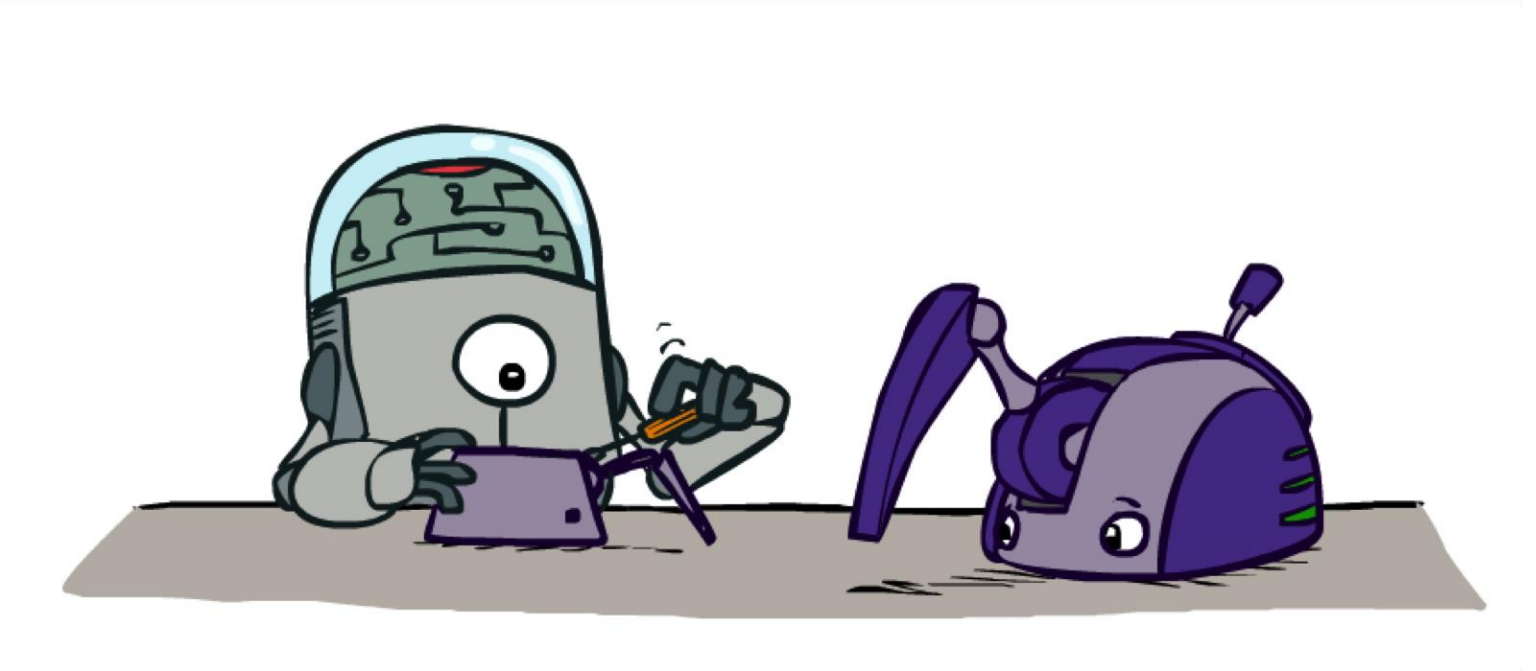
Offline (MDF) vs. Online (RL)



Offline megoldás

Online tanulás

Modellalapú tanulás

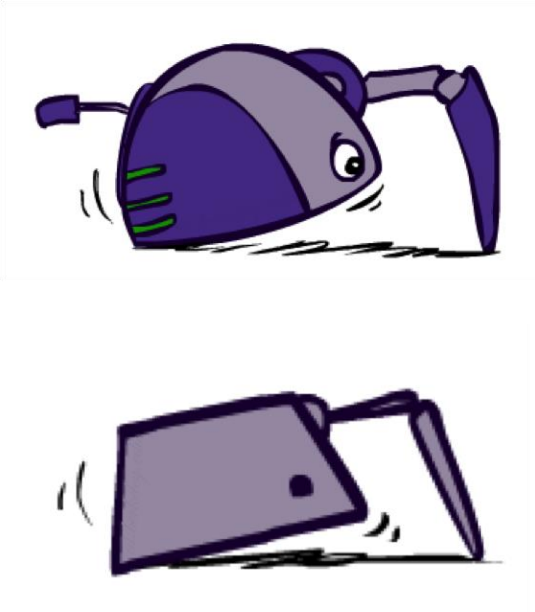


Modellalapú tanulás

- Modellalapú ötlet:
  - Tapasztalatokon alapuló közelítő modell tanulása
  - Értékek megoldása úgy, mintha a megtanult modell helyes lenne.

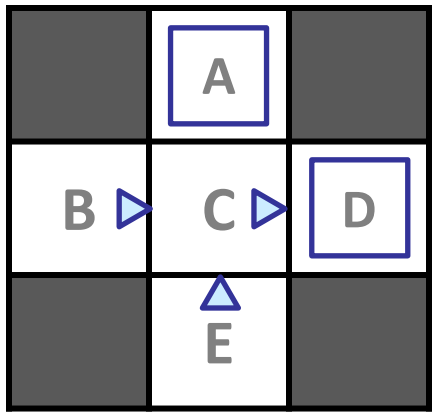
- 1. lépés: Empirikus MDF-modell tanulása
  - Számoljuk meg az s' kimeneteleket minden s, a –ra
  - Normalizáljuk ezeket, hogy becslést adjunk $\widehat{T}(s, a, s')$ -re
  - Fedezzük fel az egyes $\widehat{R}(s, a, s')$ jutalmakat (s, a, s') átmenetek megfigyelése esetén

- 2. lépés: A tanult MDF megoldása
  - Például használjuk az értékiterációt



Példa: Modellalapú tanulás

Rögzített π
eljárásmód



$\gamma = 1$

Megfigyelt epizódok (tanulás)

1. Epizód

B, east, C, -1
C, east, D, -1
D, exit, x, +10

2. Epizód

B, east, C, -1
C, east, D, -1
D, exit, x, +10

3. Epizód

E, north, C, -1
C, east, D, -1
D, exit, x, +10

4. Epizód

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Tanult modell

$\hat{T}(s, a, s')$

T(B, east, C) = 1.00
T(C, east, D) = 0.75
T(C, east, A) = 0.25
…

$\hat{R}(s, a, s')$

R(B, east, C) = -1
R(C, east, D) = -1
R(D, exit, x) = +10
…

Példa: Várható életkor

Cél: A tanulók várható életkorának kiszámítása

Ismert P(A)

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

P(A) nélkül, inkább gyűjtsünk mintákat [a_1, a_2, … a_N]

Ismeretlen P(A): "Modellalapú"

