Fakultet elektrotehnike i računarstva

Automatizirani razvoj agenata za različita okruženja

Diplomski rad

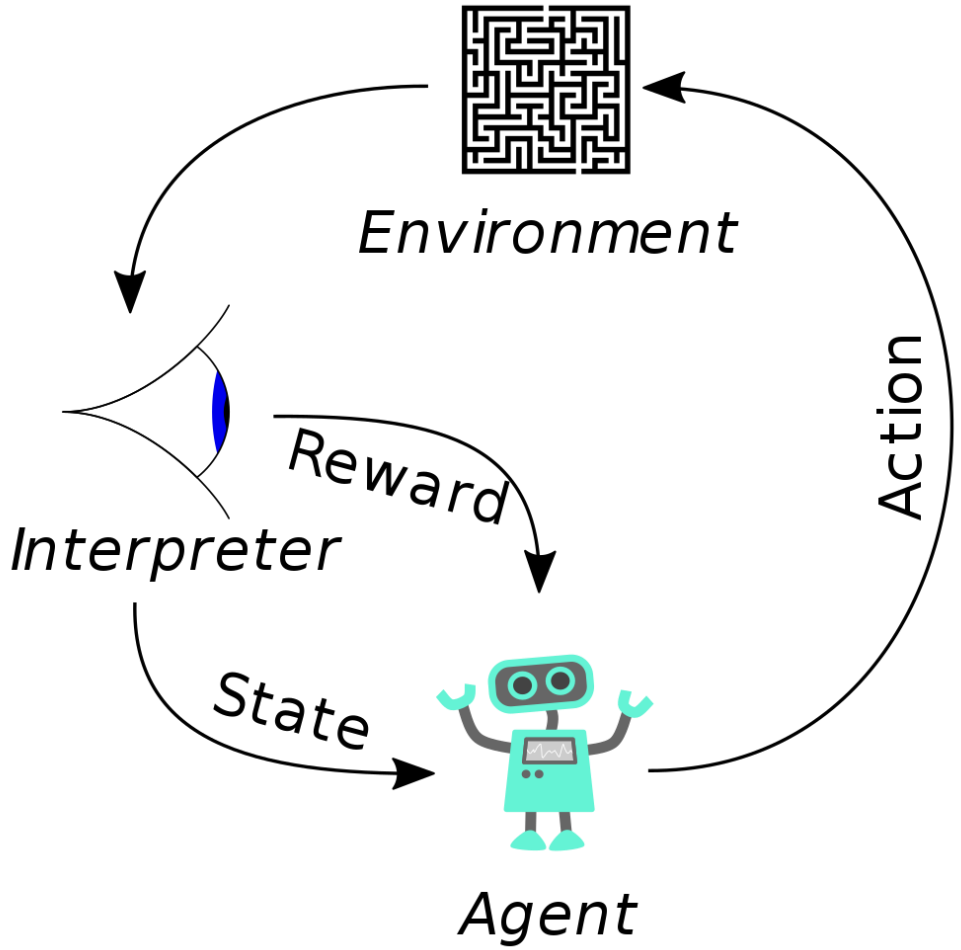
Paulo Sanković

Mentor: *doc. dr. sc.* Marko Đurasević

Zagreb, rujan 2022.

PODRŽANO UČENJE

- Bavi se optimizacijom ponašanja agenta koji u interakciji s okolinom izvršava akcije na temelju povratnih informacija okoline
- Povratna informacija okoline u obliku nagrade / kazne



