

Üzleti Elemzések Módszertana

11. Előadás: Megerősítéses tanulás

Kuknyó Dániel
Budapesti Gazdasági Egyetem

2023/24
2.félév

1 Bevezetés

2 Politika javítása

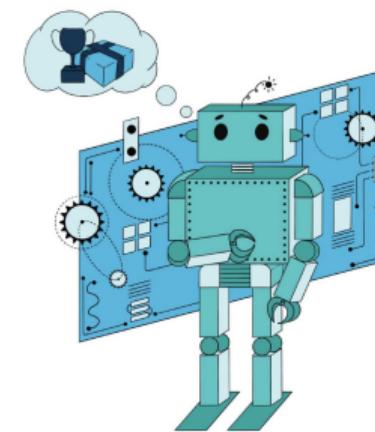
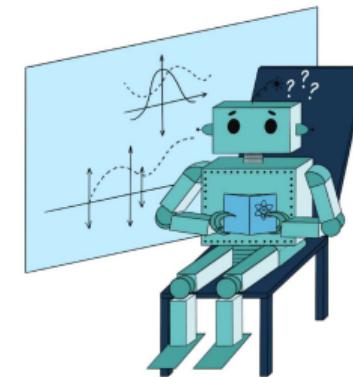
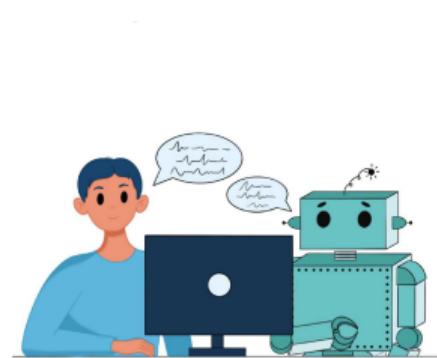
1 Bevezetés

2 Politika javítása

A gépi tanulás fajtái

A gépi tanulás 3 fő irányzata:

- Felügyelt tanulás
- Felügyelet nélküli tanulás
- Megerősítéses tanulás



Mikor alkalmazható a megerősítéses tanulás?

Az RL olyan problémák esetén használatos, ahol az **algoritmikus vagy hagyományos ML hozzáállás nem bizonyul megfelelőnek**, mert nem lehetséges tanító adatot gyűjteni vagy generálni.

Például:

- Robotok
- Autonóm vezetés
- Számítógépes játékok



Felügyelt vagy megerősítéses tanulás?

Adott például egy autóversenyző program. Ha felügyelt tanítás a választott hozzáállás, szükség van egy adatbázisra, amely jellemzi az összes szituációt, és minden szituációhoz tartozóan az elvárt output értéket.

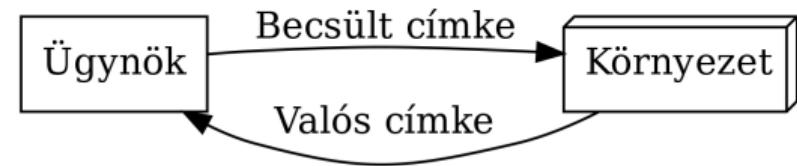
A szituációknak le kell írnia a kocsi helyzetét, a környezet állapotát, a versenytársak helyzetét. Az elvárt outputnak olyan halmazból kell kikerülnie, mint gáz, jobb, bal, fék és ezek kombinációi.



Visszajelzések a megerősítéses tanulásban

A két szemléletmód abban különbözik,
hogy a felügyelő milyen visszajelzéseket ad
a tanulónak.

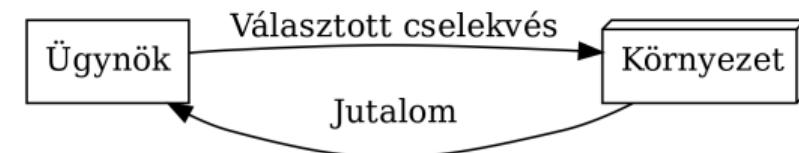
**A felügyelt tanulásban teljes
visszajelzésekéről van szó, mert a válasz
önmagában a megoldás.**



Visszajelzések a megerősítéses tanulásban

A két szemléletmód abban különbözik,
hogy a felügyelő milyen visszajelzéseket ad
a tanulónak.