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

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

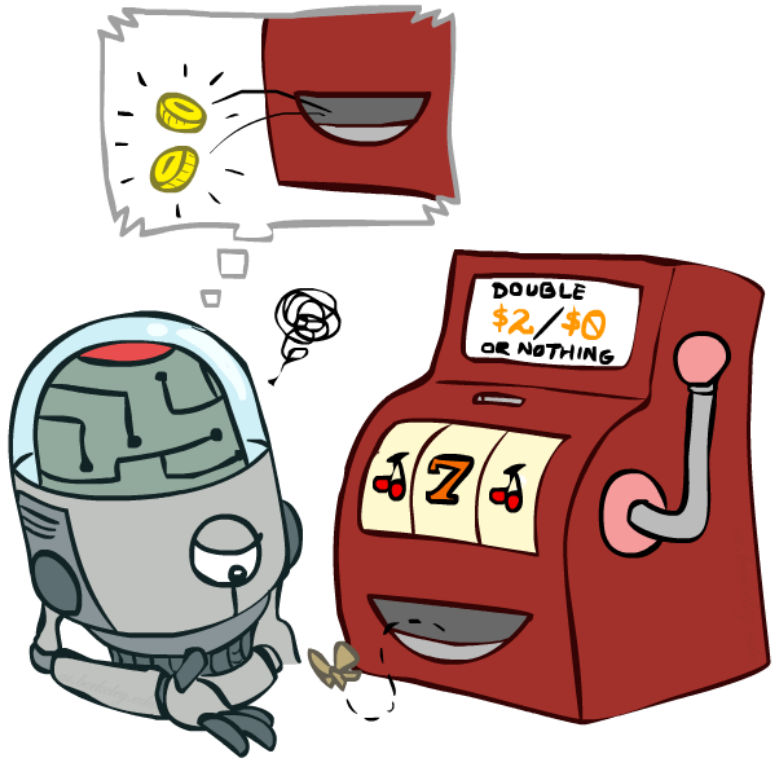
Ez miért működik? Mert végül megtanuljuk a megfelelő modellt.

Ismeretlen P(A): "Modelmentes"

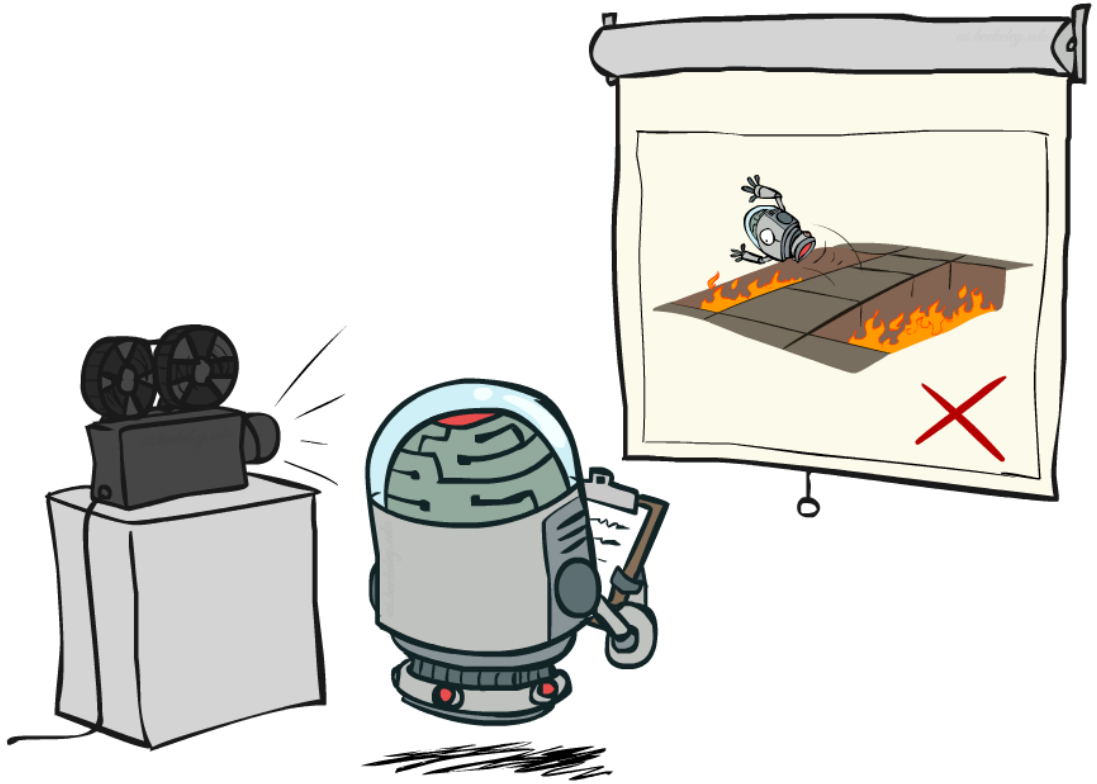
$$E[A] \approx \frac{1}{N} \sum_i a_i$$

Miért működik ez? Mert a minták a megfelelő frekvenciával jelennek meg.

Modellmentes tanulás

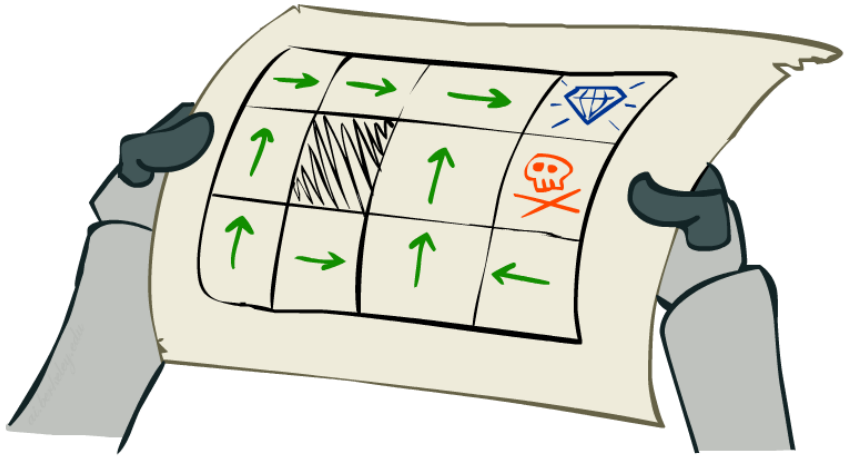


Passzív megerősítéses tanulás



Passzív megerősítéses tanulás

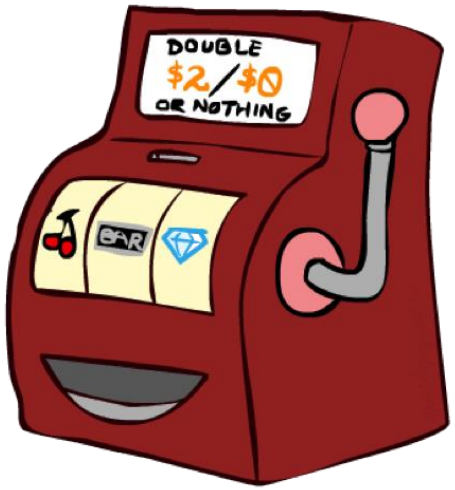
- Egyszerűsített feladat: eljárásmód értékelés
  - Bemenet: rögzített eljárásmód $\pi(s)$
  - Nem ismerjük az átmeneteket T(s,a,s')
  - Nem ismerjük a jutalmakat R(s,a,s')
  - Cél: tanuljuk meg az egyes állapotok hasznosságát