POJMOVI

Politika

Putanja

Povrat

Funkcija politike

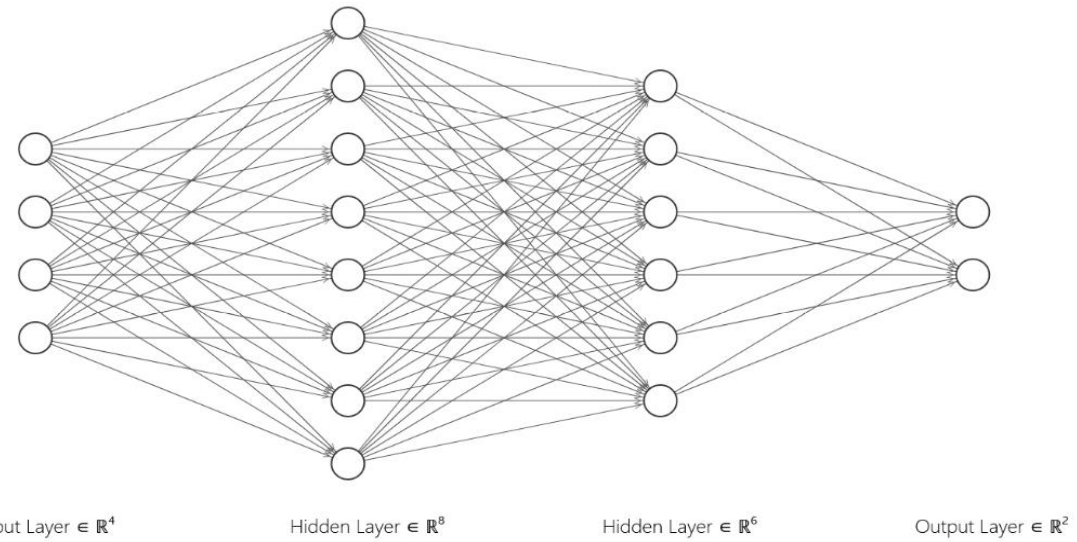
Funkcija vrijednosti stanja

Funkcija vrijednosti akcije

DUBOKI MODELI

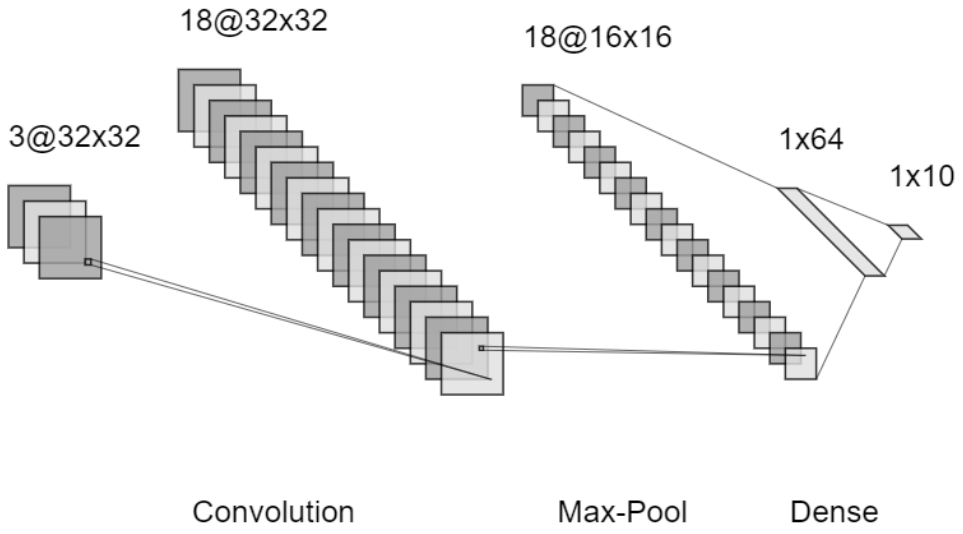
Unaprijedni potpuno povezani modeli

- Lanci potpuno povezanih slojeva (ulazni izlazni i skriveni)
- Svaki do slojeva modelira jednu nelinearnu transformaciju

