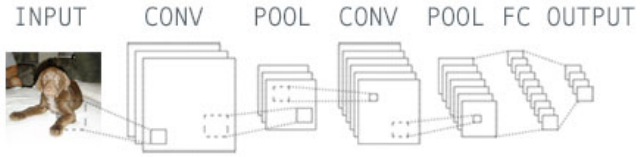
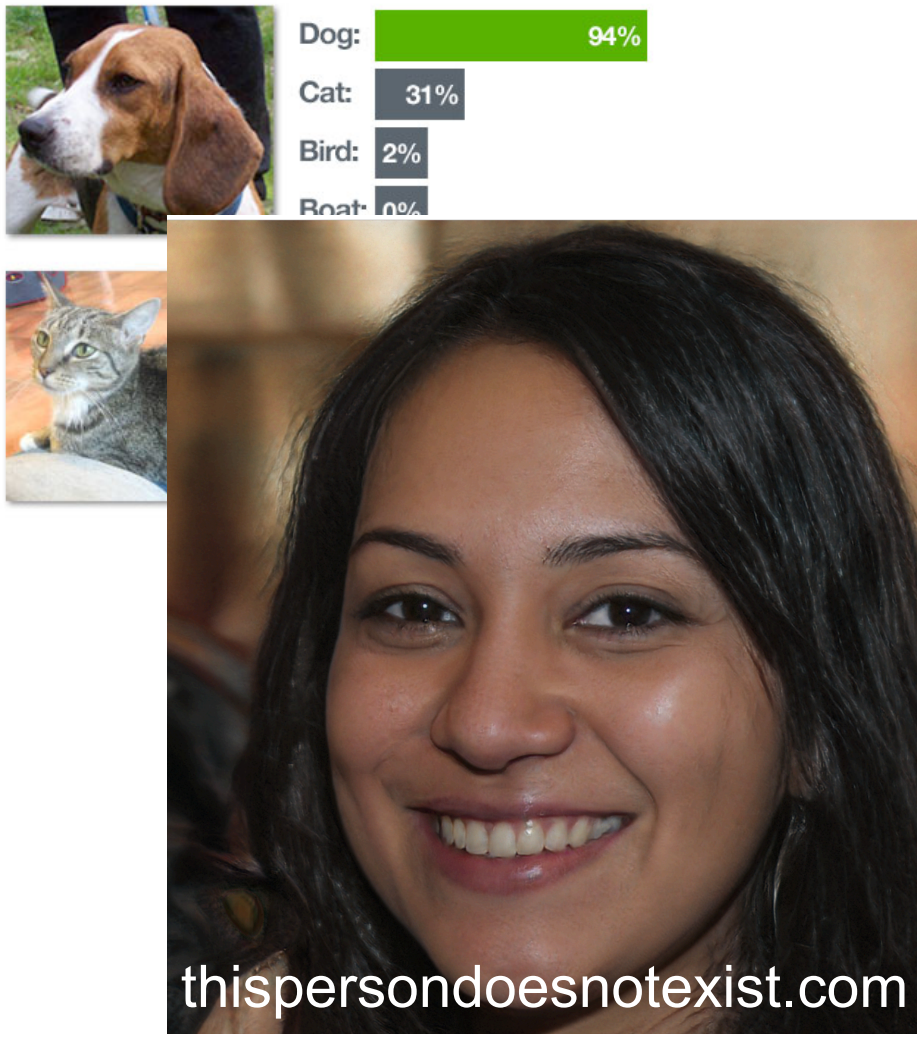
Statisztikai tanulás az idegrendszerben, 2019.

Stratégiák tanulása az agyban

Bányai Mihály
banyai.mihaly@wigner.mta.hu
http://golab.wigner.mta.hu/people/mihaly-banyai/