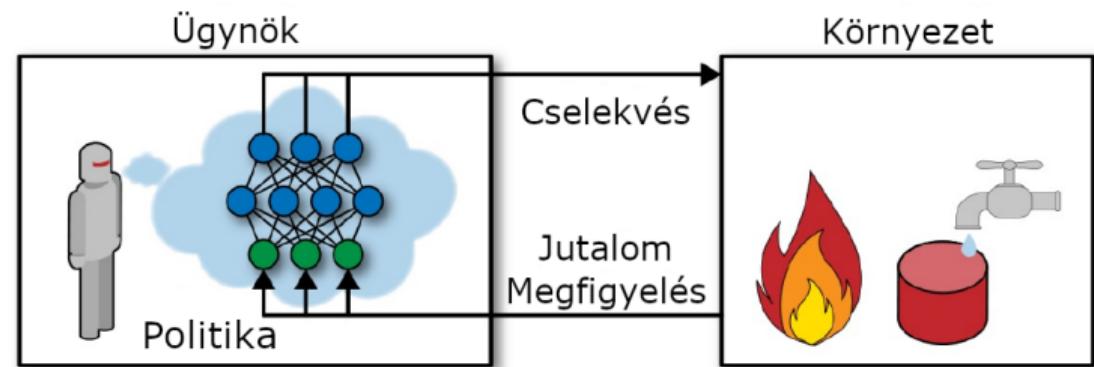
**A megerősítéses tanulásban viszont
csak részlegesek a visszajelzések. A
felügyelő válasza mindenig csak a
megoldás irányába vezet, nem
önmagában a teljes jó megoldás.**



A megerősítéses tanulás komponensei

Ügynök

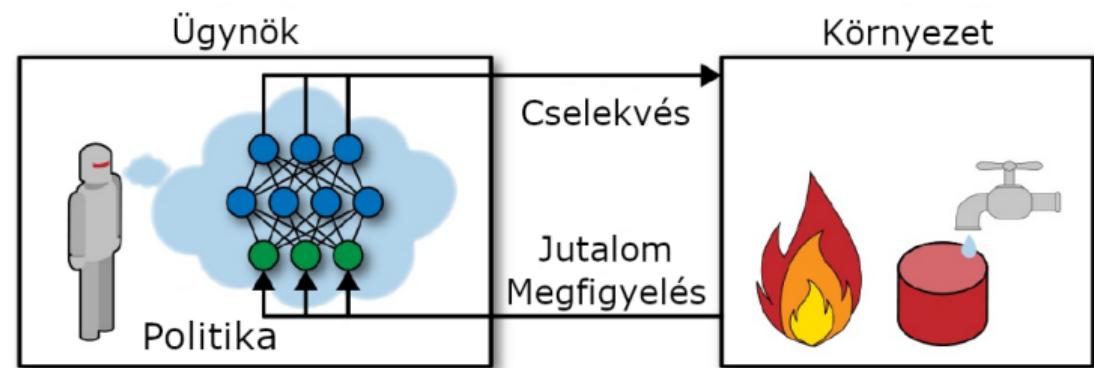
Az autonóm cselekvő, ami a feladat végrehajtására törekszik.



