Osnovi Računarske Inteligencije

Učenje Uslovljavanjem

Reinforcement Learning

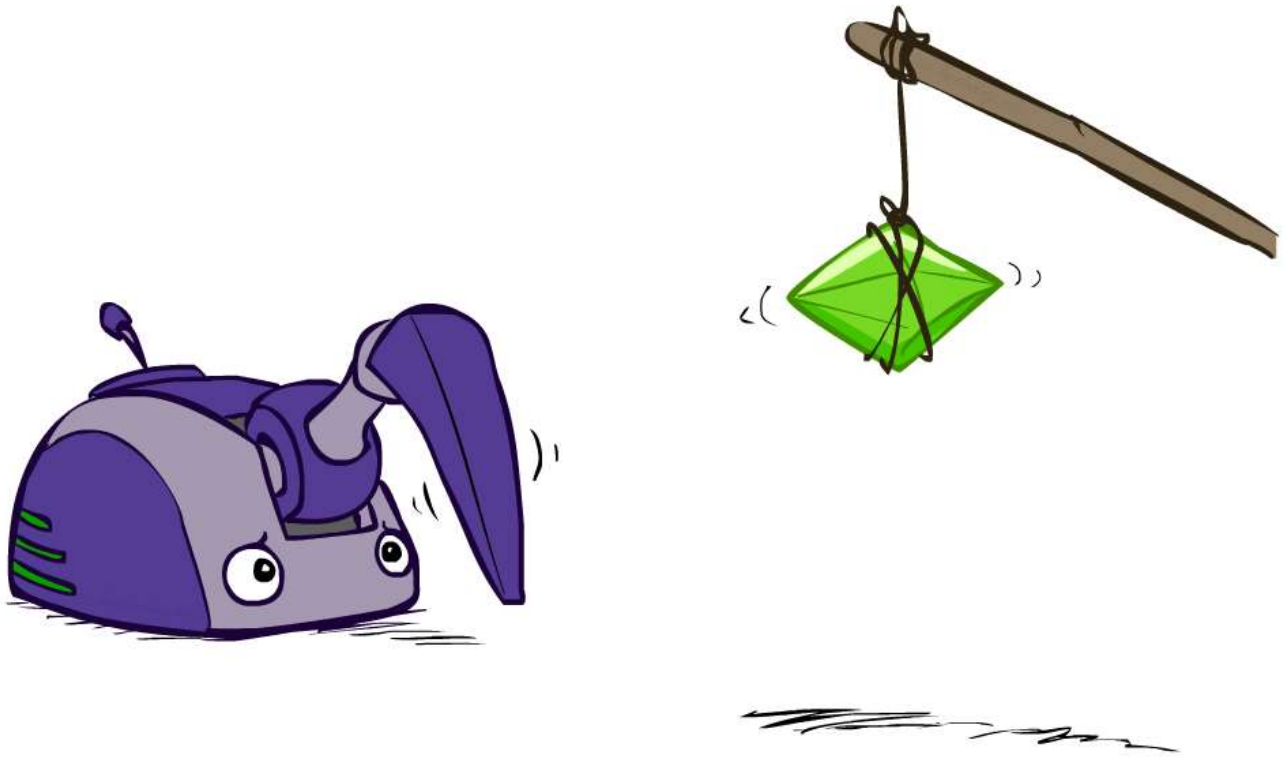
Predavač: Aleksandar Kovačević

Slajdovi preuzeti sa kursa CS188, University of California, Berkeley

http://ai.berkeley.edu/

Učenje Uslovljavanjem - Reinforcement Learning

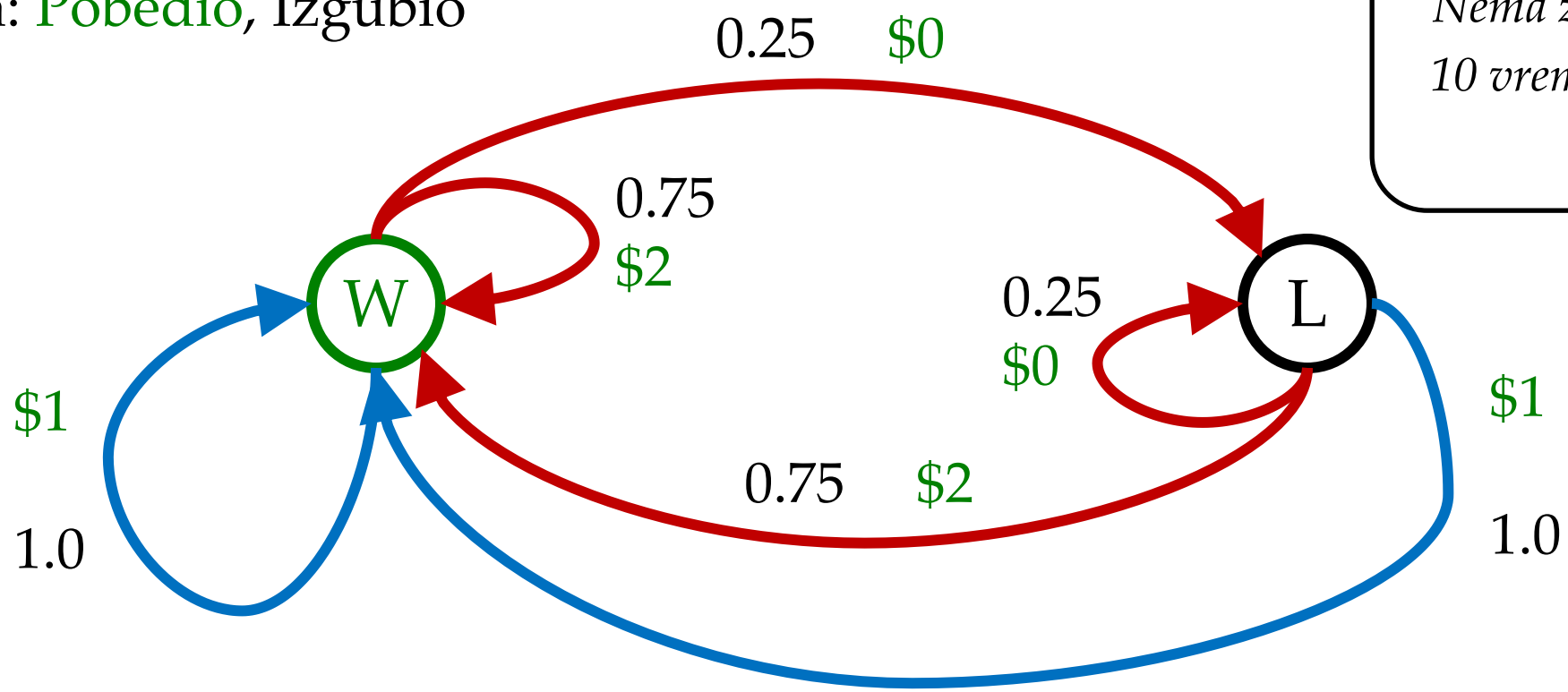
- Napomena oko terminologije:
- Ima puno prevoda za RL, Pojačano Učenje, Učenje Sa Podrškom itd.
- Tokom predavanja i kursa koristićemo termin Učenje Uslovljavanjem kao i orginalni engleski termin.



Problem: Double Bandits

Double-Bandit MDP

- Akcije: *Plavi*, *Crveni*
- Stanja: Pobedio, Izgubio



Nema zanemarivanja
10 vremenskih koraka

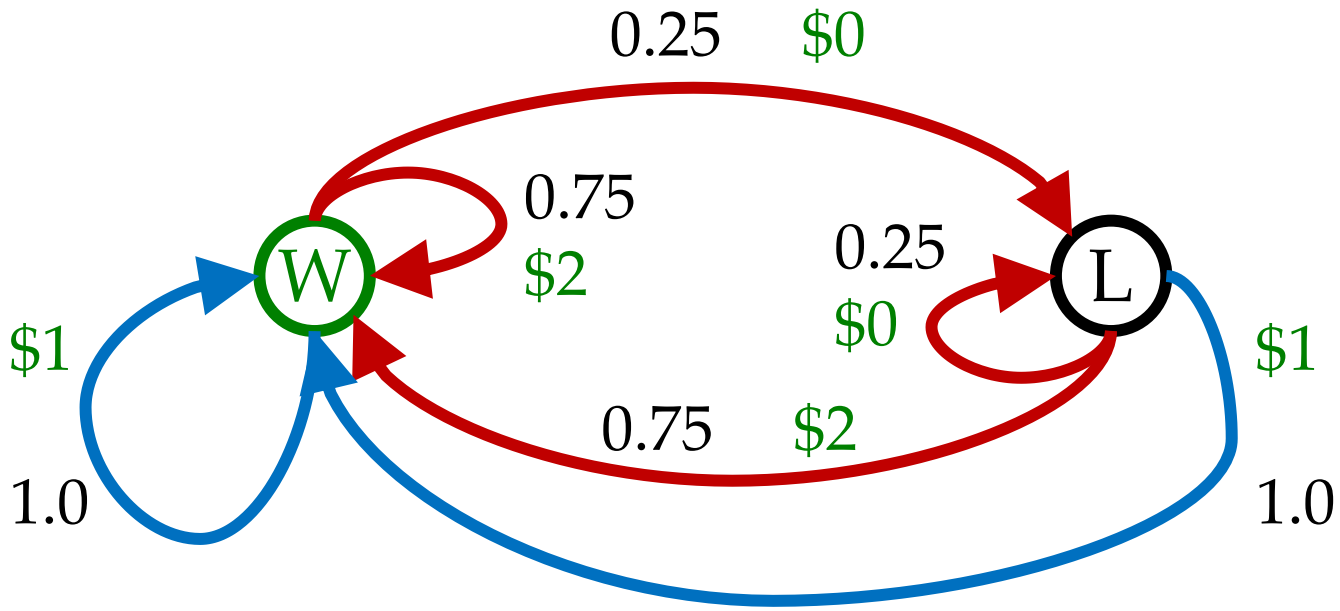
Planiranje Offline - Offline Planning

- Rešvavanje MDP je offline planiranje
  - Q i V vrednosti izračunavamo offline, agent ne vrši ackije u okruženju. Tek kada dobijemo politiku, pokrećemo agenta.
  - Moramo da znamo sve detalje (f- je prelaza i nagrade) MDP da bi mogli da izračunamo Q i V.