- Ekkor:
  - Nincs választási lehetőség a cselekvések tekintetében
  - Eljárás végrehajtása és modell tanulása
  - Ez nem offline tervezés!
  - Ténylegesen cselekvéseket kell végrehajtani és megfigyelni a következményeit

Közvetlen kiértékelés

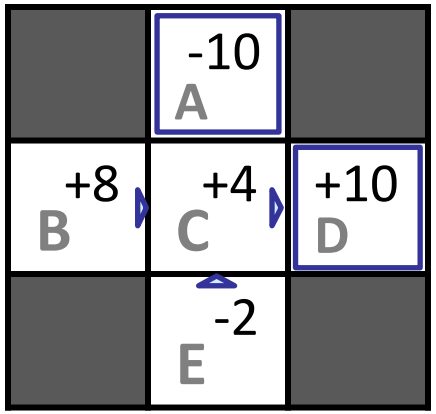
- Cél: az egyes állapotok hasznosságának kiszámítása a π eljárásmód szerint
- Ötlet: Vegyük a megfigyelt mintaértékek átlagát
  - Cselekvés π szerint
  - Minden alkalommal, amikor belép egy állapotba, rögzíteni kell, hogy mennyi lett a leszámított jutalmak összege
  - Minták átlagolása



Közvetlen kiértékelés problémái

- Előny
  - Könnyen érthető
  - Nem igényli T, R ismeretét
  - Végül kiszámítja a helyes átlagértékeket, csak az átmenetek felhasználásával.
- Hátrány
  - Az állapotok közötti kapcsolatokra vonatkozó információkat elpazarolja
  - Minden állapotot külön kell megtanulni
  - Így hosszú időbe telik megtanulni

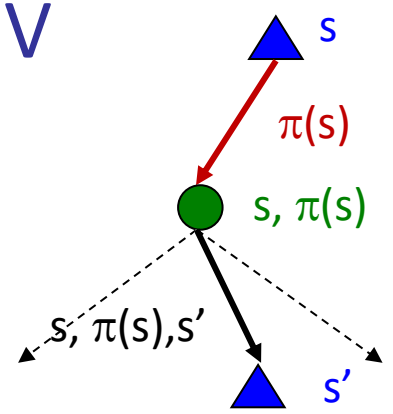
Hasznosságok



Ha mind B és E a C-be vezet akkor hogyan lehet a hasznosságuk különböző?

Eljárásmód kiértékelést használhatnánk, de…

- Egyszerűsített Bellman frissítéssel számítható V (=U) egy rögzített eljárásra:
  - $V_0^\pi(s) = 0$rdulóban cseréljük ki a U-t egy egylépéses előre tekintéssel

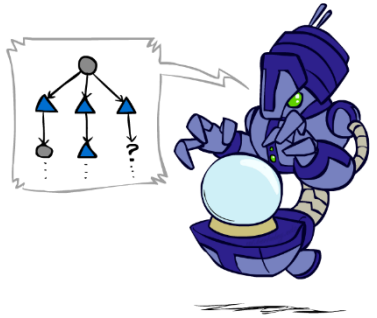


$$V_{k+1}^\pi(s) \leftarrow \sum_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V_k^\pi(s')]$$

  - Ez a megközelítés teljes mértékben kihasználta az állapotok közötti kapcsolatokat.
  - Sajnos ehhez szükségünk van T-re és R-re!

- Kulcskérdés: hogyan tudjuk elvégezni a V frissítését T és R ismerete nélkül?
  - Más szóval, hogyan vegyünk egy súlyozott átlagot a súlyok ismerete nélkül?

Mintaalapú eljárásmód kiértékelés



- Ezen átlagok kiszámításával szeretnénk javítani a V-ra vonatkozó becslésünket

$$V_{k+1}^{\pi}(s) \leftarrow \sum_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V_k^{\pi}(s')]$$

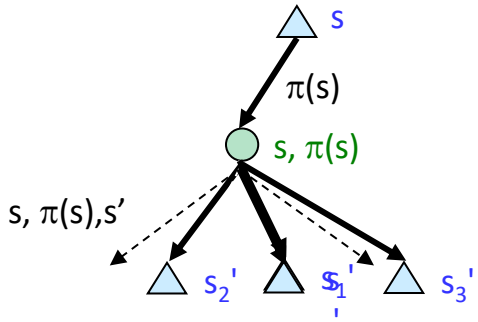
- Ötlet: Vegyünk mintákat az s' kimenetekből (a művelet elvégzésével!) és átlagoljuk őket.

$$sample_1 = R(s, \pi(s), s'_1) + \gamma V_k^{\pi}(s'_1)$$

$$sample_2 = R(s, \pi(s), s'_2) + \gamma V_k^{\pi}(s'_2)$$

$$sample_n = R(s, \pi(s), s'_n) + \gamma V_k^{\pi}(s'_n)$$

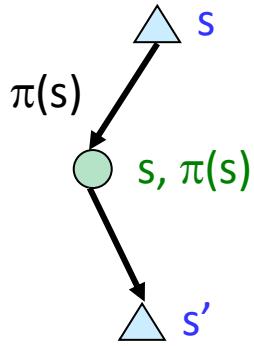
$$V_{k+1}^{\pi}(s) \leftarrow \frac{1}{n} \sum_i sample_i$$



Majdnem jó, de nem állíthatjuk vissza az időt, hogy egymás után több s' mintát kapjunk s-ből!

Időbeli különbség tanulás (Temporal Difference Learning)

- Ötlet: tanuljuk minden megtapasztalt mintából
  - Frissítjük V(s)-t minden egyes állapotátmenetnél (s, a, s', r)
  - A jellemző (gyakori) hasznosságok gyakrabban hozzájárulnak a frissítéshez



- A hasznosságok időbeli különbség szerinti tanulása
  - Az eljárásmód még mindig rögzített, még mindig értékelést végez!
  - Az hasznosságok változtatása a követő hasznosságok irányába: mozgóátlaggal

Egy mintából V(s):

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

Frissítjük V(s)-t:

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

Átalakítva:

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

Exponenciális mozgóátlag

Exponenciális mozgóátlag

$$\bar{x}_n = (1-\alpha) \cdot \bar{x}_{n-1} + \alpha \cdot x_n$$

- A futó interpolációs frissítés:
  - Az új minták fontosabbak

$$\bar{x}_n = \frac{x_n + (1-\alpha) \cdot x_{n-1} + (1-\alpha)^2 \cdot x_{n-2} + \dots}{1 + (1-\alpha) + (1-\alpha)^2 + \dots}$$

  - Elfelejti a múltat (a távoli múltbeli értékek amúgy is tévesek voltak)

- A csökkenő tanulási ráta (alfa) konvergáló átlagokat adhat.