A megerősítéses tanulás komponensei

Környezet

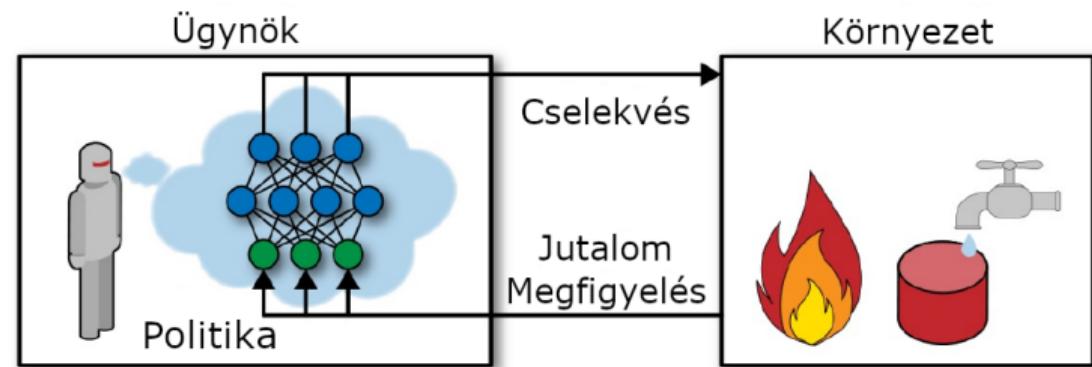
Egy fekete doboz, amely az ügynök cselekvéseinek helyszíne.



A megerősítéses tanulás komponensei

Idő

RL folyamán az időlépések diszkrétek:
 $t \in 1, 2, 3, \dots$



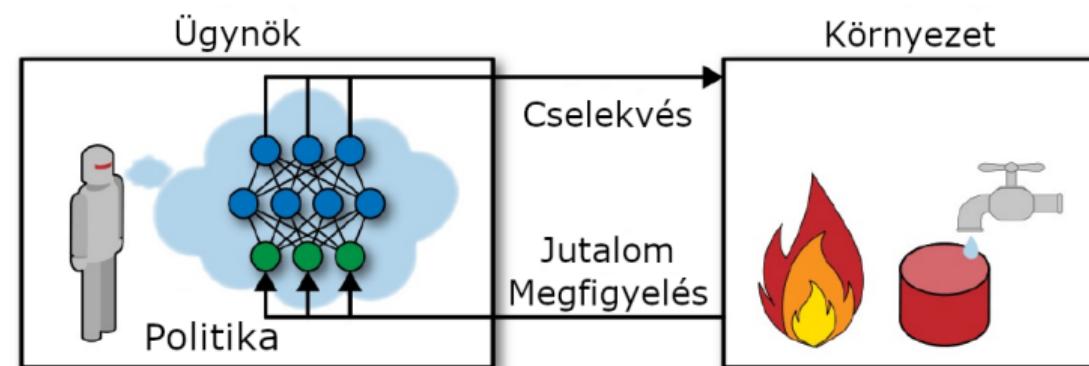
A megerősítéses tanulás komponensei

Állapot

Az ügynök megfigyelése a környezetre vonatkozóan.

A környezetet leíró változók összessége.

Jelölés: $s \in S$, ahol S az összes állapot halmaza.

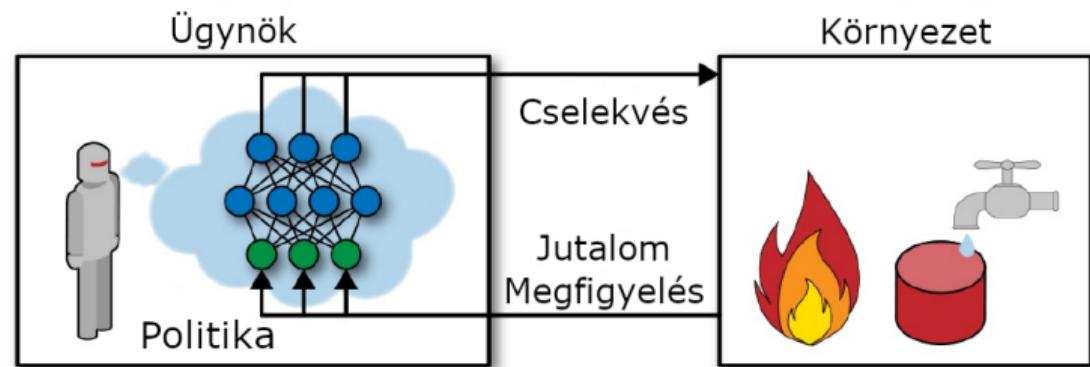


A megerősítéses tanulás komponensei

Jutalom

Az ügynök cselekvésének
jóságát jelző skalár.