Nema zanemarivanja 10 vremenskih koraka

Rezultat planiranja offline tj. rešavanja MDP:

	Vrednost
Play Red	15
Play Blue	10



Hajde da povučemo ručke!

- Rešavanjem MDP smo videli da nam se više isplati da povlačimo ručku na crvenom automatu.
- Recimo da sada stvarno to i uradimo.
- Dobili smo sledeće vrednosti.
- Ukupno 12 što je bilizu naše procene da prosečno dobijamo 15 ako koristimo crveni automat.

$2 $2 $0 $2 $2

$2 $2 $0 $0 $0

Hajde da povučemo ručke!

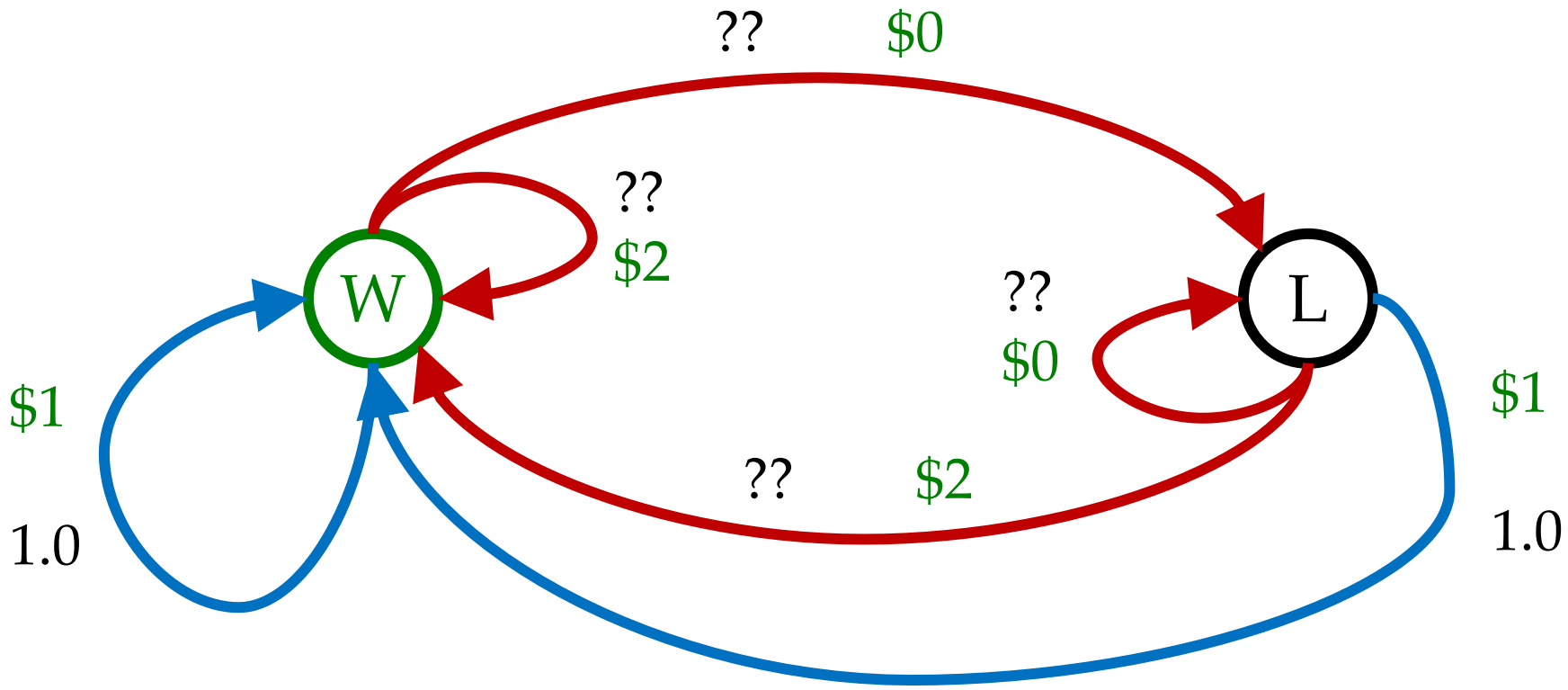
- Šta bi bilo kada ne bi imali sve verovatnoće za ovaj problem?
- Ne bi mogli da rešimo MPD offline, nedostaju nam podaci.
- Šta možemo da uradimo?
- Možemo da krenemo da stvarno povlačimo ručke (bez planiranja) i koristimo nagrade (kazne) koje dobijemo.
- To je poneta oblasti koju sada radimo.

$2 $2 $0 $2 $2

$2 $2 $0 $0 $0

Online Planiranje - Online Planning

- Kao što smo već pomenuli, pretpostavimo da nam nedostaju svi ili neki podaci o modelu sveta:



Hajde ponovo da povučemo!

- Nemamo podatke o crvenom automatu, pa ponovo 10 puta povlačimo ručku na njemu da vidimo šta ćemo dobiti.
- Ovaj put su vrednosti drugačije nego prethodni.
- Kako ćemo da iskoristimo naša iskustva sa okolinom da naučimo kako da maksimizujemo nagradu koju možemo da dobijemo?

$0 $0 $0 $2 $0

$2 $0 $0 $0 $0

Šta se sada dogodilo?

- Sada se više ne bavimo planiranjem, već učenjem!
  - Konkretnije, učenjem uslovljavanjem (*reinforcement learning*, RL)
  - Model nam je MDP, ali nismo mogli da ga rešimo offline (ne znamo unapred f-je prelaza ni nagrade).
  - Moramo da preduzimamo akcije da bi vremenom naučili podatke koji nam trebaju da rešimo MDP.

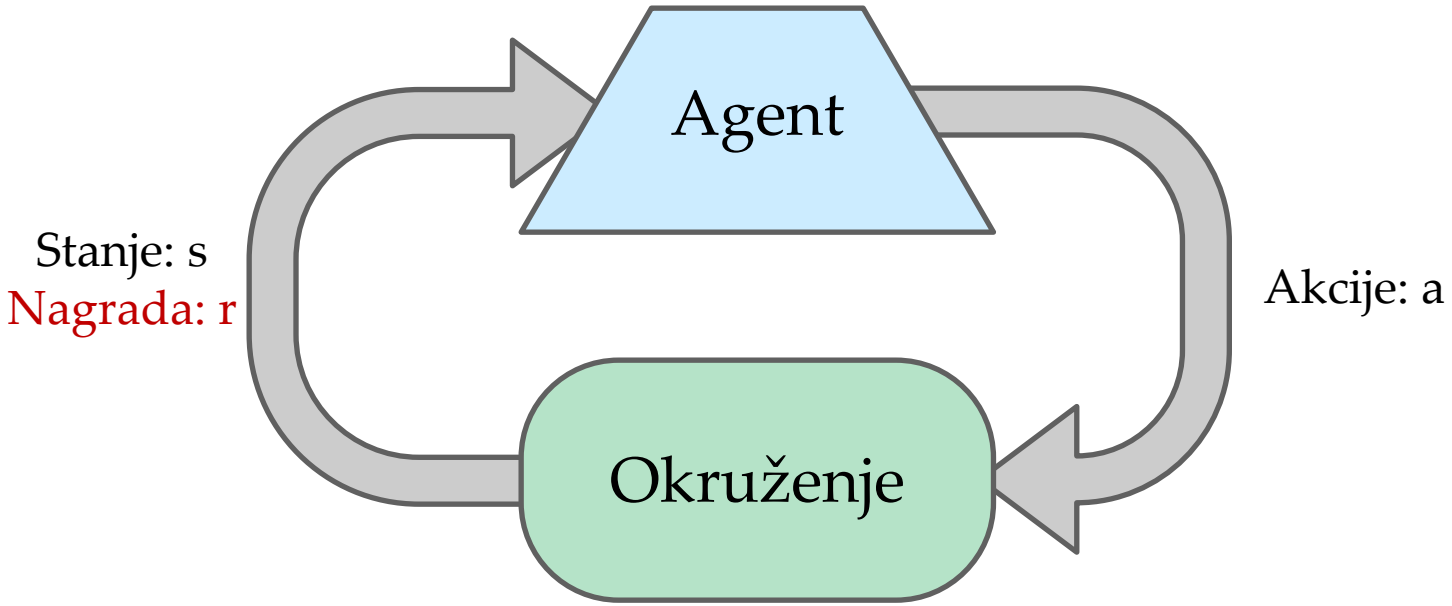
- Važni koncepti vezani za RL
  - Istraživanje: moramo da probamo nove akcije/stanja da bi dobili informacije
  - Eksploatacija: vremenom, kada naučimo, radimo akcije koje nam donose najveću očekivanu nagradu
  - Žaljenje (*Regret*): čak iako učimo inteligentno, pravićemo greške
  - Semplovanje: zbog stohastničosti sveta, moramo da ponavljamo iste akcije
  - Težina: učenje je često mnogo teže nego rešavanje datog MDP sa svim podacima