Kortárs MI


INPUT
CONV
POOL
CONV
POOL
FC
OUTPUT


Dog: 94%
Cat: 31%
Bird: 2%
thispersondoesnotexist.com


AlphaGo
Lee Sedol


Viper

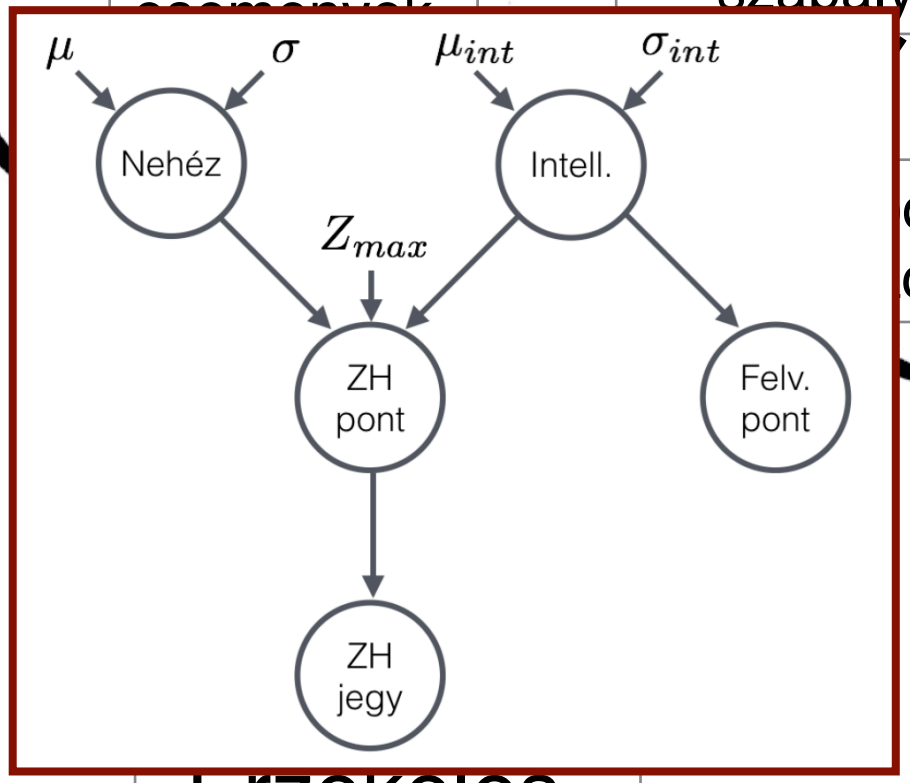
Tudás

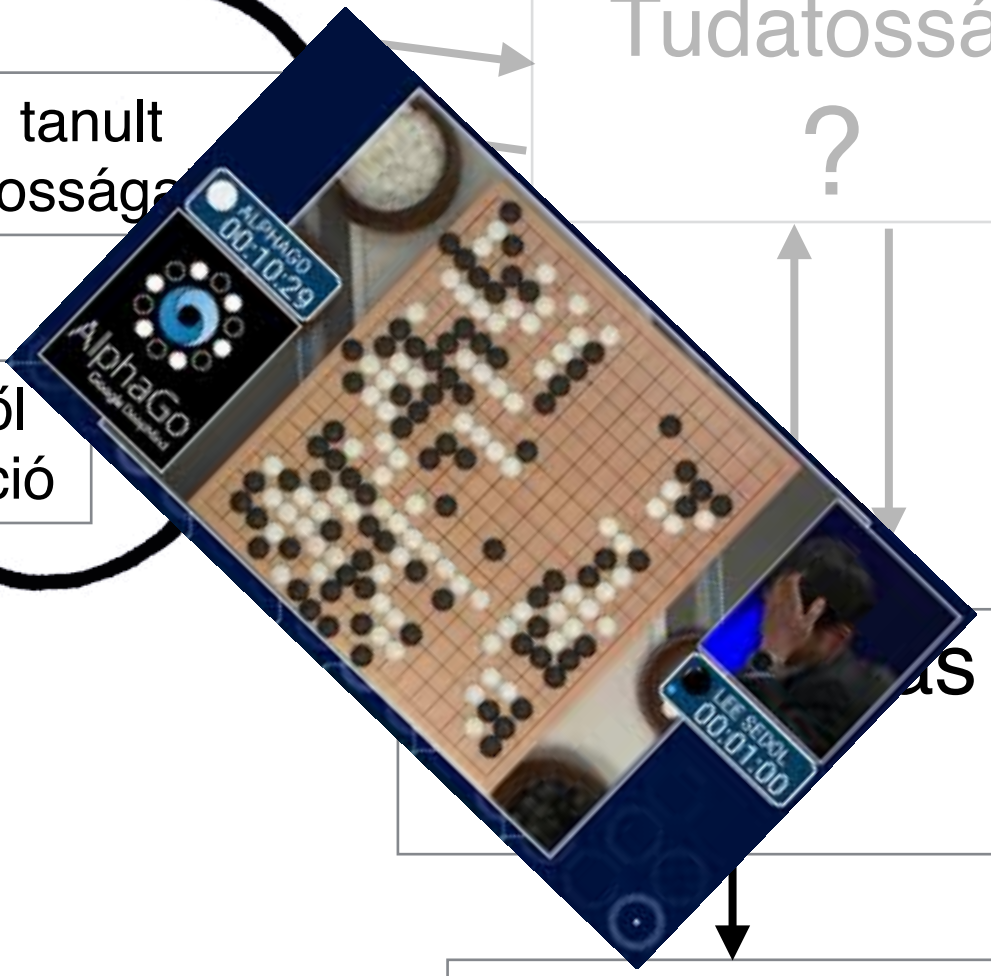
Tudatosság
?

Múltbeli

A világ tanult
szabályossága

ől
ció





Érzékelés

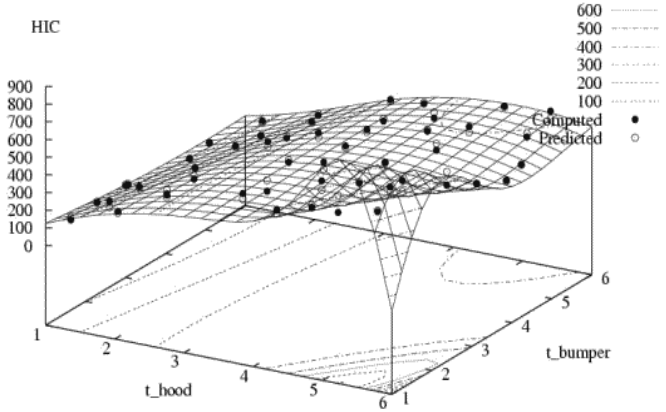
Izomvezérlés

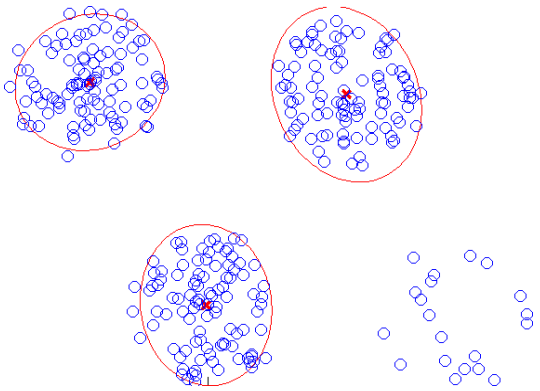
Hogyan építsünk döntési modellt az érzékelési modellre?

- Választanunk tell egy optimalizálandó célváltozót - ezt hívjuk jutalomnak (reward)
- A motoros kimenetet véges mennyiségű fix cselekvésként modellezük, amelyek módosítják a környezetet
- Az érzékelési model látens változóira adott megfigyelés mellett következtetett értékek kombinációja a környezet (érzékelt) állapota
- Ki tell találnunk, hogy a környezet mely állapotában mely cselekvést válasszuk

A tanulás alapvető típusai

- Felügyelt
  - Az adat: bemenet-kimenet párok halmaza
  - A cél: függvényapproximáció, klasszifikáció
- Megerősítéses
  - Az adat: állapotmegfigyelések és jutalmak
  - A cél: optimális stratégia a jutalom maximalizálására
- Nem felügyelt, reprezentációs
  - Az adat: bemenetek halmaza