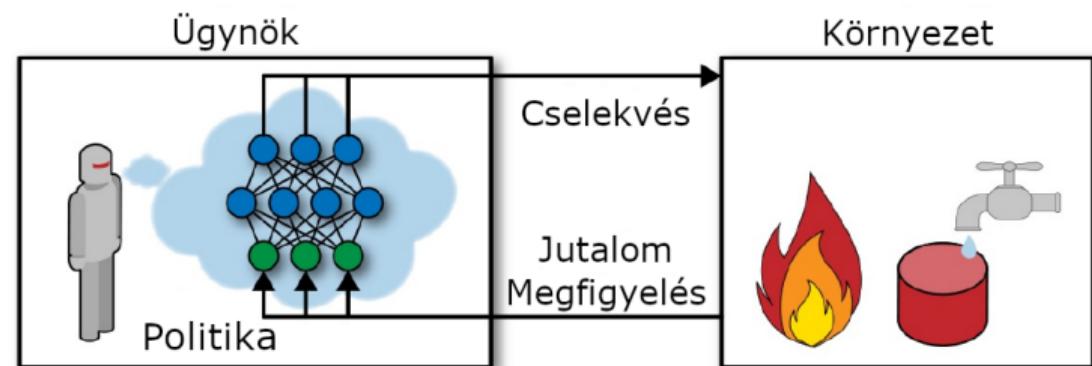
Jelölés: $r \in \mathbb{R}$



A megerősítéses tanulás komponensei

Cselekvés

Az ügynök által végrehajtott művelet, ami a környezetet befolyásolja. Jelölés: $a \in A$, ahol A az összes cselekvés halmaza.



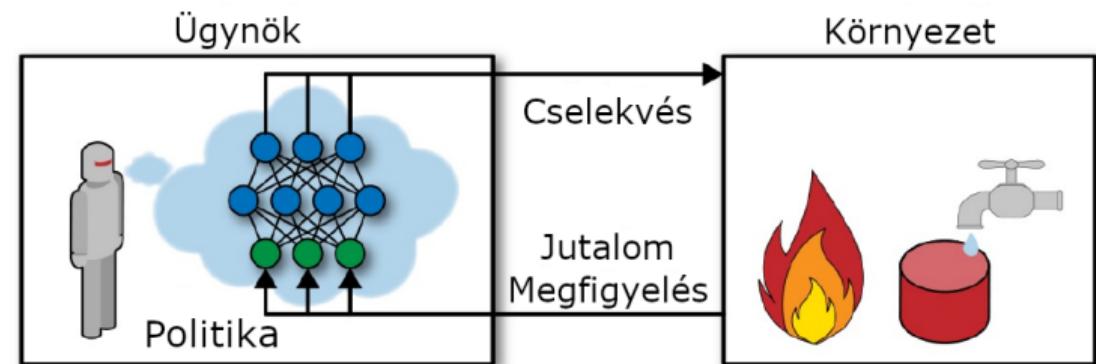
A megerősítéses tanulás komponensei

Politika

Egy állapot \rightarrow cselekvés leképezés. Az ügynök cselekvéseinek szabályait adja meg.

Jelölés:

- Determinisztikus:
 $\pi \in S \rightarrow A$
- Sztochasztikus:
 $\pi \in S \times A \rightarrow [0, 1]$
Röviden: $\pi(s, a)$
Vagy: $\pi(a|s)$



Az ügynök

A megerősítéses tanulásban egy ügynök (**cselekvő**) megfigyeli a környezetet, és ezalapjából cselekvéseket tesz a környezetben. A cselekvéseiért és a környezet változásáért jutalmat kap.