Reinforcement Learning

- I dalje računamo da imamo Markovljev Proces Odlučivanja (MDP):
  - Skup stanja $s \in S$
  - Skup akcija A
  - Funkciju prelaza $T(s,a,s')$
  - Funkciju nagrade $R(s,a,s')$
- I dalje tražimo politiku $\pi(s)$

- Međutim, sada: ne znamo T i R
  - Ne znamo koja stanja su dobra niti koji su upošte rezultati naših akcija
  - Moramo da probamo akcije i odemo u stanja da bi učili

Reinforcement Learning



- Generalna ideja:
  - Dobijamo povratnu informaciju (*feedback*) u vidu nagrada
  - Korisnost agenta definišemo pomoću funkcije nagrade
  - Moramo da naučimo kako da se ponašamo da bi maksimizovali očekivanu nagradu
  - Svo naše učenje zasnovano je na semplovima povratnih informacija koje dobijamo kao rezultate naših akcija u okruženju!

Primer: Robot uči da hoda

Initial

A Learning Trial

After Learning [1K Trials]

[Kohl and Stone, ICRA 2004]

Primer: Robot uči da hoda

Initial

[Kohl and Stone, ICRA 2004]

[Video: AIBO WALK – initial]

Primer: Robot uči da hoda

Training

[Kohl and Stone, ICRA 2004]

[Video: AIBO WALK – training]

Primer: Robot uči da hoda

Finished

[Kohl and Stone, ICRA 2004]

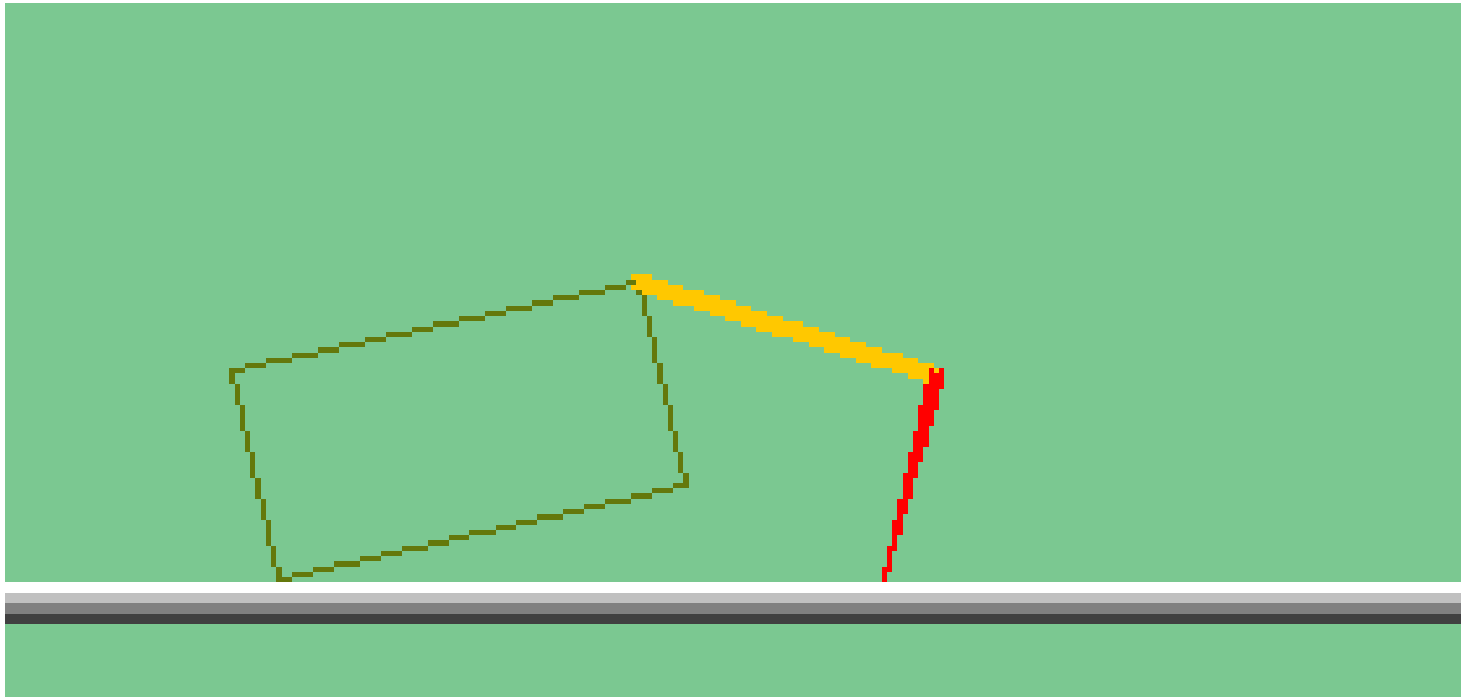
[Video: AIBO WALK – finished]

Primer: Robot uči da hoda

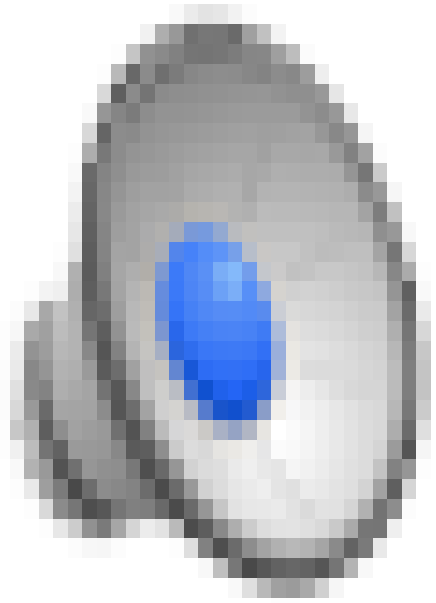
[Tedrake, Zhang and Seung, 2005]

[Video: TODDLER – 40s]

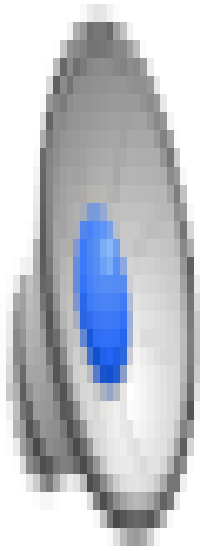
Kako naučiti ovog robota da puzi!



Demo robota koji puzi



DeepMind Atari (©Two Minute Lectures)

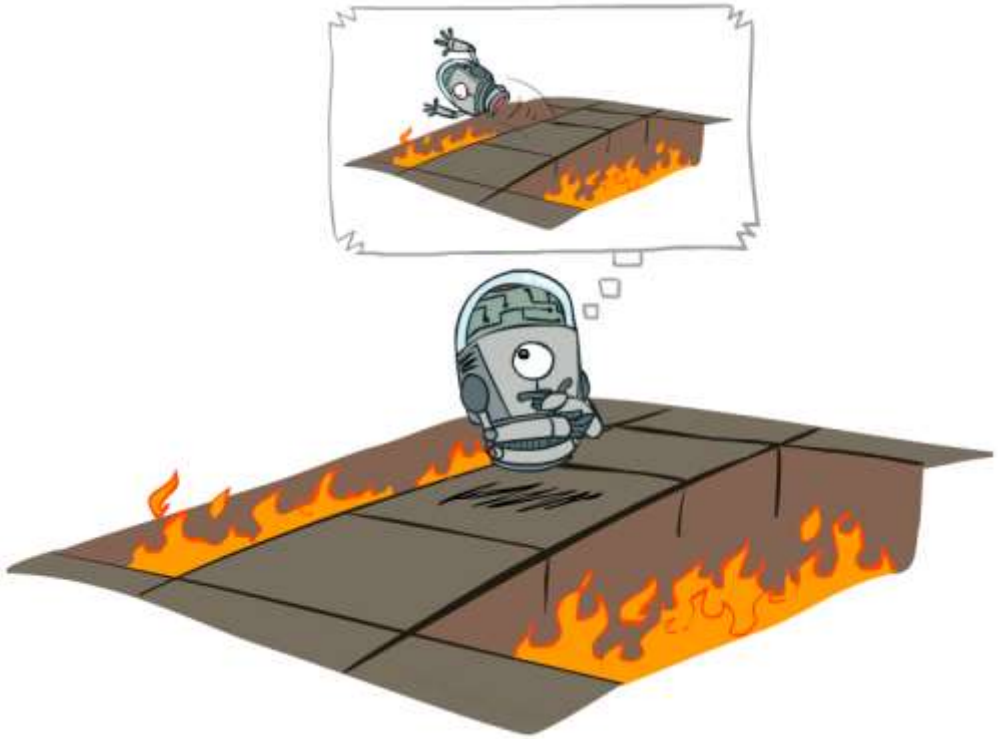


Reinforcement Learning

- I dalje računamo da imamo Markovljev Proces Odlučivanja (MDP):
  - Skup stanja $s \in S$
  - Skup akcija A
  - Funkciju prelaza $T(s,a,s')$
  - Funkciju nagrade $R(s,a,s')$
- I dalje tražimo politiku $\pi(s)$