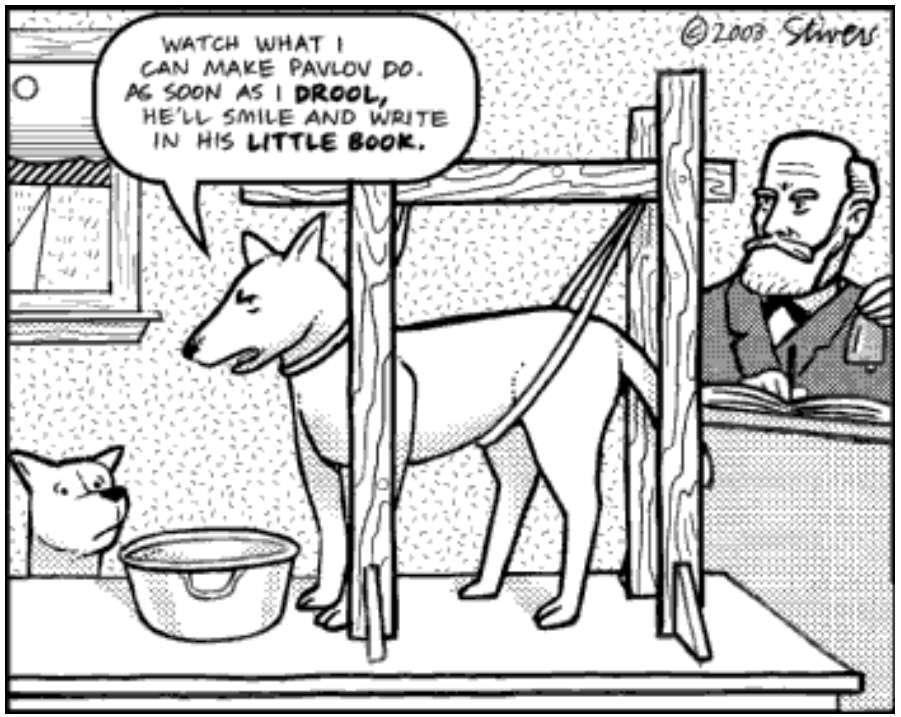


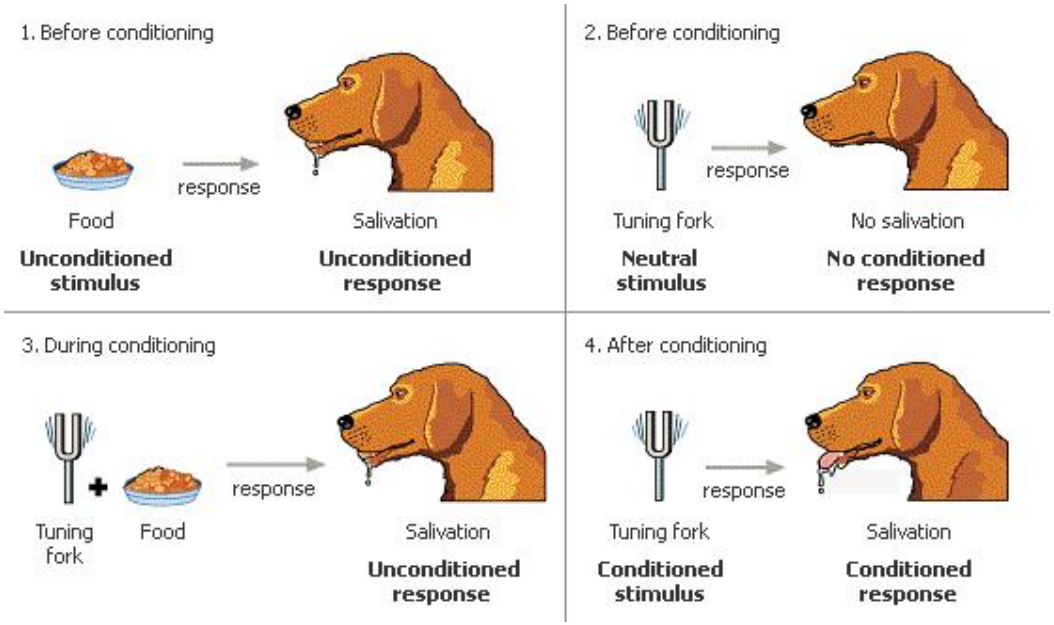


A jutalom előrejelzésének tanulása

Pavlovi - klasszikus kondicionálás

- csak az állapotok értékét tanuljuk, cselekvés nincs

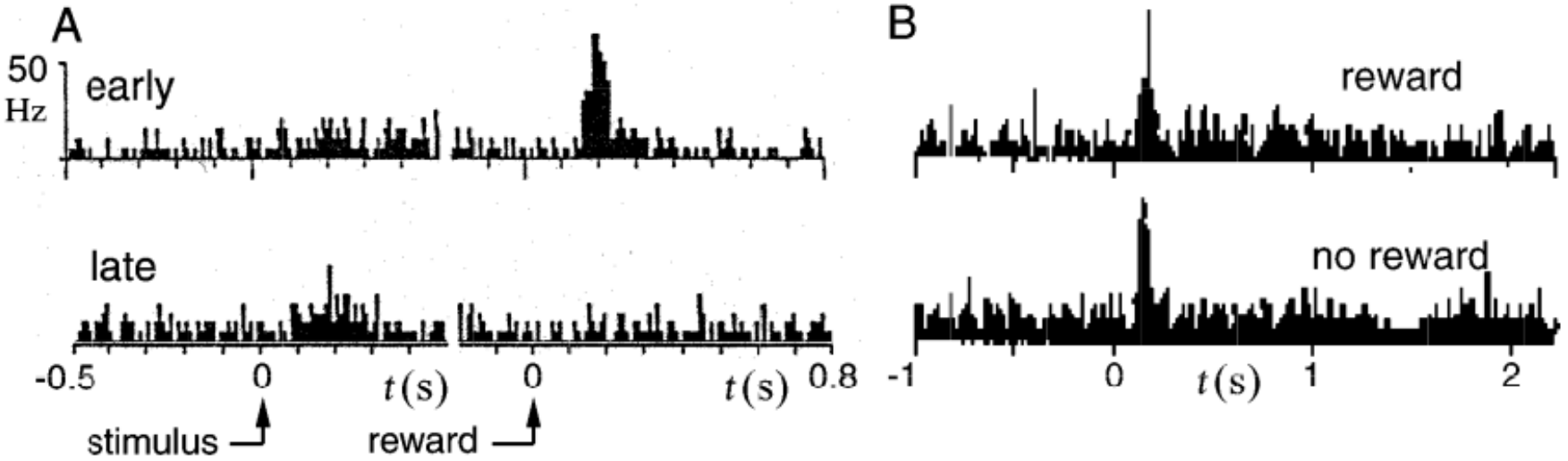




operáns kondicionálás

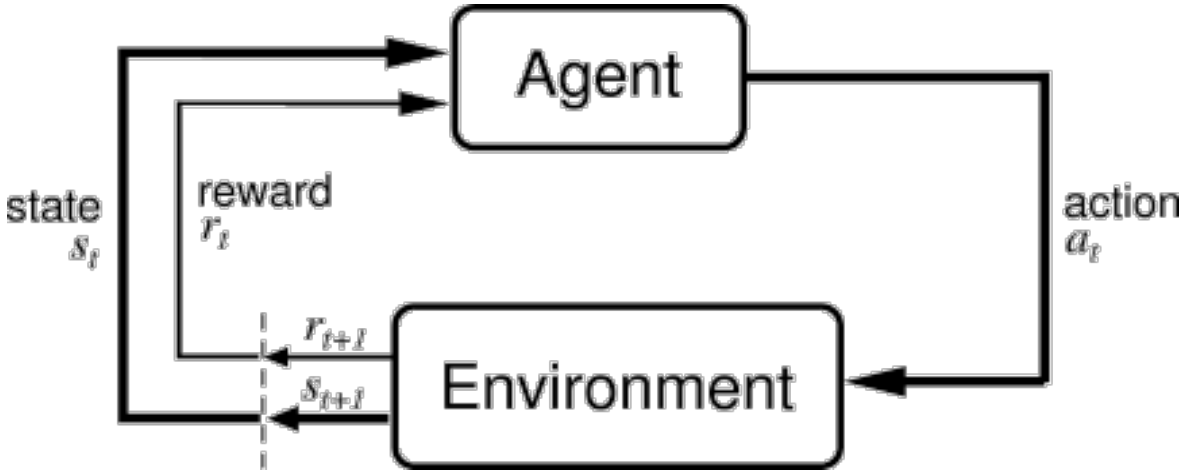
- a cselekvések határozzák meg a jutalmat

A jutalom kódolása az agykéregben



- Egyes dopamintermelő neuronok a majom agykérgében a jutalom és egyes környezeti változók között tanult kapcsolat alapján tüzelnek
- Az aktivitás a meglepetés mértékével és előjelével arányos