Az ügynök célja, hogy a jutalmakat hosszú távon maximalizálja. Az ügynök lehet:

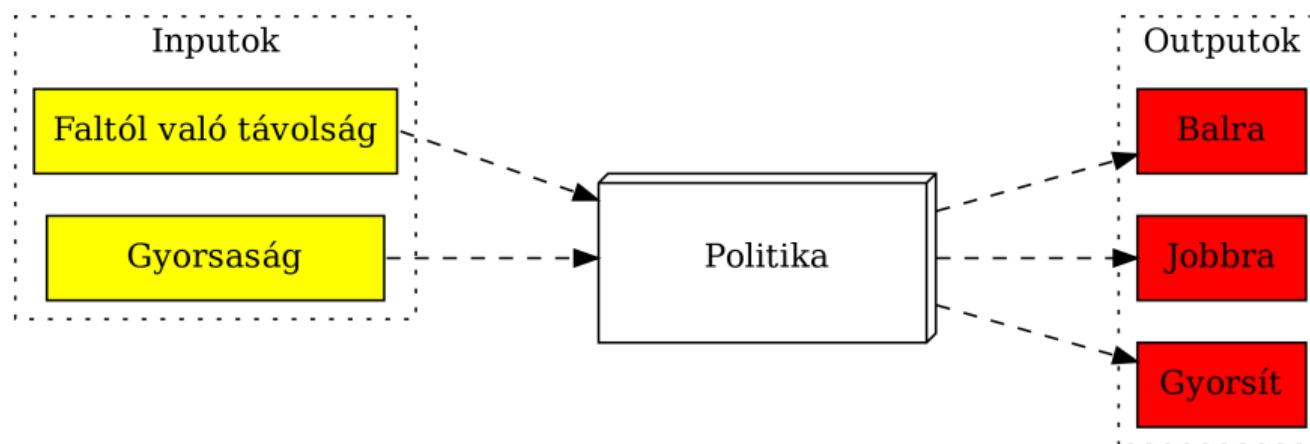
- A program, ami egy robotot irányít
- A program, ami PacMan-t irányítja
- Egy Go-t játszó program
- Lehetséges egy okos termostát is
- Kereskedő a tőzsden



Politika

Az az algoritmus, ami az ügynököt irányítja. A politika által határozza meg az ügynök a cselekvéseit.

A politika egy modell, ami a környezetet leíró változókat fogadja bemenetként, és az outputja egy cselekvés a cselekvések választható halmazából.



Interakció a környezettel

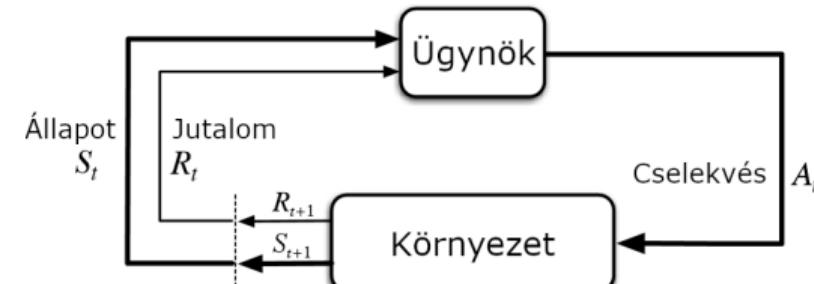
- Az ügynök és a környezet egymásra hatnak. Az ügynök cselekszik, ennek hatására a környezet megváltozik. Az ügynök megfigyeli a környezetet, majd ismét cselekszik:

$$s_1 \rightarrow a_1 \rightarrow s_2 \rightarrow a_2 \rightarrow \dots \rightarrow s_t \rightarrow a_t$$

- A jutalom azonnali, és cselekvés-állapot párosért jár: $R(s, a)$
- A környezet változását az átmeneti valószínűségek adják: $P(s'|s, a)$, ami s' következő állapot valószínűsége s állapotból, a cselekvést követően. Ez a környezet dinamikája.

- Az ügynök célja a lehető legmagasabb jutalom összegyűjtése hosszú távon:

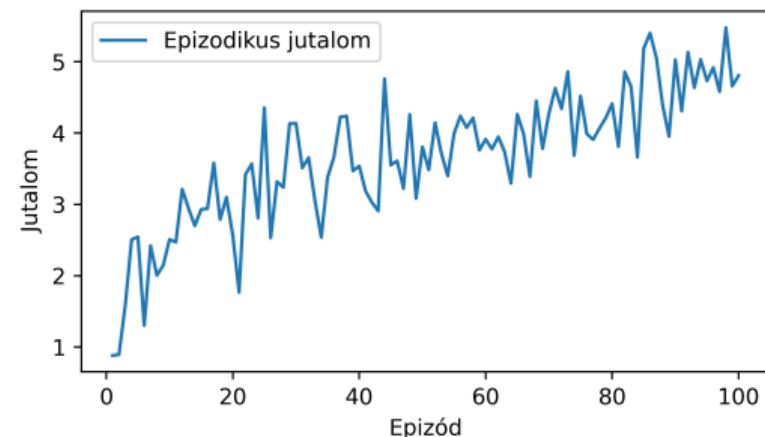
$$E_{\pi}(r_1 + r_2 + r_3 + \dots) \rightarrow \max$$



Epizódok

A megerősítéses tanulás egyetlen tanítási iterációja egy epizód. **Egy epizód addig tart, míg az ügynök el nem ér valamilyen vég/terminális állapotba.** A végállapot lehet:

- A cél teljesítése
- Teljes bukás / halál
- Időkeret lejárása
- Részfeladat teljesítése
- Jutalom összeg összegyűjtése

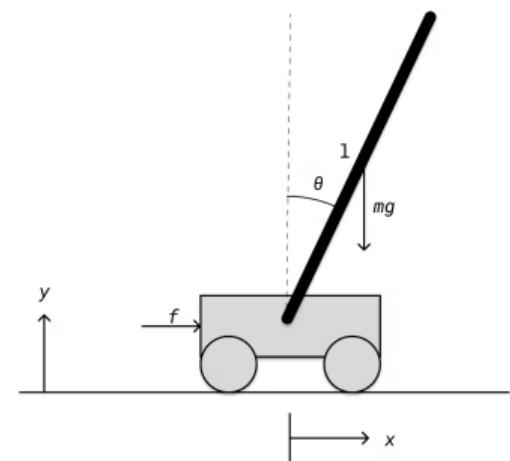


1 Bevezetés

2 Politika javítása

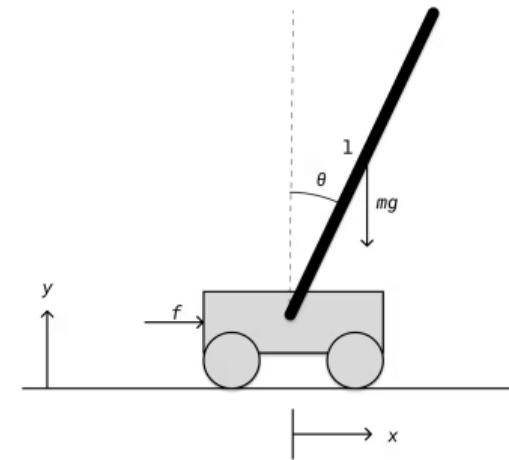
Egy példa környezet: CartPole

A CartPole egy egyszerű, de kihívást jelentő probléma. Egy oszlop egy szekérre van helyezve, és az ügynök célja, hogy a szekeret mozgatva az oszlopot egyensúlyban tartsa.



Egy példa környezet: CartPole

- **Állapotok:** Az állapot négy valós szám, amelyek a székér pozícióját, sebességét, az oszlop szögét és szögsebességét írják le.
- **Cselekvések:** Két lehetséges cselekvés van: a székér mozgatása balra vagy jobbra.
- **Jutalmak:** minden lépésért, amely során az oszlop nem esik le, a rendszer egy pontot ad. A cél az, hogy minél tovább fenn tartsuk az oszlopot, ezzel maximalizálva a kumulatív jutalmat.
- **Epizód:** Az epizód akkor ér véget, ha az oszlop egy bizonyos szögnél jobban elhajlik, vagy ha a székér kimegy a meghatározott határokon kívülre.