- Međutim, sada: ne znamo T i R
  - Ne znamo koja stanja su dobra niti koji su upošte rezultati naših akcija
  - Moramo da probamo akcije i odemo u stanja da bi učili

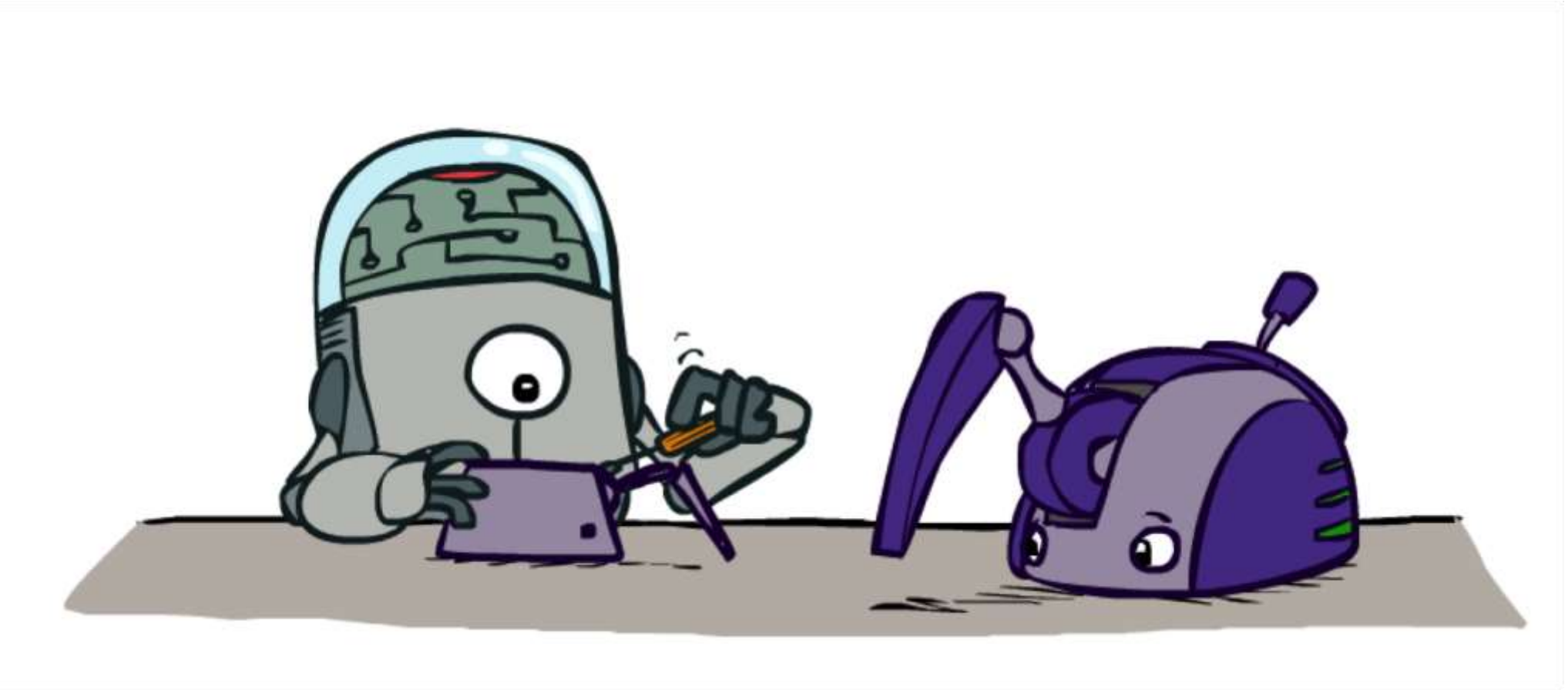
Offline (MDPs) vs. Online (RL)



Offline Rešenje

Online Učenje

Učenje Zasnovano Na Modelu
Model-Based Learning



Učenje Zasnovano Na Modelu

- Ideja:
  - Naučiti aproksimaciju modela na osnovu iskustava iz okruženja
  - Konkretno za MPD pod modelom misli se na funkcije T i R
  - Kada naučimo aproksimaciju rešimo model metodama koje smo ranije radili

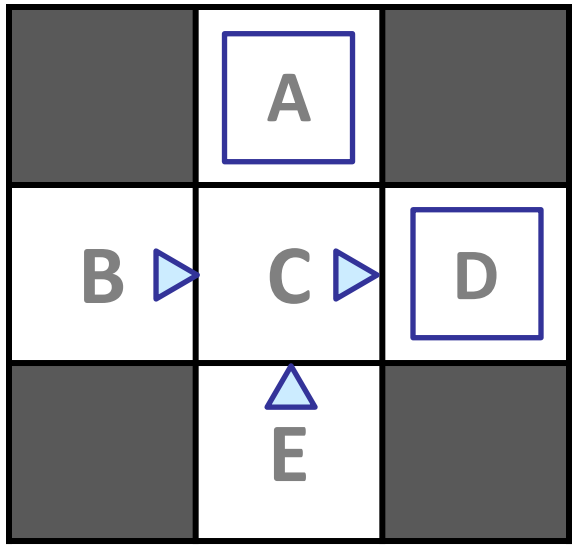
- Korak 1: Emprijiski učimo MDP model
  - Brojimo ishode akcija *a* u stanjima *s*
  - Normalizujemo frekvencije da bi dobili $\hat{T}(s, a, s')$
  - Otkrivamo $\hat{R}(s, a, s')$ kad god iskusimo (s, a, s′)

- Korak 2: Rešavmo MDP koji smo naučili u Koraku 1
  - Npr. koristimo iteriranje vrednosti

Primer: Učenje Zasnovano Na Modelu

Ulaz: politika π

kakva god, još uvek ne zamo ništa



Koristimo: $\gamma = 1$

Iskustvo iz okruženja (Učenje)

Epizoda počinje odakle god i zaršava kad izađemo

Epizoda 1

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

Epizoda 2

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

Epizoda 3

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

Epizoda 4

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

Naučeni model

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

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
…

Šta je problematično kod učenja zasnovanog na modelu?

- Potrebno nam je jako mnogo podataka iz okruženja (epizoda) da bi dobro procenili verovatnoće za funkciju prelaza.
- Za okružanja sa malim brojem stanja i akcija prikupaljenje podataka nije problem.
- Međutim, kod kompleksnih okruženja proces prikupljanja ishoda svih mogućih parova (stanje, akcija) nije efikasan proces.
  - Mnogo je prirodnije da agent radi niz akcija u epziodama nego da ga primoravamo da ponavlja određene akcije u određenim stanjima jer nemamo podataka za njih, naročito kada ne radimo sa simulatorom nego u realnom svetu (npr. robot).
- Na koji način možemo da rešimo ovaj problem?

Šta je problematično kod učenja zasnovanog na modelu?

- Na koji način možemo da rešimo ovaj problem?
- Vratićemo se našem primarnom cilju, a to je procena očekivane nagrade za stanje ili par (stanje, akcija).
- Da li je učenje modela jedini način da procenimo očekivane nagrade?
- Nije! Postoje dva načina za određivanje očekivane vrednosti i jedan od njih ne zahteva model.
- Oba načina ilustrujemo na primeru na sledećem slajdu.

Očekivane vrednosti: *Model-Based* vs. *Model-Free*

Cilj: Izračunati očekivano godište studenata na III godini FTN.

Moramo da znamo P(A) - to je naš model

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

Bez P(A), prikupljamo semplove [a_1, a_2, … a_N]

Nepoznato P(A): "Model-Based"

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

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

Zašto ovo radi? Uz dovoljno semplova naučićemo model.