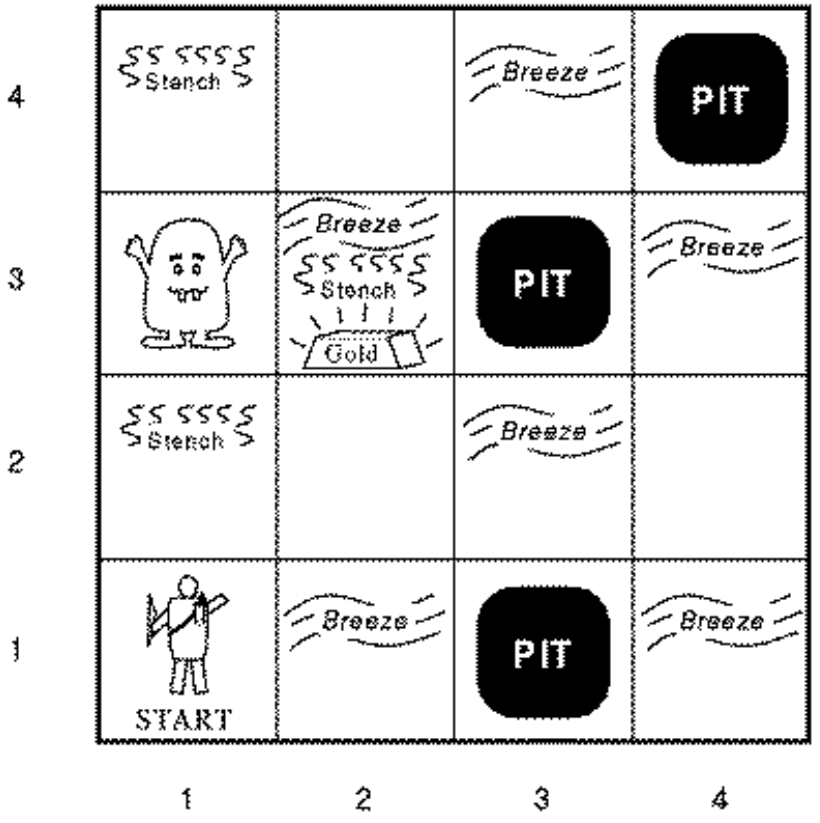
Az ágens-környezet modell



- Ágens: a tanulórendszer
- Környezet: minden, amit nem belsőleg állít elő
- Bemenet: aktuális állapot, jutalom
- Kimenet: cselekvés

Állapotok értékének tanulása

- Legegyszerűbb eset: véges mennyiségű állapotból áll a környezet
- minden állapot, *s,* a mentális model lateens változóinak ego kikövetkeztetett értékkombinációja (pl. vehetjük az *a posteriori* legvalószínűbb értékeket)
- minden állapothoz hozzárendelünk egy értéket, *V(s)*, amely az állapot kívánatosságát kódolja
- minden döntés után, *t* időpillanatban, frissítjük az állapotértékek becslését a korábbi becslések és az aktuális jutalom alapján
- intuíció: egy állapot értékét meghatározza az abban kapható jutalom és a belőle kevés cselekvéssel elérhető további állapotok értéke



Russell & Norvig, 1995

https://cs.stanford.edu/people/karpathy/reinforcejs/gridworld_td.html

A jutalom hosszútávú maximalizációja

- Összesített jutalom: valamilyen becslést kell alkalmazni a jövőre nézve
- Különbség a felügyelt tanulástól: egyiknél azt tudjuk, hogy a képtérben milyen messze vagyunk az optimálistól, a másiknál arra kapunk egy becslést, hogy a paramétertérben mennyire térünk el az optimálistól

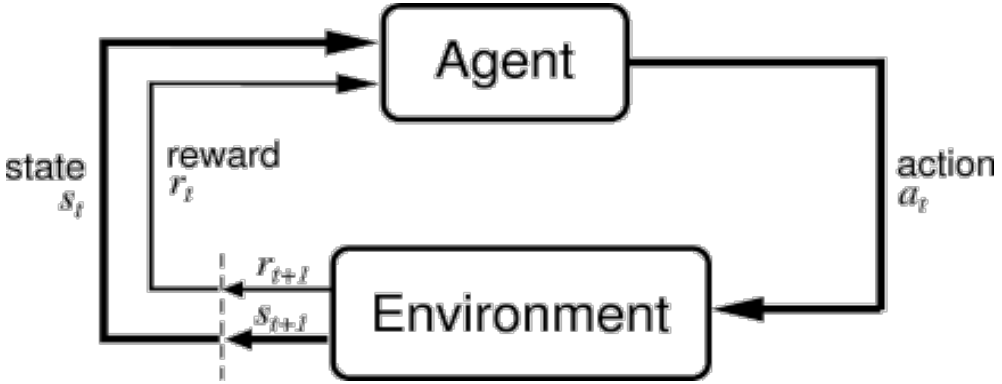
Temporal Difference tanulás

- A predikciós hibából tanulunk
- Az aktuálisan meglátogatott állapot értéke az előzőleg meglátogatottét önmagához hasonlóbbá teszi
- Korábban meglátogatott állapotokra is visszaterjeszthetjük a változtatást

$$V(S_t) \leftarrow \underbrace{V(S_t)}_{\text{Previous estimate}} + \alpha[\underbrace{R_{t+1}}_{\text{Reward t+1}} + \underbrace{\gamma V(S_{t+1})}_{\text{Discounted value on the next step}} - V(S_t)]$$

TD Target

Cselekvésválasztás



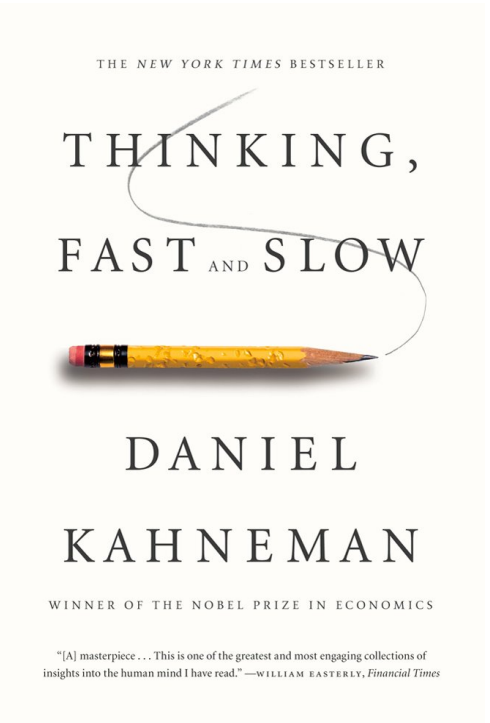
- Az ágens a környezet állapotaihoz értéket rendelő függvény tanul
  - **V**: s -> R
- Ahhoz, hogy kiválaszthassunk egy cselekvést, tudnunk kell, hogy az aktuális állapotból melyik cselekvés melyik állapotba vezet át
  - **M**: (s,a) -> s
  - illetve általánosabban M: (s1,s2,a) -> P
  - ezt az M függvényt nevezzük a környezet **modelljének**
- Minden olyan RL megoldásban, ahol állapotérték-függvényt tanulunk, meg kel tanulnunk a modellfüggvényt is
  - ezeket **modellalapú** megoldásoknak nevezzük

Modellmentes RL

- A környezet modelljének megtanulása helyett tannulhatjuk közvetlenül állapot-cselekvés párok értékét
  - **Q**: (s,a) -> R
  - hasonló tanulóalgoritmusok használhatóak ebben az esetben is
  - a cselekvésválasztáshoz csak a Q függvényre van szükség, modellre nem