Egy egyszerű kezdeti politika

Az első egy keménykódolt politika: **olyan statikus szabályrendszer, amelyet nem gépi tanulás segítségével ismer meg az ügynök, hanem egy determinisztikus program vezérlí:**

```
for episode in range(n_episodes):
    for step in range(n_steps):
        if obs.theta < 0:
            action = 0
        else:
            action = 1
        obs, reward = env.step(action)
```

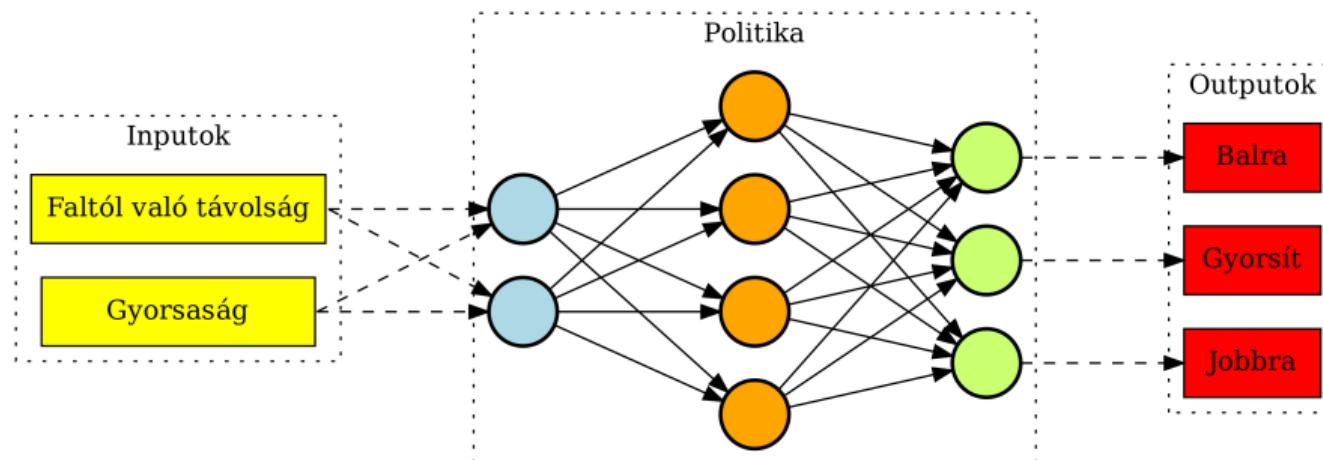
A politika szerint ha a θ oszlop dőlési szöge kevesebb, mint 0, a kocsit balra tolja, egyébként jobbra.

Az `env.step(action)` függvény segítségével hajtja végre az ügynök a választott cselekvését, majd a környezet visszaadja neki a jutalmat és a következő állapotot.

Neurális hálózat politika

A szabályok kézzel való implementálása hosszas és túlságosan specifikus. Ettől egy jobb hozzáállás, ha egy gépi tanulás modell becsüli a cselekvéseket a környezeti változók alapján.

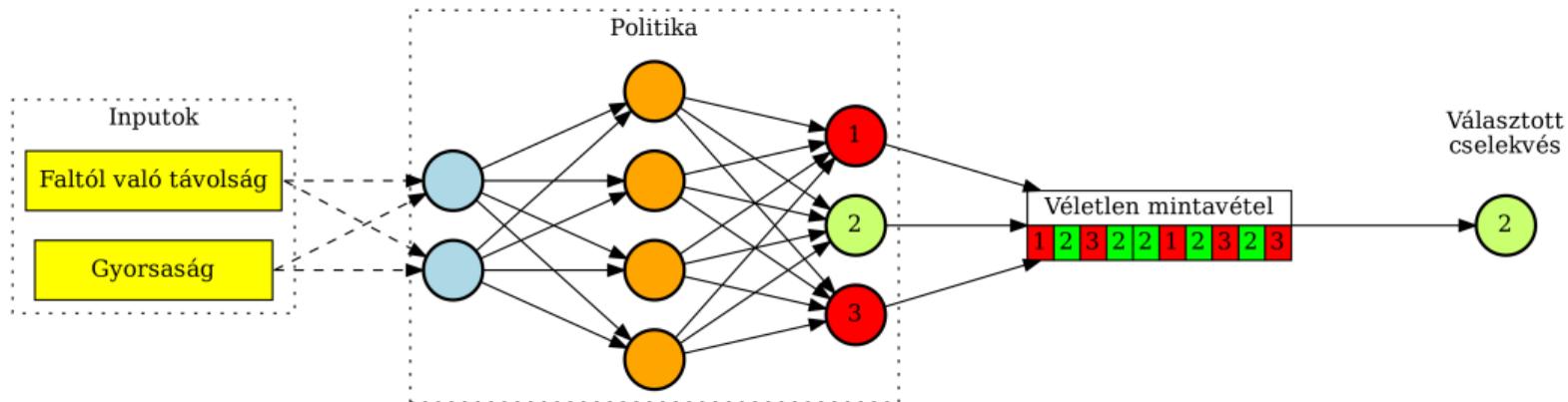
A modell bemenete ebben az esetben a környezeti változók vektora, a kimenete pedig a cselekvés, amit az ügynök végre fog hajtani. A tanítás eljárása pedig a neurális hálózat javításaként értelmezhető.