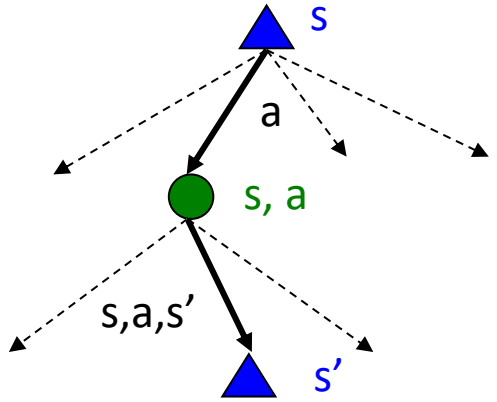
Az időbeli különbség (IK) tanulás problémái

- Az IK tanulás egy modellmentes módja az eljárásmódok értékelésének, amely mozgóátlagokkal utánozza a Bellman-frissítéseket.
- Ha azonban a hasznosságokat(új) eljárásmóddá akarjuk alakítani, akkor gond adódik:

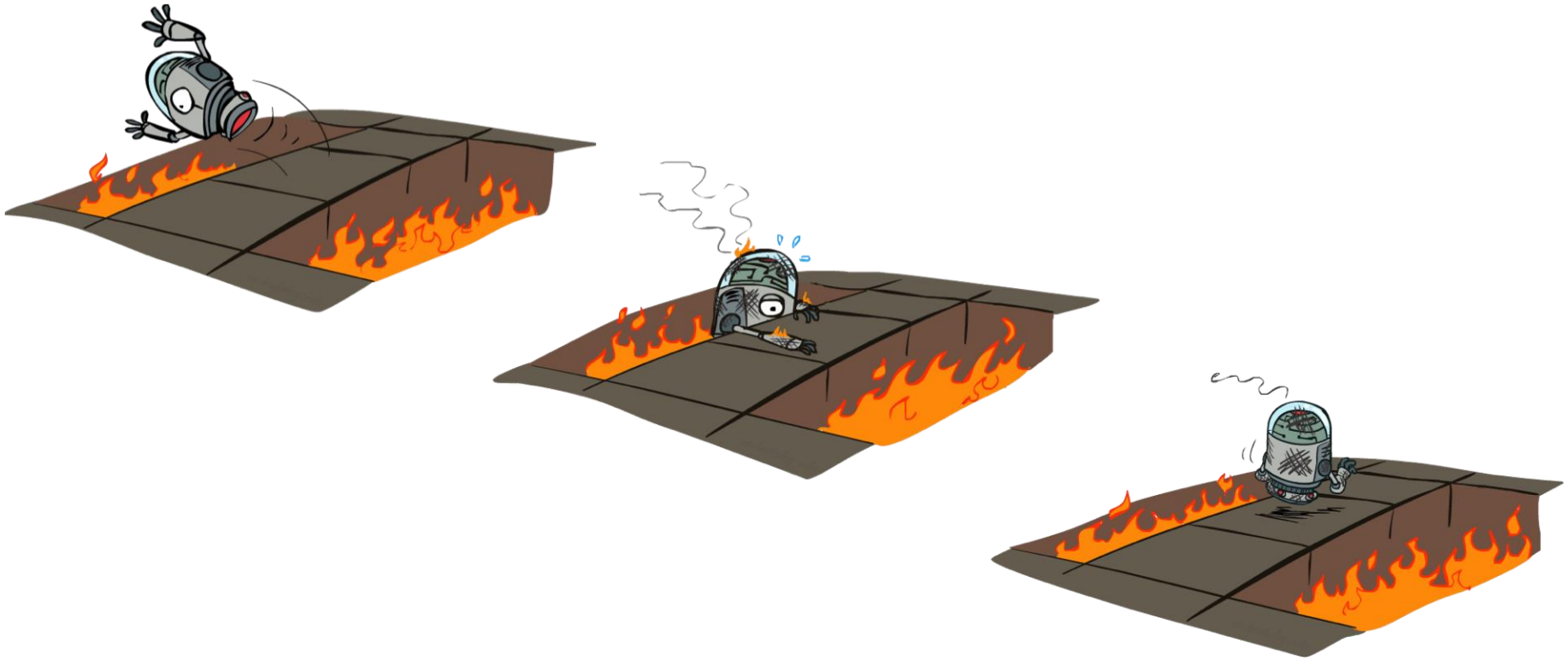
$$\pi(s) = \arg\max_a Q(s,a)$$

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

- Ötlet: Q-értékeket tanuljunk, ne hasznosságokat
- Ez a cselekvés kiválasztását is modellmentessé teszi

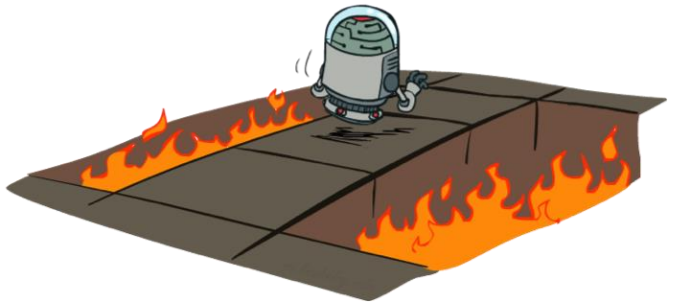


Aktív megerősítéses tanulás



Aktív megerősítéses tanulás

- Teljes megerősítéses tanulás: optimális eljárásmód meghatározása (pl. értékiteráció)
  - Nem ismerjük a T(s,a,s') átmeneteket.
  - Nem ismerjük az R(s,a,s') jutalmakat
  - Ki kell választani a cselekvéseket
  - Cél: optimális eljárásmód meghatározása



- Ekkor:
  - A tanuló döntéseket hoz!
  - Alapvető kompromisszum: felfedezés vs. kihasználás
  - Ez NEM offline tervezés!
  - Ténylegesen lépéseket teszel a világban, és megtudod, mi történik...

Kitérő: Q-érték iteráció

- Érték iteráció: egymást követő (mélységkorlátozott) értékek keresése

U=V

  - Kezdjük $V_0(s)$ = 0-val, amelyről tudjuk, hogy helyes.
  - Adott Vk, számítsuk ki a k+1 mélységű értékeket az összes állapotra:

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V_k(s') \right]$$

- De a Q-értékek hasznosabbak, ezért inkább ezeket számítsuk ki!
  - Kezdjük $Q_0(s,a)$ = 0-val, amiről tudjuk, hogy helyes.
  - Adott Qk, számítsuk ki a k+1 mélységű q-értékeket az összes q-állapotra:

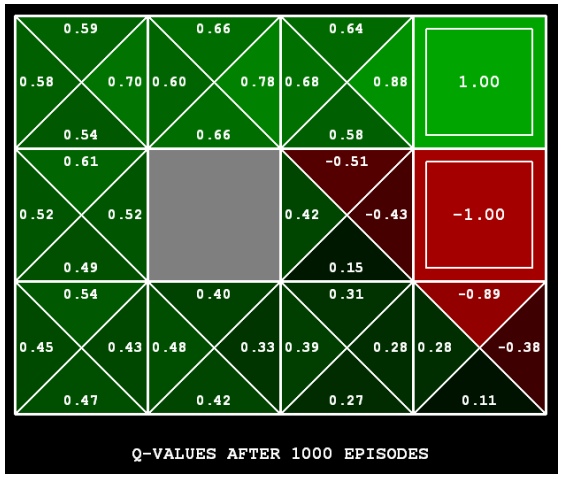
$$Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma \max_{a'} Q_k(s',a') \right]$$