Nepoznatno P(A): "Bez Modela"

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

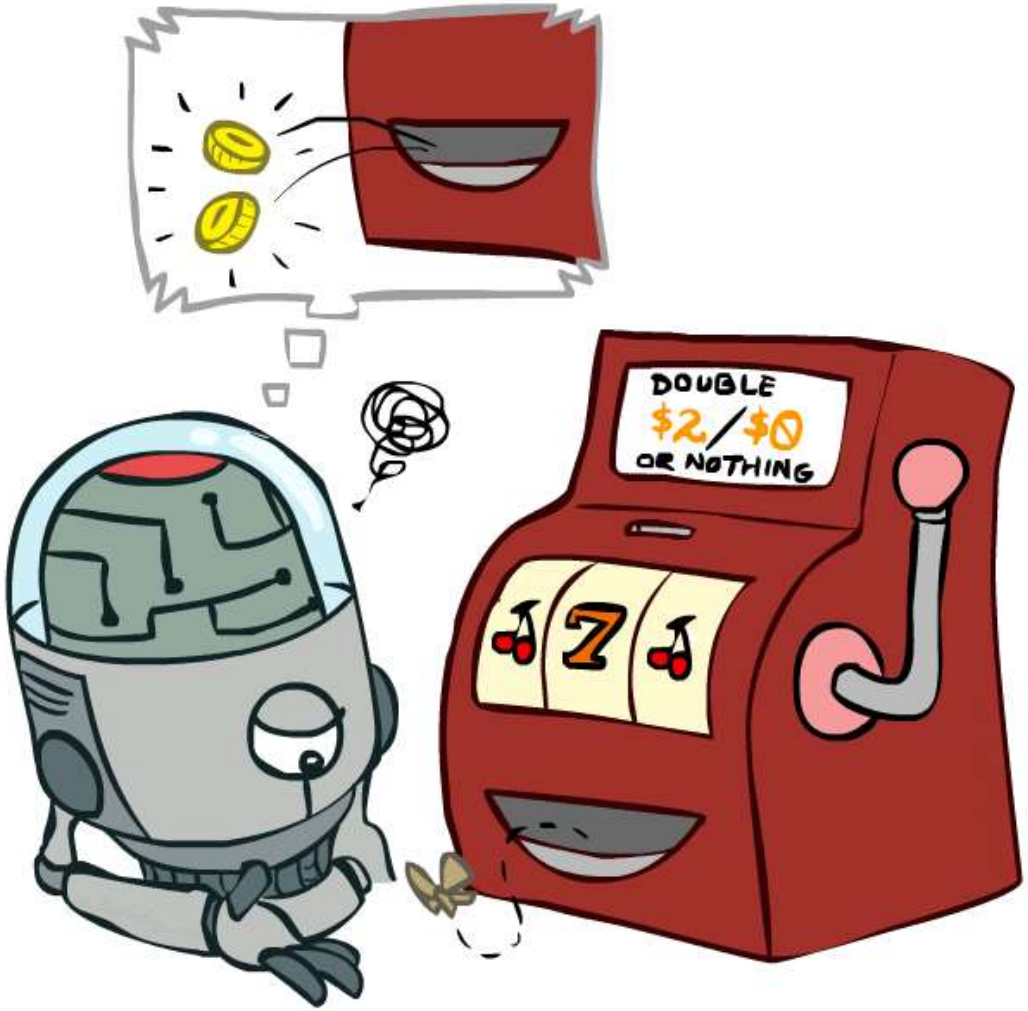
Koristimo prosek serije semplova. Zašto ovo radi? Prosek sam po sebi implicitno sadrži verovatnoće iz semplova.

Ako ubacimo P(a) u formulu levo dobijamo prosek. Verovatnoća da će nam se 21 pojaviti u semplu je jednaka verovatnoći da neko ima 21 godinu tj. P(21)

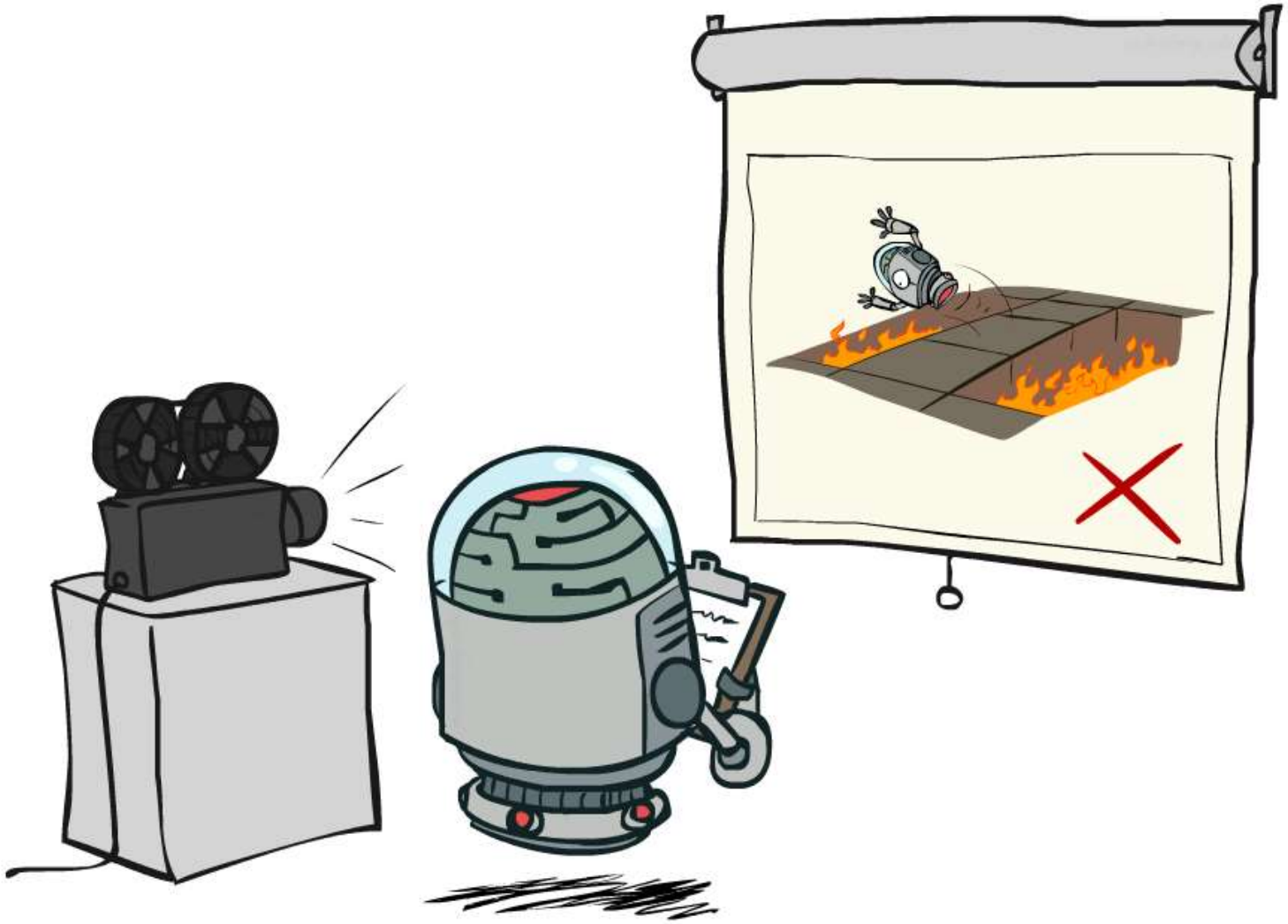
Šta možemo da zaključimo sa prethodnog slajda?

- Očekivanu vrednost nagrade možemo da procenimo bez modela!
- Sve što nam je potrebno su semplovi (epizode) stanja i akcija iz okruženja i nagrade koje tada dobijamo.
- Ovo je vrlo važan zaključak koji se dosta koristi u oblasti učenja uslovljavanjem.

Učenje Bez Modela
Model-Free Learning

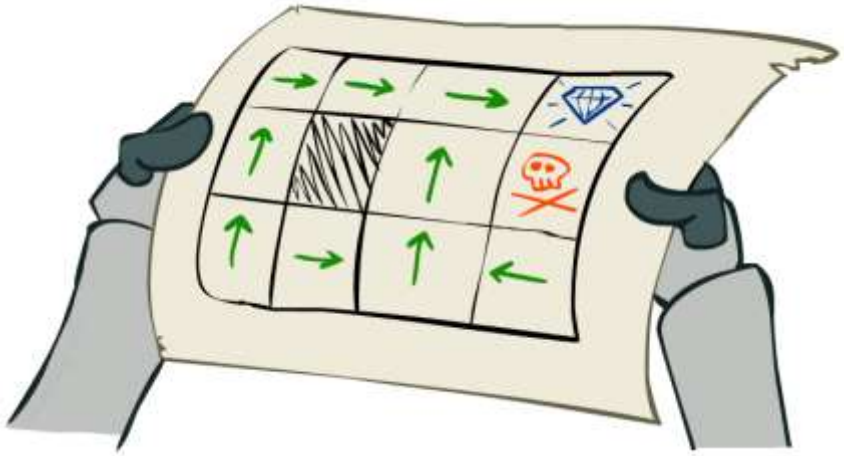


Pasivno Učenje Uslovljavanjem
Passive Reinforcement Learning



Pasivno učenje uslovljavanjem

- Kod pasivnog učenja imamo fiksnu politiku i cilj nam je da odredimo vrednosti (očekivane nagrade) za stanja ili parove stanja i akcija.
  - Dok određujemo vrednosti politika se ne menja!
- Pojednostavljen zadatak: evaluacija politike
  - Ulaz: fiksna politika $\pi(s)$
  - Ne znamo f-ju prelaza T(s,a,s')
  - Ne znamo nagrade R(s,a,s')
  - Cilj: naučiti vrednosti za stanja



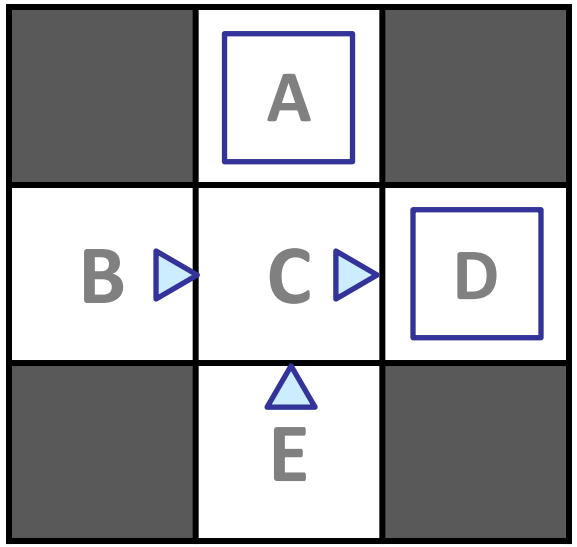
- U ovom slučaju:
  - Agent samo sluša fiksnu politiku (*"along for the ride"*)
  - Nema izbora o tome koje akcije da radi
  - Samo radi po datoj politici i uči iz iskustva
  - Ovo nije offline planiranje! Agent stvarno radi akcije u okruženju.