Deliberatív és reaktív rendszerek

- A Q függvények (#állapotok x #cselekvések) mennyiségű parametere van, míg a V-nek csak (#állapotok)
  - sokkal több adatra van szükségünk a Q becsléséhez mint a V-éhez
  - Modellalapú megoldásokban az M újrahasznosítható új feladatnál, a Q-t nulláról kell újratanulni
- Modellalapú megoldásokban sokkal számításigényesebb a cselekvésválasztás
  - de a modell segítségével készíthetünk hosszútávú terveket
  - a modellmentes megoldások reflexek halmazához hasonlók
  - a kit rendszer kiegészítheti egymást, főként egy feladat elsajátításának különböző fázisaiban



Exploration vs. exploitation

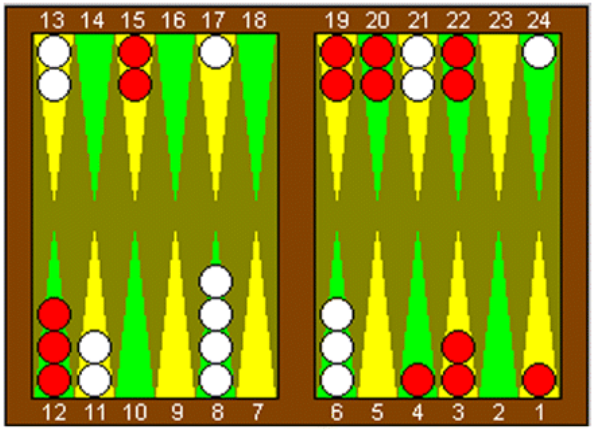
- Amikor még keveset tudunk a környezetről, nincs értelme az első jutalomhoz vezető cselekvéssort ismételni a végtelenségig
- Amikor többet tudunk, akkor viszont érdemes a lehető legjobb stratégiát alkalmazni folyamatosan
- Szokásos megoldás: próbálkozzunk véletlenszerűen az elején, és fokozatosan emeljük a legjobbnak becsült cselekvés választásának valószínűségét

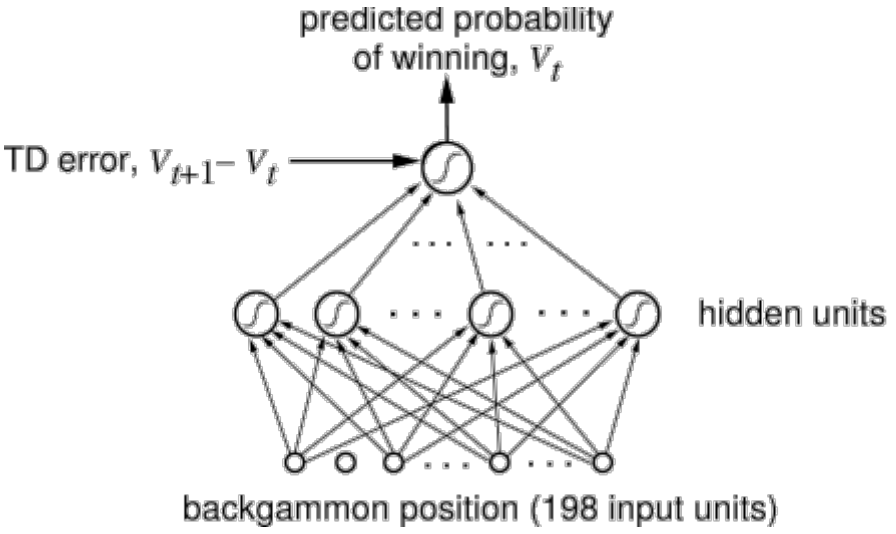
Az állapotérték-függvény reprezentációja

- Ha nincs olyan sok érték, használhatunk táblázatot
- Nagy illetve esetleg folytonos állapotterekben
  - csak az állapotváltozókat tudjuk közvetlenül reprezentálni, nem az összes lehetséges érték-kombinációt
  - általánosítanunk kell olyan állapotokra, amiket sosem látogattunk meg korábban
  - a többrétegű neurális hálózatok alkalmasak mindkét probléma megoldására
  - mivel ezek tanításához szükséges egy elvárt kimenet minden lépésben, ezt a TD algoritmus predikciós hibájából kell megalkotnunk

Optimális döntések tanulása neurális hálózattal

- Gerald Tesauro: TD-Gammon
  - Többrétegű neuronhálózat
  - Bemenet: a lehetséges lépések nyomán elért állapotok
  - Kimenet: a nyerés valószínűsége az adott állapotból
- Ez alapján ki lehet választani, hogy melyik állapotba szeretnénk kerülni
- Eredmény: a legjobb emberi játékosokkal összemérhető
- A teljes tanítási folyamat ma: 5s





TD neurális reprezentációval

- Prediction error felhasználása a tanuláshoz
- Az állapotérték frissítése neurális reprezentációban:

$$w(\tau) \leftarrow w(\tau) + \varepsilon\delta(t)u(t-\tau) \qquad \delta(t) = \sum_t r(t+\tau) - v(t)$$

- A prediction error kiszámítása
  - A teljes jövőbeli jutalom kellene hozzá
  - Egylépéses lokális közelítést alkalmazunk

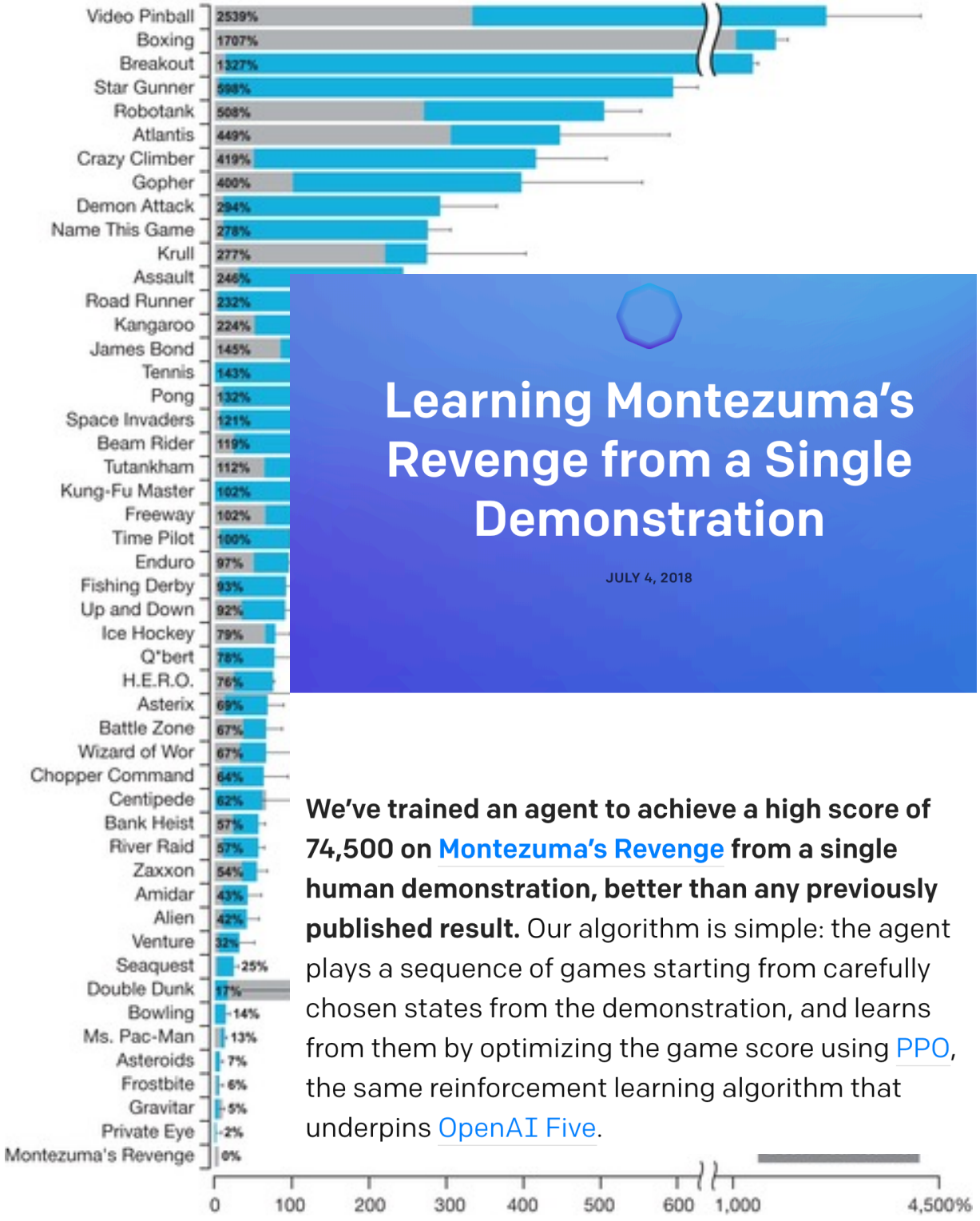
$$\sum_t r(t+\tau) - v(t) \approx r(t) + v(t+1)$$