Q-tanulás

- Q-tanulás: minta alapú Q-érték iteráció

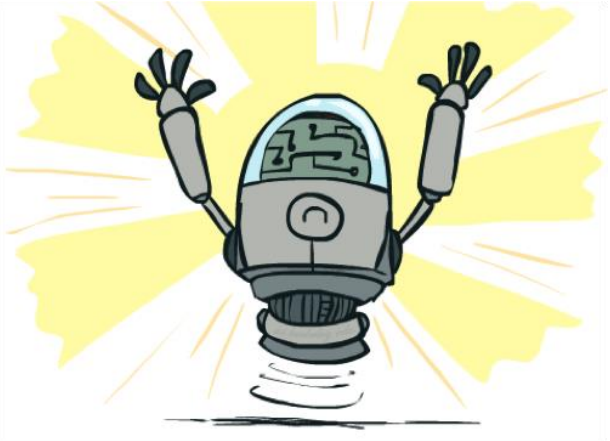
$$Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma \max_{a'} Q_k(s',a') \right]$$

- Q(s,a) értékeket kell tanulni menet közben
  - Beérkezi egy minta: (s,a,s',r)
  - Tekintésük a korábbi becslést: $Q(s,a)$
  - Tekintsük az új minta alapján kapott becslést: $sample = R(s,a,s') + \gamma \max_{a'} Q(s',a')$
  - Építsük be az új értéket egy mozgóátlagba $Q(s,a) \leftarrow (1-\alpha)Q(s,a) + (\alpha)\,[sample]$



Q-tanulás tulajdonságai

- Érdekes eredmény: A Q-tanulás konvergál az optimális eljárásmódhoz -- még akkor is, ha szuboptimálisan cselekszünk
- Ezt hívják off-policy / eljárásmódon kívüli tanulásnak



- Lehetséges problémák:
  - Az állapotokat fel kell fedezni elégséges mértékben
  - A tanulási rátának alacsonnyá kell válnia, de nem szabad túl gyorsan csökkenteni
  - Alapvetően, hosszú távon nem számít, hogyan választjuk ki a cselekvéseket (!)

Markov-döntési folyamat (MDF) vs RL

Ismert MDF: Offline megoldás

Cél	Módszer
U*, Q*, π* számítása	Érték- / eljárásmód- iteráció
Rögzített π eljárásmód kiértékelése	Eljárásmód kiértékelés

Ismeretlen MDF: Modellalapú

Cél	Módszer
U*, Q*, π* számítása	Érték / eljárásmód iteráció közelítő MDF -en
Rögzített π eljárásmód kiértékelése	Eljárásmód kiértékelés közelítő MDF-en

Ismeretlen MDF: Modellmentes

Cél	Módszer
U*, Q*, π* számítása	Q-tanulás
Rögzített π eljárásmód	Hasznosságok tanulása

Q-tanulás

- Szeretnénk minden Q-állapothoz Q-érték frissítést végezni:

$$Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma \max_{a'} Q_k(s',a') \right]$$

  - De ezt a frissítést nem lehet kiszámítani T, R ismerete nélkül.

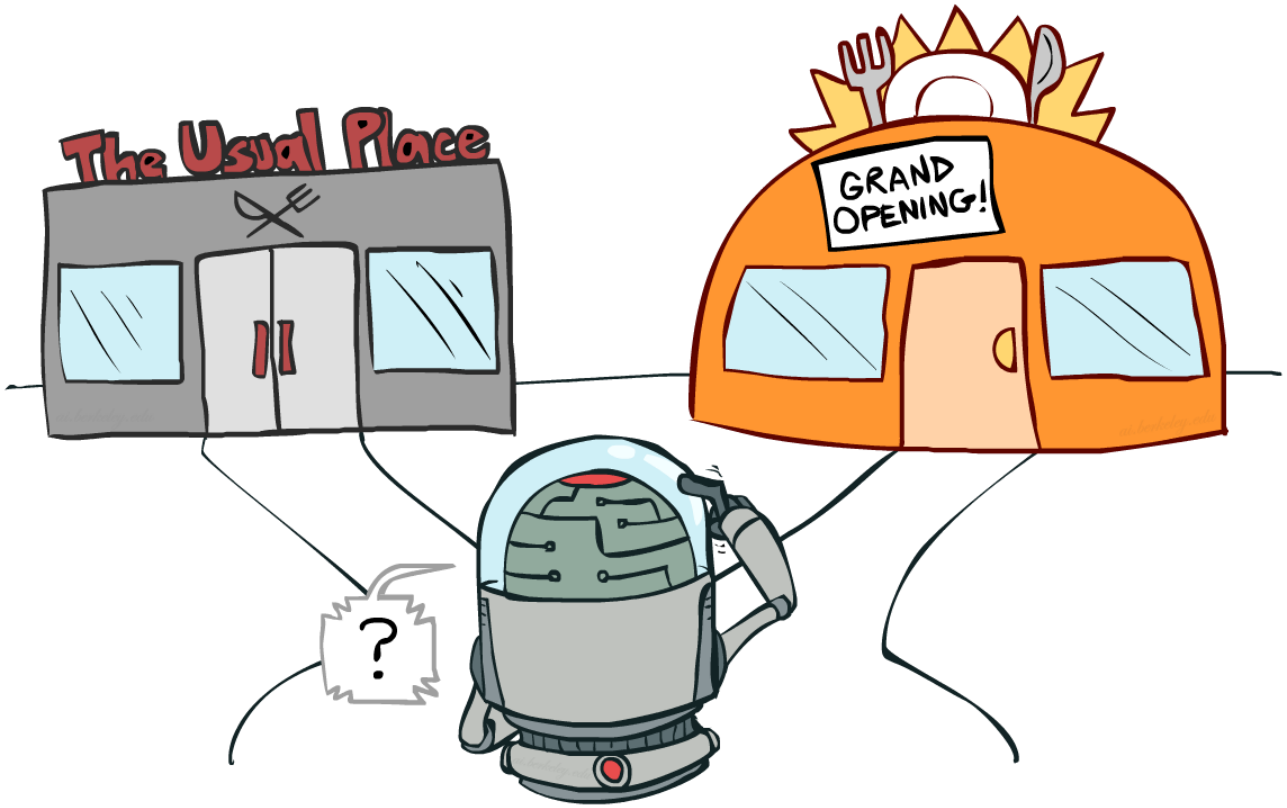
- Ehelyett átlagot számítunk menet közben
  - Megfigyelünk egy állapotátmenetet (s,a,r,s')
  - Ez alapján:

$$Q(s,a) \approx r + \gamma \max_{a'} Q(s',a')$$

  - De átlagolni szeretnénk további (s,a) –ból érkező minták szerint
  - Mozgóátlagot számítunk

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + (\alpha) \left[r + \gamma \max_{a'} Q(s',a') \right]$$

Felfedezés vs. Kizsákmányolás (Exploration vs. Exploitation)



Hogyan fedezzünk fel?