Direktna Evaluacija

- Jedan od pasivnih metoda
- Cilj: Izračunati vrednosti za sva stanja za π
- Ideja: Uzimamo proseke vrednosti semplova koje smo dobili iskustvom (tj. ponašanjem u okruženju)
  - Ponašamo se kako nam kaže π
  - Svaki put kada posetimo neko stanje zapišemo koja je tada bila suma očekivanih nagrada (uz zanemarivanje)
  - Onda za svako stanje uzmemo prosek toga što smo zapisali

Primer: Direktna Evaluacija

Fiksirana politika π



Koristimo: $\gamma = 1$

Epizode dobijene iskustvom (Učenje)

Epizoda 1

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

Epizoda 2

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

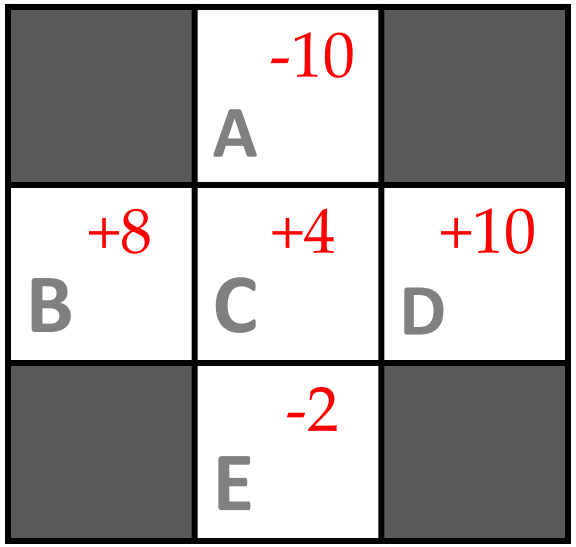
Epizoda 3

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

Epizoda 4

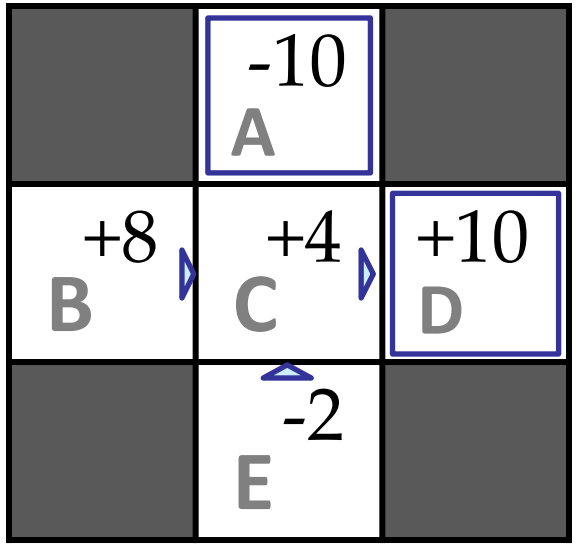
E, north, C, -1
C, east, A, -1
A, exit, x, -10

Rezultujuće vrednosti za stanja



Problemi sa Direktnom Evaluacijom

- Šta je dobro sa direktnom evaluacijom?
  - Laka je za shvatanje
  - Ne moramo da znamo T, R
  - Vremenom ćemo dobiti tačne vrednosti za stanja
- Šta je loše?
  - Ne koristimo informaciju o direktnoj povezanosti stanja (B i E vode samo u C, ali imaju drastično drugačije vrednosti)
  - Vrednosti za svako stanje moramo da učimo zasebno
  - Samim tim, treba mu jako puno vremena da nauči



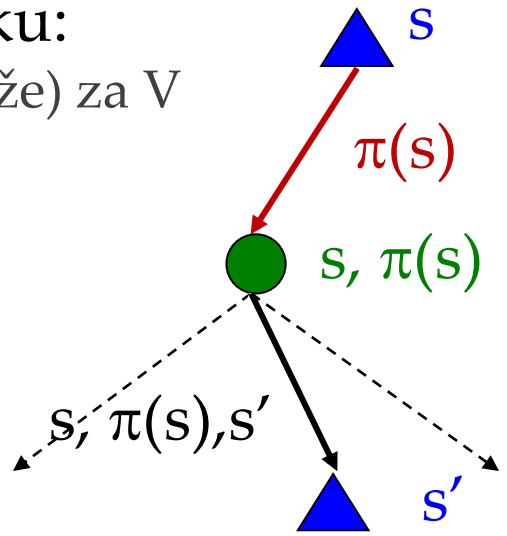
Ako po ovoj politici B i E oba vode u C zašto imaju toliko drugačije vrednosti?

Zašto Ne Koristimo Evaluaciju Politike?

- Uprošćene Belmanove jednakosti, računamo V za fiksnu politiku:
  - Za svaku iteraciju, zameni V sa jednim pogledom unapred (jedan sloj niže) za V

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

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



  - Ovaj pristup koristi veze između stanja
  - Nažalost, za njega nam trebaju T i R!

- Ključno pitanje: da li ovo možemo da uradimo bez T i R?
  - Drugim rečima, da li možemo da izračunamo prosek vrednosti pomnoženih težinama, ako te težine ne znamo? (da li vam ovo liči na primer sa godištem studenta?)

Evaluacija Politike Pomoću Semplova?

(drugi pasivni metod)

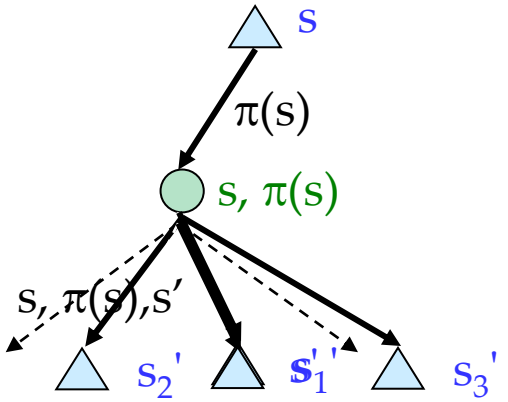
- Hoćemo da poboljšamo našu procenu V koristeći prosek semplova:

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

- Ideja: Prikupiti semplove u s (radeći akciju koju kaže politika!) i onda uzeti prosek

$$sample_1 = R(s, \pi(s), s'_1) + \gamma V_k^{\pi}(s'_1)$$
$$sample_2 = R(s, \pi(s), s'_2) + \gamma V_k^{\pi}(s'_2)$$
$$\ldots$$
$$sample_n = R(s, \pi(s), s'_n) + \gamma V_k^{\pi}(s'_n)$$