  - Ha a környezet megfigyelhető, akkor az optimális stratégiához konvergál
- A hibát visszaterjeszthetjük a korábbi állapotokra is

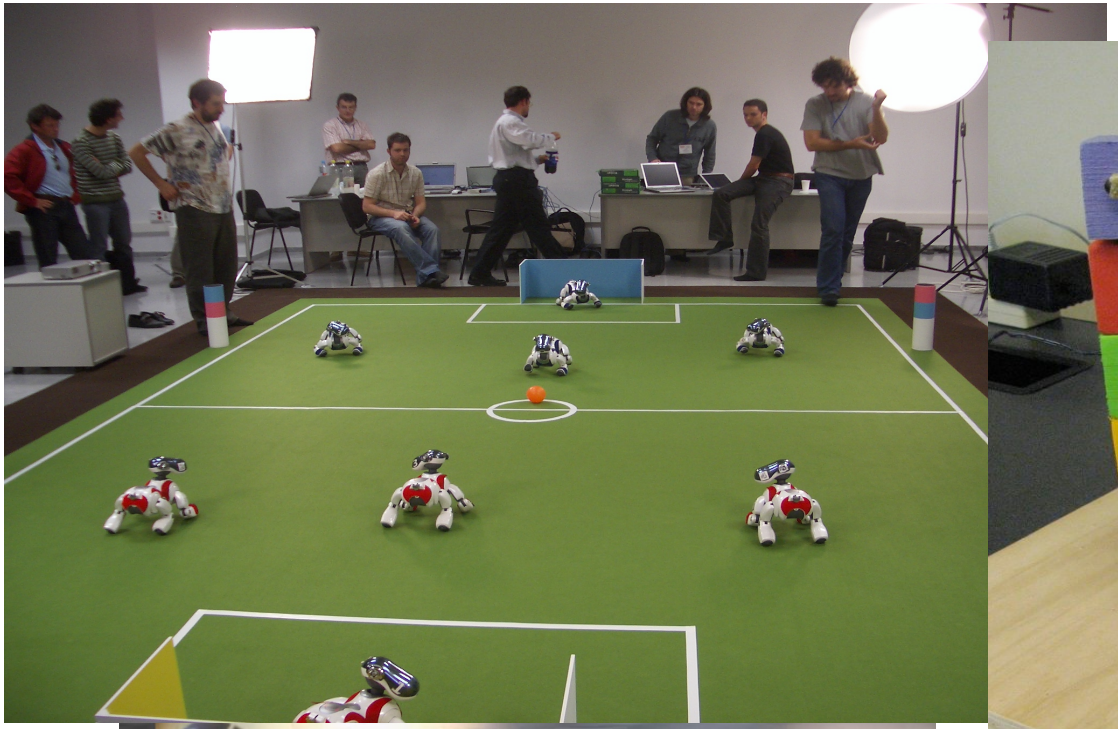
Számítógépes játékok tanulása

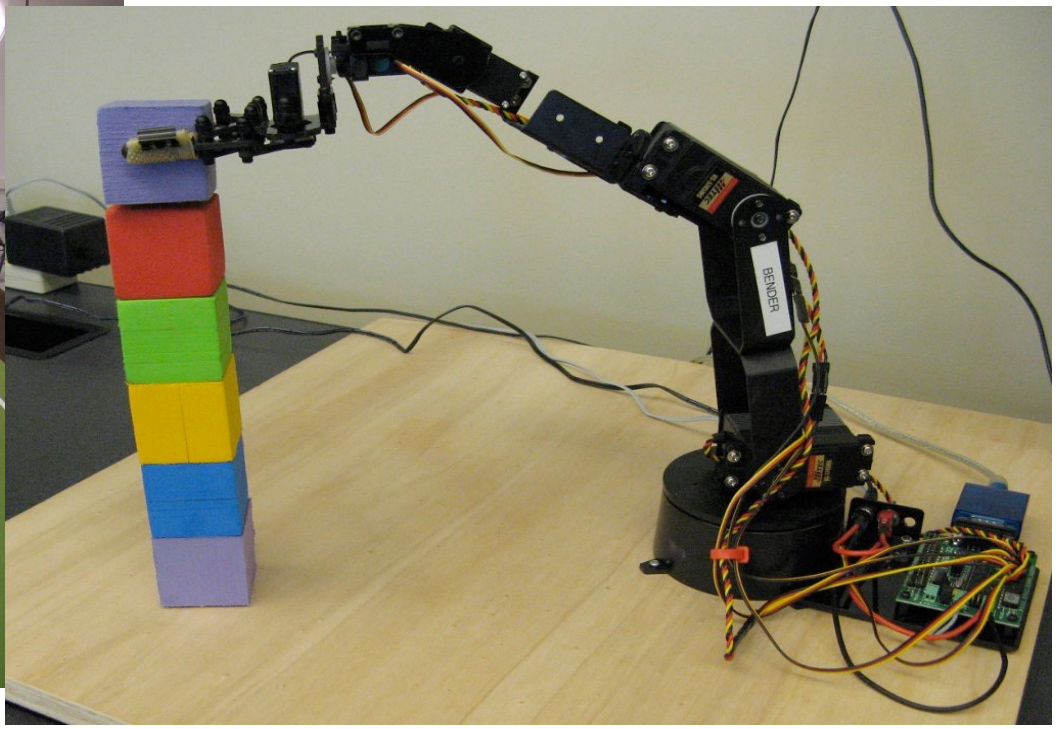
- reinforcement learning + deep networks
- megfigyelés: a képernyő pixelei + pontszám