- Számos séma létezik a felfedezés kikényszerítéséhez kényszerítéshez
  - Legegyszerűbb: véletlenszerű akciók (ε-mohó)
  - Minden időlépésnél dobjunk fel egy érmét
    - (Kis) valószínűséggel ε, véletlenszerű cselekvés.
    - (Nagy) valószínűséggel 1-ε, az aktuális eljárásmód szerint cselekszünk.
  - Problémák a véletlenszerű cselekvésekkel?
    - Végül is felfedezi a teret, de a tanulás befejezése után is folytatni kell a „túrázást”.
    - Egy megoldás: idővel alacsonyabb ε
    - Másik megoldás: felfedezési függvények

Felfedezési függvények

- Mikor kell felfedezni?
  - Véletlenszerű akciók: fedezzen fel egy meghatározott mennyiségnyit
  - Jobb ötlet: fedezzen fel olyan területeket, amelyek rosszaságát még nem állapították meg, végül hagyja abba a felfedezést

- Felfedezési függvények
  - Vesz egy becsült u értéket és egy n látogatási számot, és „optimista” hasznosságot ad vissza, pl. $f(u, n) = u + k/n$

Normális Q-frissítés: $Q(s,a) \leftarrow_\alpha R(s,a,s') + \gamma \max_{a'} Q(s',a')$

Módosított Q-frissítés: $Q(s,a) \leftarrow_\alpha R(s,a,s') + \gamma \max_{a'} f(Q(s',a'), N(s',a'))$

  - Megjegyzés: ez a "bónuszt" visszaterjeszti az ismeretlen állapotokhoz vezető állapotokra is!

Az állapottér felderítése

Kompromisszum: a **jelenlegi jutalom**, amit a pillanatnyi hasznosság-becslés tükröz, és a **hosszú távú előnyök** közt.

Két megközelítés a cselekvés kiválasztásában:

„**Hóbortos, Felfedező**„: véletlen módon cselekszik, annak reményében, hogy végül is felfedezi az egész környezetet

„**Mohó**„: a jelenlegi becslésre alapozva maximalizálja a hasznot.

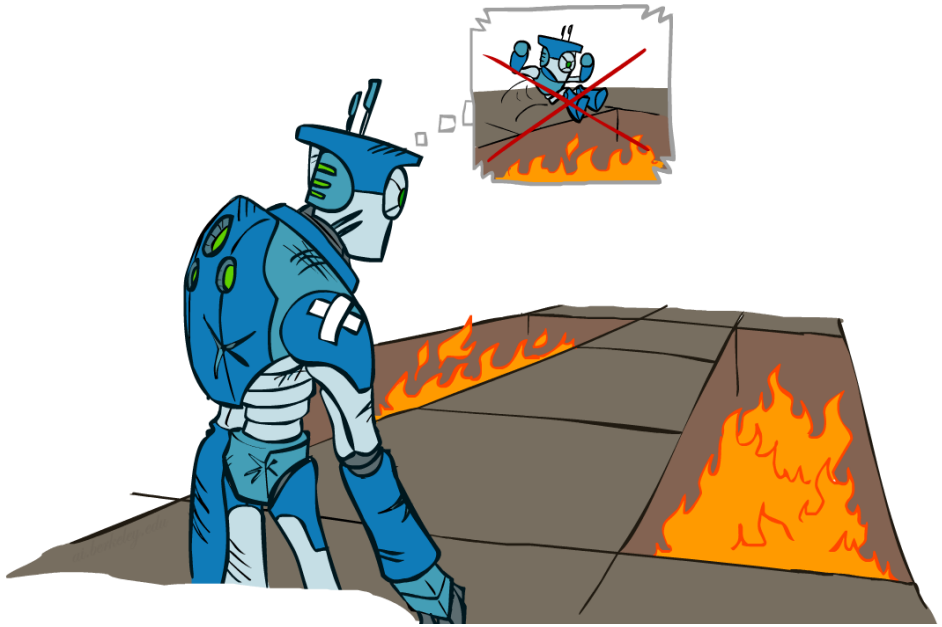
Hóbortos: képes jó hasznosság-becsléseket megtanulni az összes állapotra. Sohasem sikerül fejlődnie az optimális jutalom elérésében.

Mohó: gyakran talál egy jó utat. Utána ragaszkodik hozzá, és soha nem tanulja meg a többi állapot hasznosságát.

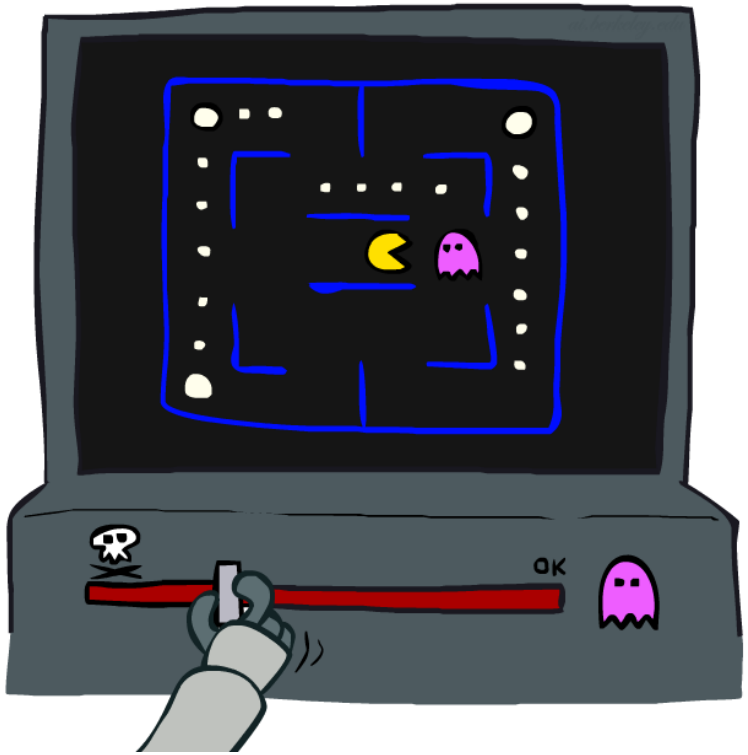
Ágens addig legyen hóbortos, amíg kevés fogalma van a környezetről, és legyen mohó, amikor a valósághoz közeli modellel rendelkezik.

Megbánás/sajnálat (Regret)

- Még ha meg is tanulja az optimális eljárásmódot, akkor is követ el hibákat az út során!
- A megbánás a teljes hibaköltséget méri: a különbséget a (várható) jutalmak (beleértve a korai szuboptimalitást is) és az optimális (várható) jutalmak között.
- A megbánás minimalizálása túlmutat azon, hogy megtanuljuk, hogy optimálisak legyünk - ehhez optimálisan kell megtanulnunk, hogy optimálisak legyünk.
- Példa: a véletlenszerű felfedezés és a felfedező függvények mindkettő optimálisnak bizonyul, de a véletlenszerű felfedezés nagyobb megbánással jár.