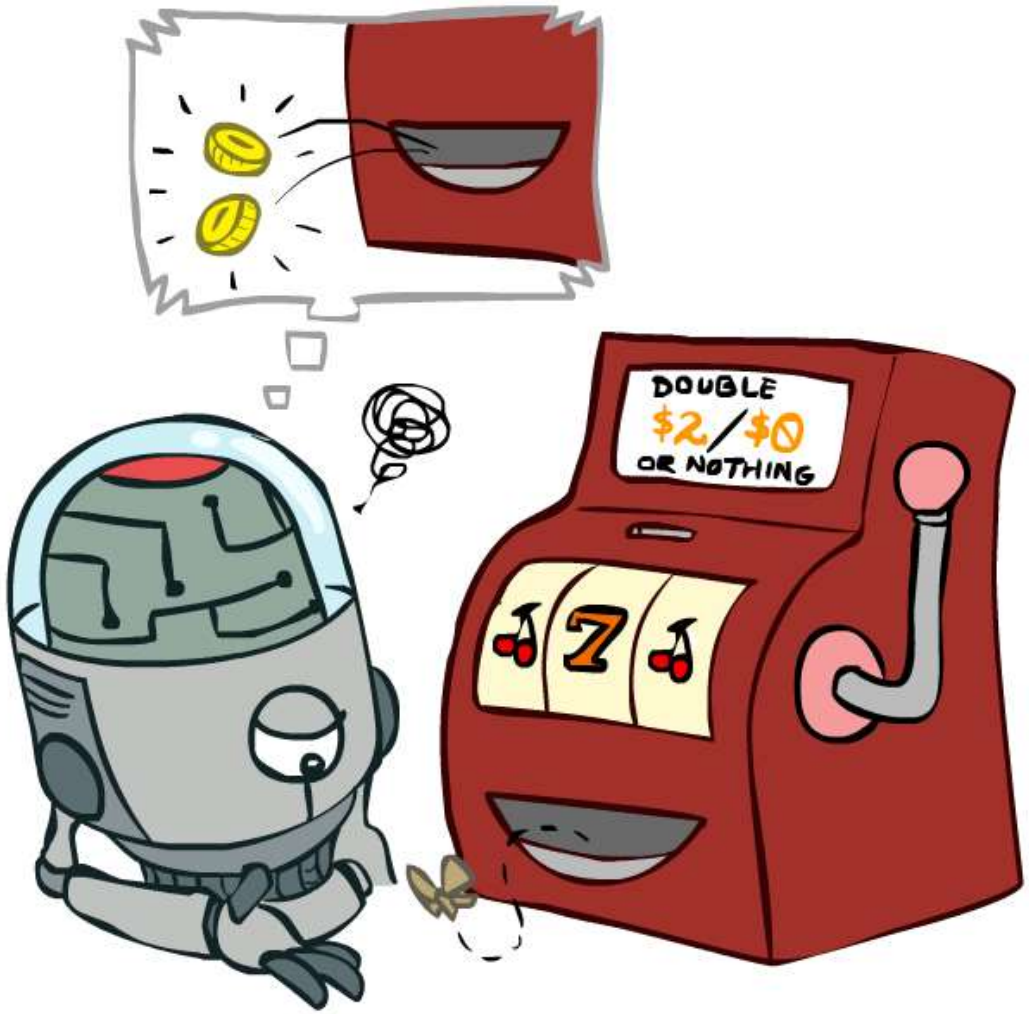
$$V_{k+1}^{\pi}(s) \leftarrow \frac{1}{n} \sum_i sample_i$$



Skoro smo tu! Ali da li je pametno vraćati stalno u s i raditi akcije dok ne dobijemo dovoljno semplova.

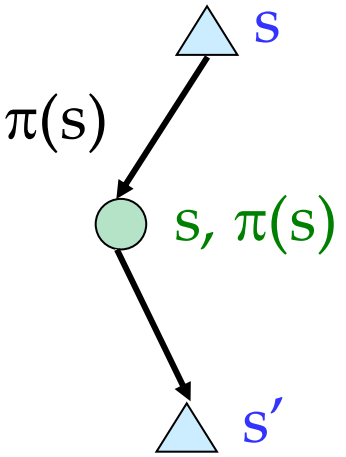
Temporal Difference Learning
Učenje Pomoću Razlike Kroz Vreme

(treći pasivni metod)



Učenje Pomoću Razlike Kroz Vreme (TLD)

- Generalna Ideja: učimo iz svakog iskustva (ne čekamo da se nakupi dovoljno semplova)!
  - Popravljamo V(s) svaki put kad nam se dogodi prelaz (s, a, s', r)



- TDL za vrednosti
  - Politika je i dalje fiksirana, i dalje radimo evaluaciju politike!
  - Menjamo vrednost V u pravcu iskustva koje nam se upravo dogodilo: tekući prosek (*running average*)

Uradili smo neku akciju u s, po politici i dogodilo nam se da smo dobili nagradu i prešli u s'.

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

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

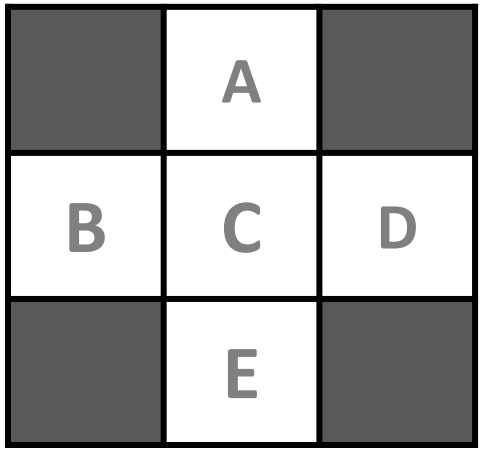
Drugačiji zapis: $V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha(sample - V^{\pi}(s))$

Eksponencijalno Pomerajući Prosek
Exponential Moving Average

- Eksponencijalno Pomerajući Prosek
  - Popravak koji koristimo: $\bar{x}_n = (1 - \alpha) \cdot \bar{x}_{n-1} + \alpha \cdot x_n$
  - Popravak je takav da su iskustva koje smo sad iskusili važnija od prethodnih
  - Zaboravljamo prošlost (jako stare vrednosti su i tako pogrešne)
- Ako postepeno smanjujemo alfa (tempo učenja - *learning rate*) konvergencija će biti brža
- Često se koristi alfa=1/n, gde n broji koliko puta smo promenili trenutno x.

Primer: Temporal Difference Learning

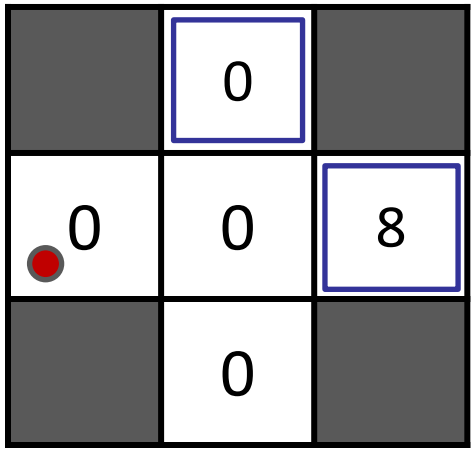
Stanja



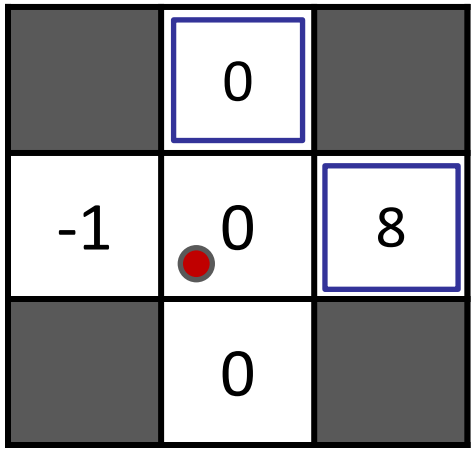
Koristimo: $\gamma = 1$, $\alpha = 1/2$

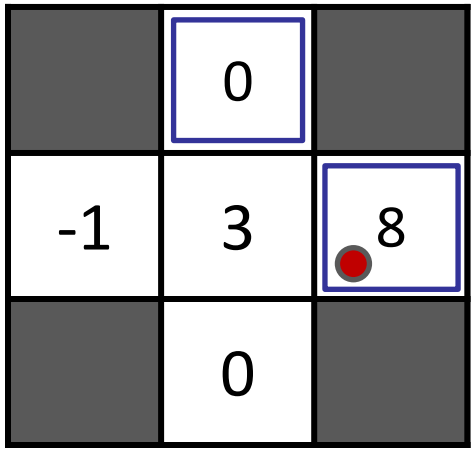
Iskustva koja su nam se dogodila

B, east, C, -2



C, east, D, -2





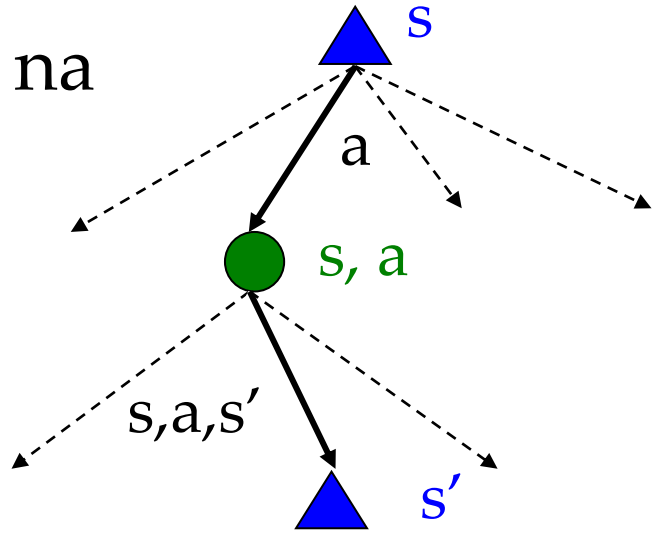
$$V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + \alpha\left[R(s,\pi(s),s') + \gamma V^{\pi}(s')\right]$$

Problemi sa TDL

- Šta je rezultat TDL: vrednosti za svako stanje.
- Da li možemo da pretvorimo te vrednosti u politiku?
- Nažalost ne, za način na koji smo to pre radili trebaju na T i R.

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

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



- Ideja: hajde da učimo Q-vrednosti (Q-values), a ne V vrednosti
- Sad je učenje selekcije akcija takođe bez modela!