Mnih et al, 2015

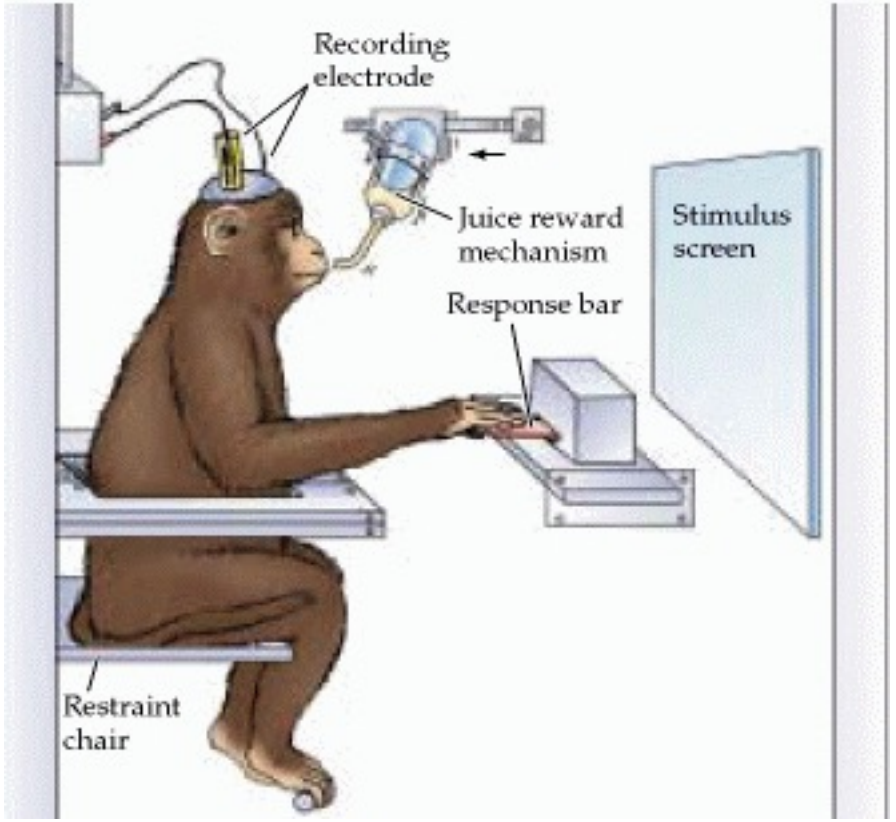


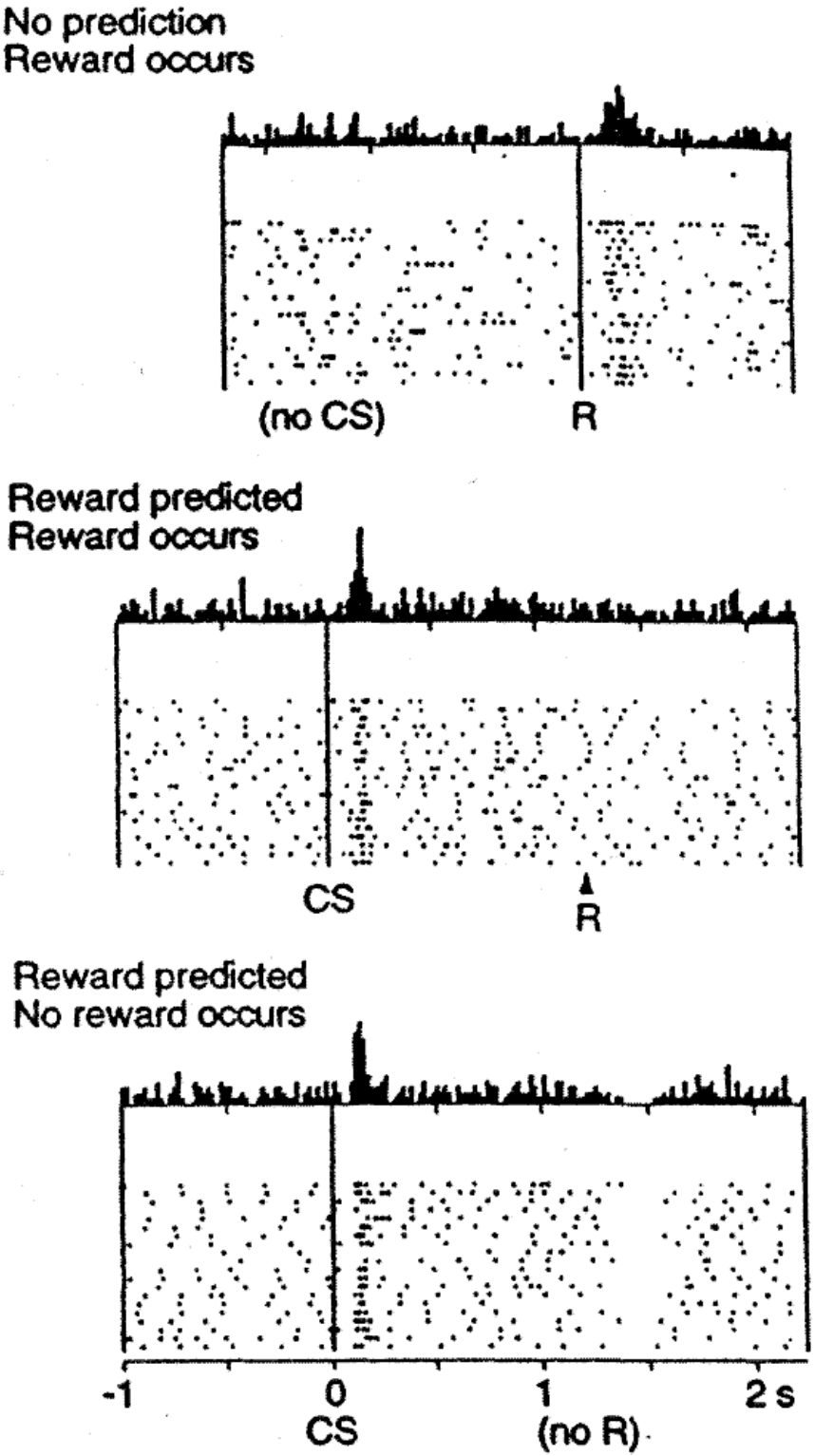
Fizikai mozgás



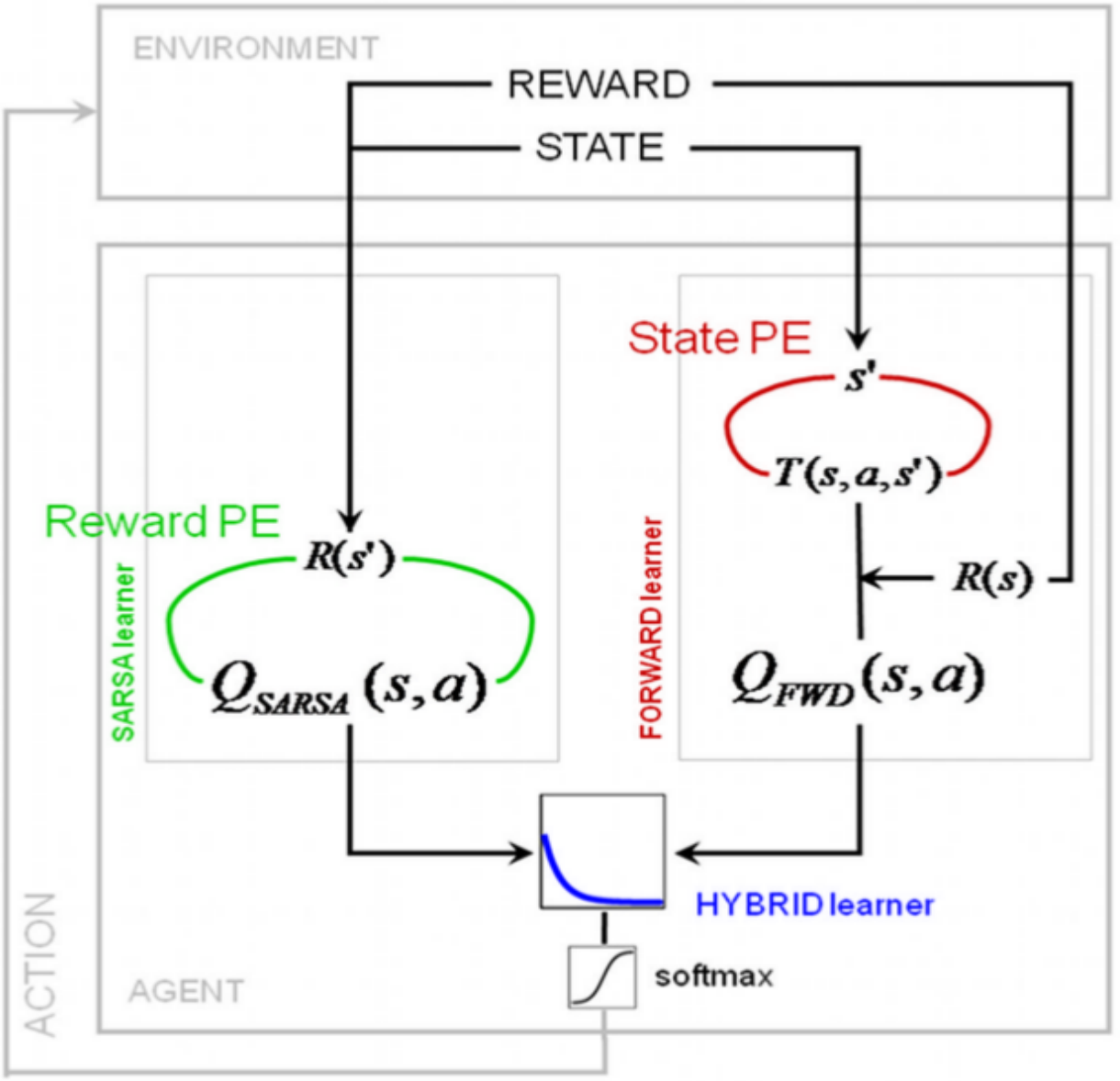


A jutalom reprezentációja a agyban





RL-változatok korrelátumai az agyban

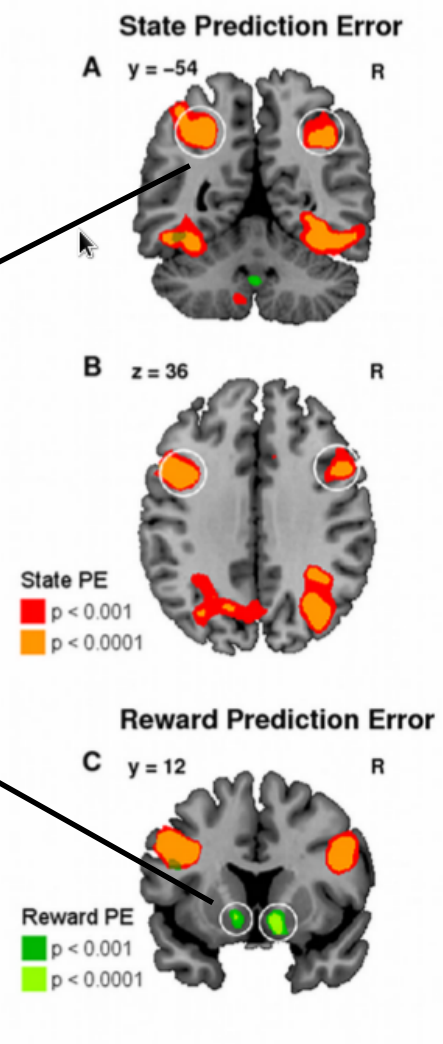


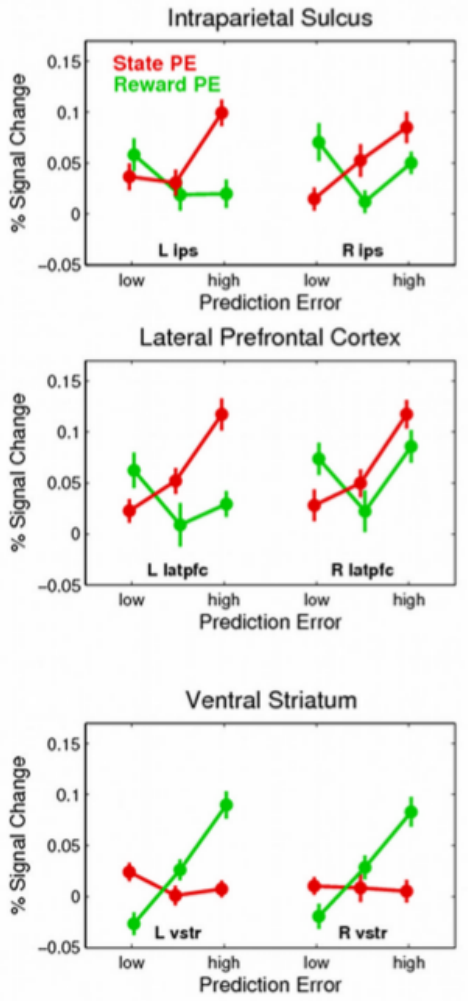
Glascher, Daw, Dayan & O'Doherty, Neuron, 2010

RL-változatok korrelátumai az agyban

Deliberatív modellalapú RL
prefrontális és parietális kéreg

Reaktív modellmentes RL
kéreg alatti struktúrák





The Big Program of the Brain

```
function MotorOutput = Brain(SensoryInput)

if (Dead ~= 1)
    for i = 1: NumSenses
        ExtractedFeatures(i) = ProcessSense(i, SensoryInput);
    end
    WorldModel = UpdateWorldModel(ExtractedFeatures,InternalState);
    MotorOutput = GenerateAction(WorldModel);
end
```

HF

- Add meg a kincskeresős játékhoz tartozó állapot- és cselekvésteret
- Valósítsd meg valamilyen programnyelven az ágens és a környezet szimulált interakcióját
- Q-learning segítségével tanuld meg, melyik állapotban melyik cselekvés a leghasznosabb
- Eredményedet ábrákkal illusztráld