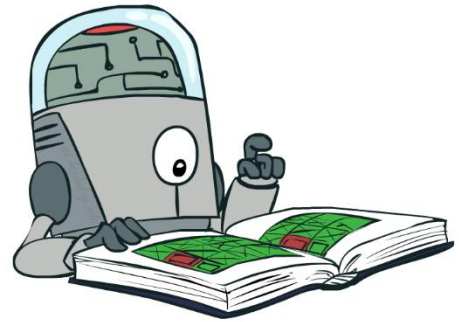


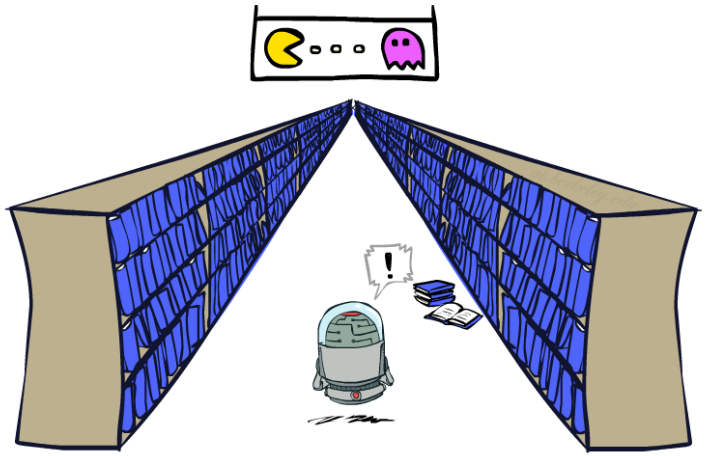
Közelítő Q-Learning



Általánosítás állapotok között

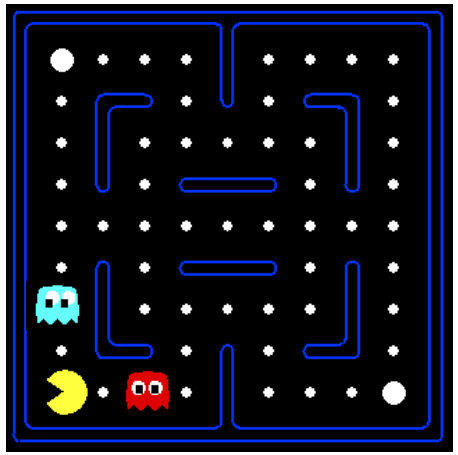
- Az alap Q-tanulás egy táblázatot vezet az összes q-értékről.
  - Reális helyzetekben nem tudunk minden egyes állapotot megismerni!
  - Túl sok az állapot ahhoz, hogy mindet meglátogassuk a tanítás során
  - Túl sok az állapot ahhoz, hogy a q-táblákat a memóriában tartsuk.
- Ehelyett általánosítani szeretnénk:
  - Néhány kis számú gyakorlóállapotot tanulni a tapasztalatból.
  - Általánosítani ezt a tapasztalatot új, hasonló helyzetekre
  - Ez egy alapvető gondolat a gépi tanulásban, és újra és újra találkozni fogunk vele.



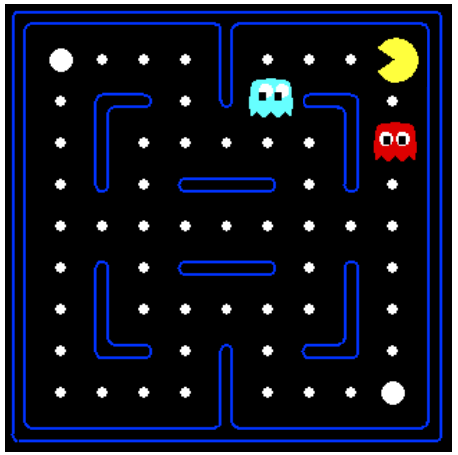


Példa: Pacman

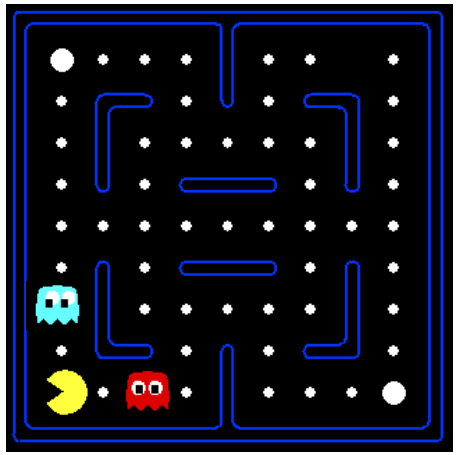
Tegyük fel, hogy tapasztalat útján rájövünk, hogy ez az állapot rossz:



Az alap q-tanulásnál semmit sem tudunk erről az állapotról:

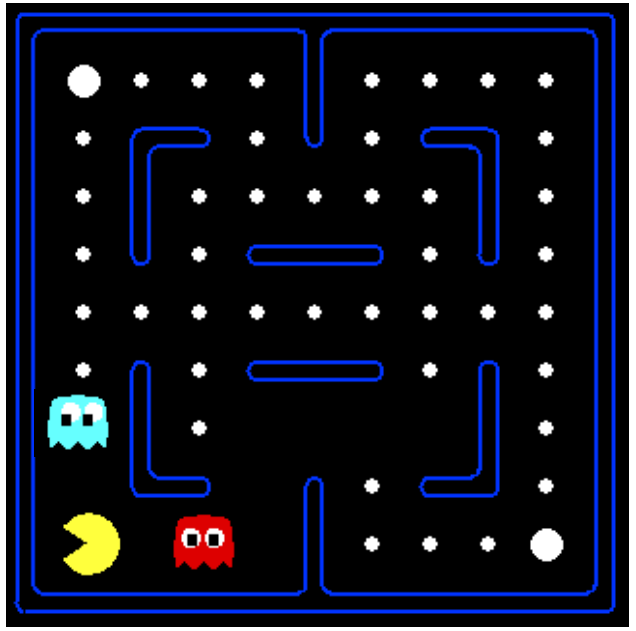


Vagy erről:



Jellemző alapú reprezentációk

- Megoldás: egy állapot leírása a jellemzők (tulajdonságok) vektorával



  - A jellemzők olyan függvények, amelyek az állapotoktól valós számokra (gyakran 0/1) képeznek le, és az állapot fontos tulajdonságait rögzítik.
  - Példa jellemzőkre:
  - Távolság a legközelebbi szellemtől
  - Távolság a legközelebbi ponthoz
  - Szellemek száma
  - 1 / (távolság a pontig)^2
  - Pacman egy alagútban van? (0/1)..... stb.
  - Egy q-állapot (s, a) is leírható jellemzőkel (pl. az akció közelebb visz az ételhez).