Digresija: Q-Iteriranje Vrednosti

- Iteriranje vrednosti: odrediti sledeće (ograničene dubinom) vrednosti
  - Krećemo od $V_0(s) = 0$
  - Za dobijeno V_k, izračunavamo vrednosti k+1:

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

- ali, Q-vrednosti su korisnije (daju nam direktno politiku) pa računamo njih
  - Krećemo od $Q_0(s,a) = 0$
  - Za dobijeno Q_k, izračunavamo q-vrednosti za k+1:

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

Q-Učenje - Q-Learning

- Q-Učenje: Q-iteriranje vrednosti, ali ovaj put zasnovano na semplovima

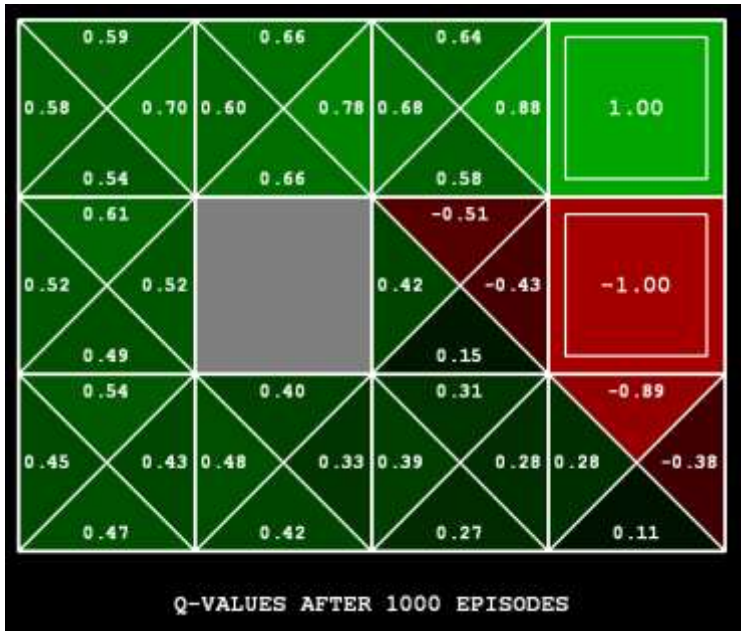
$$Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma \max_{a'} Q_k(s',a') \right]$$

- Učimo vrednosti Q(s,a) dok radimo akcije
  - Iskusimo (s,a,s',r)
  - Pogledamo našu staru procenu: $Q(s,a)$
  - Pogledamo trenutno iskustvo i očekivanu nagradu koju dobijamo:

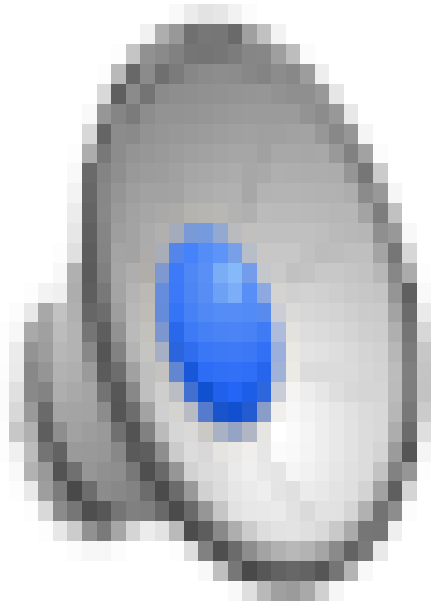
$$sample = R(s,a,s') + \gamma \max_{a'} Q(s',a')$$

  - Oba koristimo da dobijemo novu procenu:

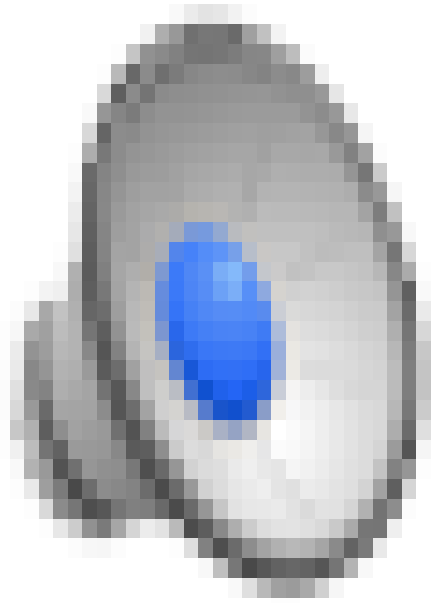
$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + (\alpha)\,[sample]$$



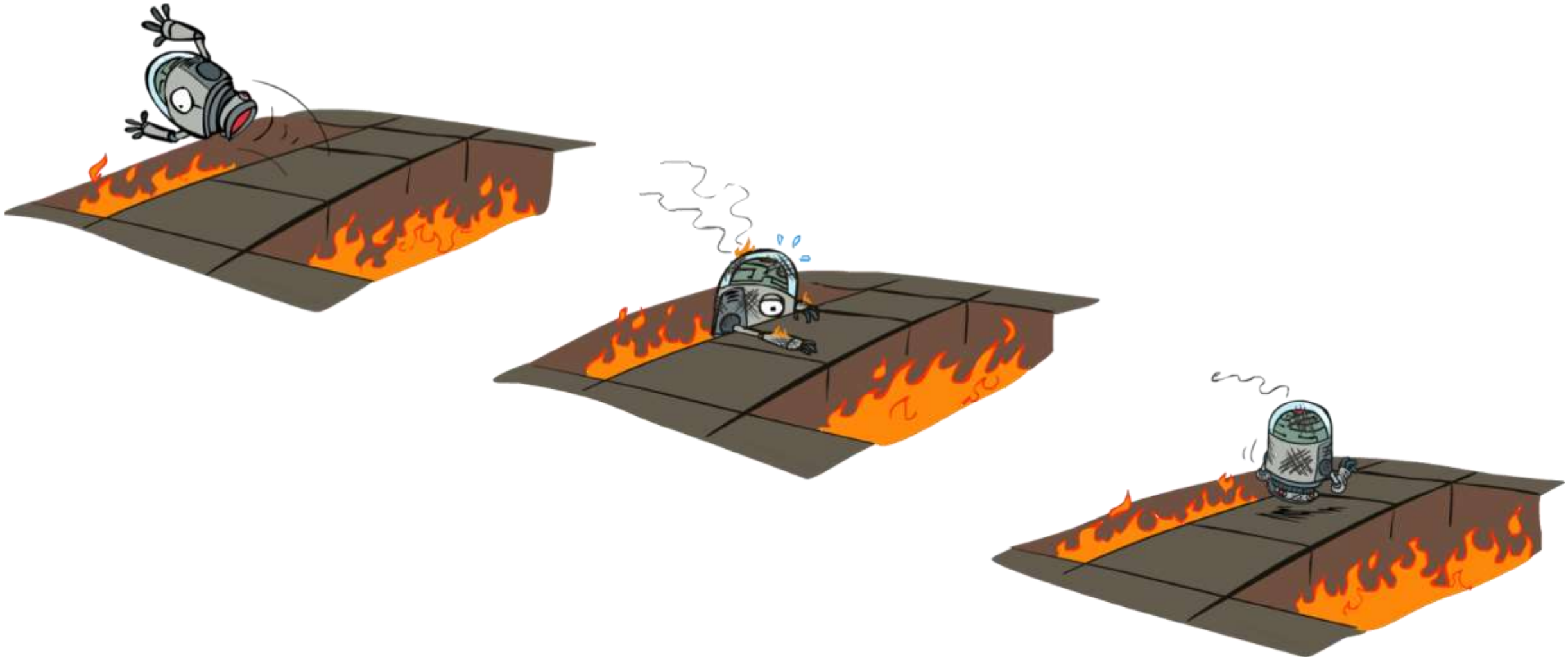
Demo Q-Učenje – Mrežasti Svet



Demo Q-Učenje – Robot Koji Puzi



Aktivno Učenje Uslovljavanjem



Aktivno vs. pasivno učenje

- Algoritmi koje smo pokazali do sada služili su nam da za fiksnu politiku odredimo vrednosti stanja (V) i stanja i akcija (Q).
- Nakon određivanja vrednosti politiku možemo korigovati tako što ćemo za svako stanje raditi akciju koja je argmax po Q.

- Da li možemo politiku da menjamo dok radimo akcije u okruženju?

- Možemo! To je aktivno učenje.

- Svaki put kada promenimo Q ili V vrednosti, osvežimo politiku. Detaljnije u nastavku kursa.

Q-Učenje:
radi po trenutno optimalnim akcijama (ali i istražuj…)

- Kompletno učenje uslovljavanjem: tražimo optimalnu politiku (slično iteriranju vrednosti)
  - Ne znamo T(s,a,s')
  - Ne znamo R(s,a,s')
  - Nemamo fiksnu politiku, agent sam bira akcije
  - Cilj: optimalna politika / vrednosti

- Sada:
  - Agent sam donosi odluke o akcijama! (na osnovu trenutnih Q vrednosti, a ne fiksne politike!)
  - Važna trampa (*tradeoff*): istraživanje vs. eksploatacija (*exploration vs. exploitation*)
  - Ovo nije offline planiranje! Agent stvarno radi akcije u okruženju i uči iz onoga što mu se događa.

Q-Učenje Karakteristike

- Značajan rezultat: Q-učenje konvergira do optimalne politike – čak iako dok učimo radimo akcije koje nisu optimalne!

- Na šta moramo da obratimo pažnju:
  - Moramo dovoljno da istražujemo
  - Takođe polako moramo da smanjimo tempo učenja
  - … ali ne prebrzo