Politika hálózat működése

A hálózat output neuronjai azt a valószínűségetbecsülik meg, hogy mekkora valószínűsséggel az adott cselekvés lesz a leginkább jövedelmező az ügynök számára.

Ezután történik egy véletlen mintavétel, ahol ε valószínűsséggel a legjobb cselekvés fog szerepelni, $1 - \varepsilon$ valószínűsséggel pedig véletlen cselekvés. Ezzel lesz képes az ügynök felfedezni véletlen cselekvéseket a nem várt, de magas jutalom reményében.

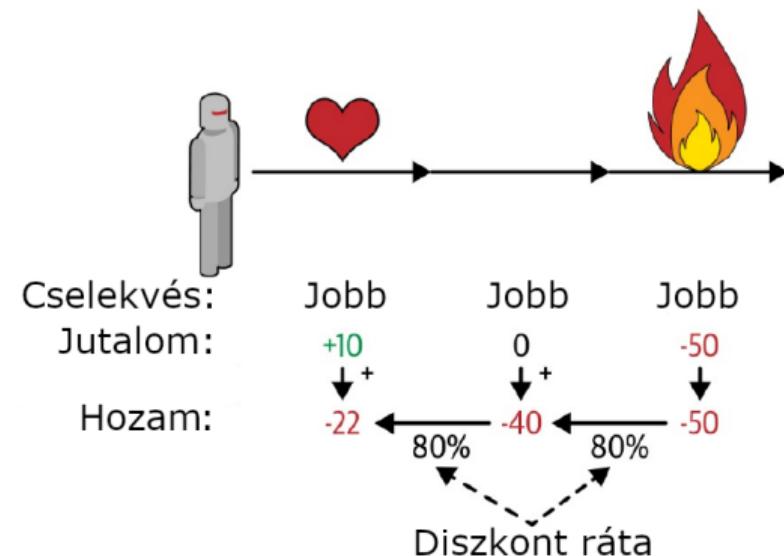


A kredit hozzárendelési probléma

Ha lehetséges lenne tudni minden lépésben, hogy melyik az optimális cselekvés a neurális hálózat egyszerűen tanítható lenne a keresztrópia minimalizálásával.

Viszont az RL-ben a visszajelzések ritkák és késleltetettek. Az ügynök egyetlen visszajelzése amit kap, a jutalom, és nem mindig az utolsó cselekvés az, ami felelős a jutalomért.

Például: ha az ügynök 100 lépésen keresztül egyensúlyozza a rudat, majd leejt, honnan lehet tudni melyik cselekvés volt a felelős a leejtésért?



A kredit hozzárendelési probléma

A probléma megoldására az RL bevezet egy γ diszkontálási tényezőt, amely **megadja a jövőbeli jutalmak jelenbeli értékét**. Valamely r jutalom értéke k időlépés után γ^{k-1} .

Példa: ha az ágens háromszor jobbra megy, ezután a három jutalma $[+10, 0, -50]$, és $\gamma = 0.8$ az első cselekvés visszatérése $10 + 0 \cdot \gamma + \gamma^2 - 50 = -22$.

Ha $\gamma = 0$, a modell a jelenbeli jutalmat részesíti előnyben. Ha viszont 1 közelí az értéke, a hosszú távú jutalomra törekzik.