Lineáris érték/hasznosságfüggvények

- A tulajdonságreprezentáció segítségével néhány súly segítségével q-függvényt (vagy értékfüggvényt) írhatunk bármely állapotra:

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

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

- Előnye: tapasztalataink néhány értékben összegződnek

- Hátrány: az állapotoknak lehetnek közös jellemzőik, de valójában nagyon különböző hasznosságúak lehetnek!

Közelítő Q-tanulás

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

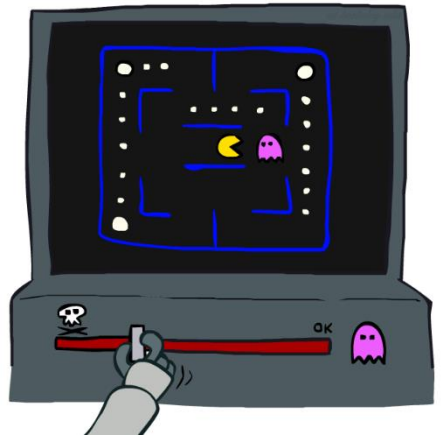
- Q-tanulás lineáris Q-függvényekkel:

$$\text{transition} = (s, a, r, s')$$

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

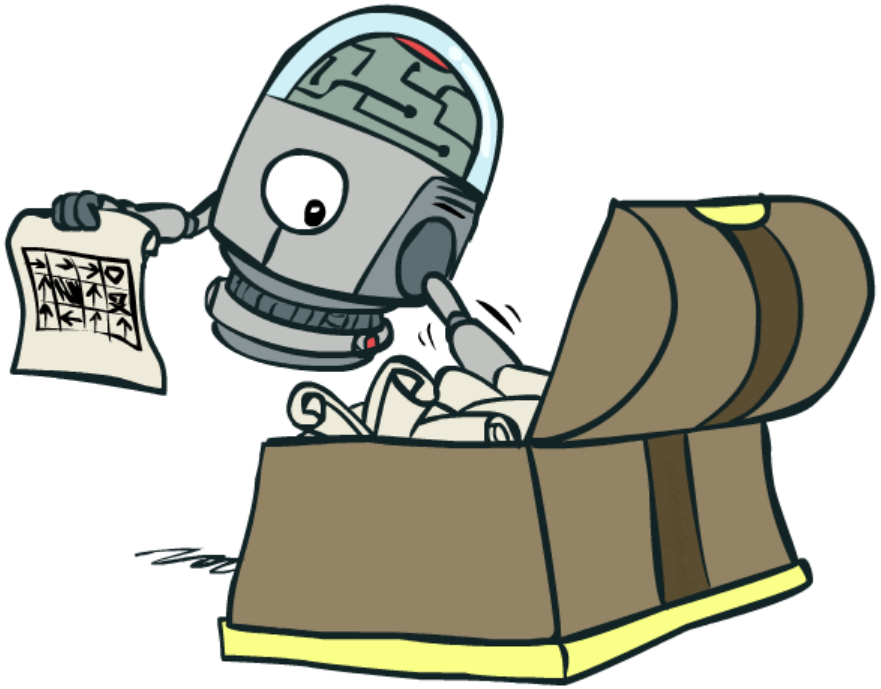
$$Q(s,a) \leftarrow Q(s,a) + \alpha\,[\text{difference}] \qquad \text{Egzakt Q}$$

$$w_i \leftarrow w_i + \alpha\,[\text{difference}]\, f_i(s,a) \qquad \text{Közelítő Q}$$



- Intuitív értelmezés:
  - Az aktív jellemzők súlyainak beállítása
  - Pl. ha valami váratlanul rossz történik, hibáztassuk az aktív jellemzőket: diszpreferáljuk az összes olyan állapotot, amely az adott állapot jellemzőivel rendelkezik

Eljárásmód-keresés



Eljárásmód-keresés

- Probléma: gyakran nem azok a jellemzőalapú eljárásmódok működnek jól (játékokat nyernek, maximalizálják a hasznosságot), amelyek a legjobban közelítik a U / Q értéket.
  - A Q-tanulás prioritása: a Q-értékek közelítése (modellezés)
  - A cselekvés kiválasztásának prioritása: a Q-értékek helyes sorrendbe állítása (előrejelzés).
  - Még találkozunk a modellezés és az előrejelzés közötti különbségtétellel.

- Megoldás: olyan eljárásmódot tanuljunk, amely maximalizálja a jutalmakat, nem pedig az azokat előrejelző értékeket.

- Eljárámód-keresés: kezdjük egy megfelelő megoldással (pl. Q-tanulás), majd finomhangolást végezzünk a jellemző súlyok hangolásával hegymászás al.

Eljárásmód-keresés

- A legegyszerűbb eljárásmód-keresés:
  - Kezdjük egy kezdeti lineáris értékfüggvénnyel vagy Q-függvénnyel.
  - Az egyes jegyek (jellemzők) súlyát felfelé és lefelé módosítjuk, és megnézzük, hogy az eljárásmód jobb-e, mint korábban.

- Problémák:
  - Honnan tudjuk, hogy az eljárásmód jobb lett?
  - Sok mintaepizódot kell lefuttatni!
  - Ha sok jellemző van, ez nem lehet praktikus

- Jobb módszerek kihasználják a lookahead struktúrát, bölcs mintavételezés, több paraméter megváltoztatása...

RL alkalmazások - Ipari automatizálás

- Hűtés/energiafelhasználás optimalizálása
  - Ötpercenként pillanatfelvételek készítése az adatközpontok adataiból, és ezek betáplálása a mély neurális hálózatokba.
  - Ezután megjósolja, hogy a különböző kombinációk hogyan befolyásolják a jövőbeli energiafogyasztást.
  - Azonosítja azokat az intézkedéseket, amelyek minimális energiafogyasztáshoz vezetnek, miközben fenntartják a biztonsági kritériumok meghatározott szabványát.
  - Elküldi és végrehajtja ezeket az intézkedéseket az adatközpontban

https://deepmind.com/blog/article/safety-first-ai-autonomous-data-centre-cooling-and-industrial-control

RL a robotmozgás tervezésében és kivitelezésében

- A mélytanulás és a megerősítő tanulás alkalmazásával olyan robotok képezhetők ki, amelyek képesek különböző tárgyakat megragadni - még azokat is, amelyeket a képzés során nem láttak.
- Ez például felhasználható a termékek összeszerelősoron történő építésénél.

https://www.youtube.com/watch?v=W4joe3zzgIU

RL Kereskedelmi és pénzügyi alkalmazások

- A felügyelt idősororos modellek felhasználhatók a jövőbeli eladások előrejelzésére, valamint a részvényárfolyamok előrejelzésére.
- Ezek a modellek azonban nem határozzák meg, hogy egy adott részvényárfolyamnál milyen lépéseket kell tenni.
- Egy RL-ügynök képes dönteni egy ilyen feladatról; arról, hogy egy részvényt megtartsunk, vásároljunk vagy eladjunk.
- Az RL-modellt piaci benchmark-szabványok alapján értékelik, hogy biztosítsák az optimális teljesítményt.

https://medium.com/ibm-data-ai/reinforcement-learning-the-business-use-case-part-2-c175740999