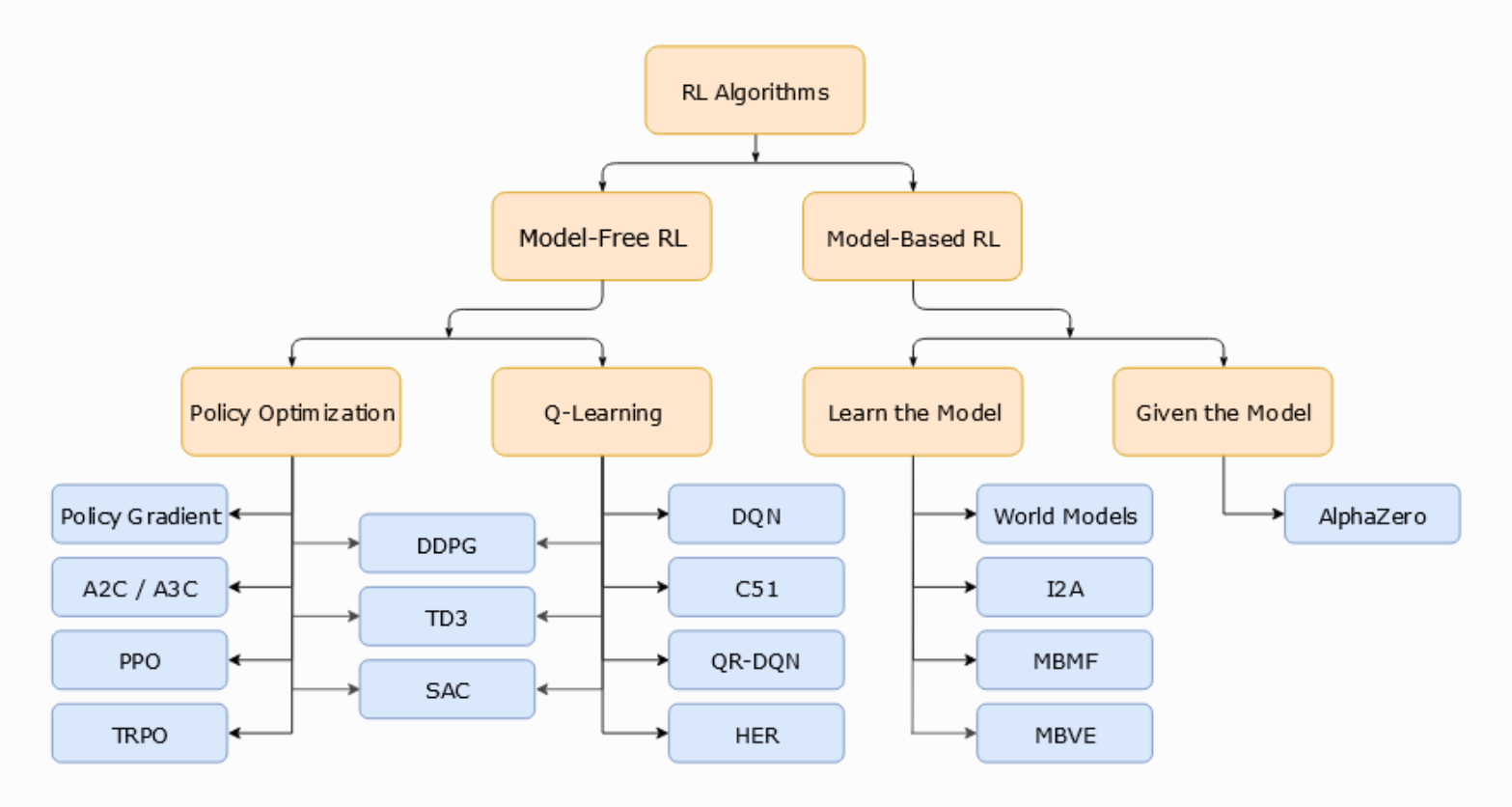


Konvolucijski modeli

- Uz potpuno povezane slojeve imaju najmanje 1 konvolucijski sloj
- Operacije konvolucije, sažimanja (maksimumom i srednjom vrijednosti)



ALGORITMI PODRŽANOG UČENJA

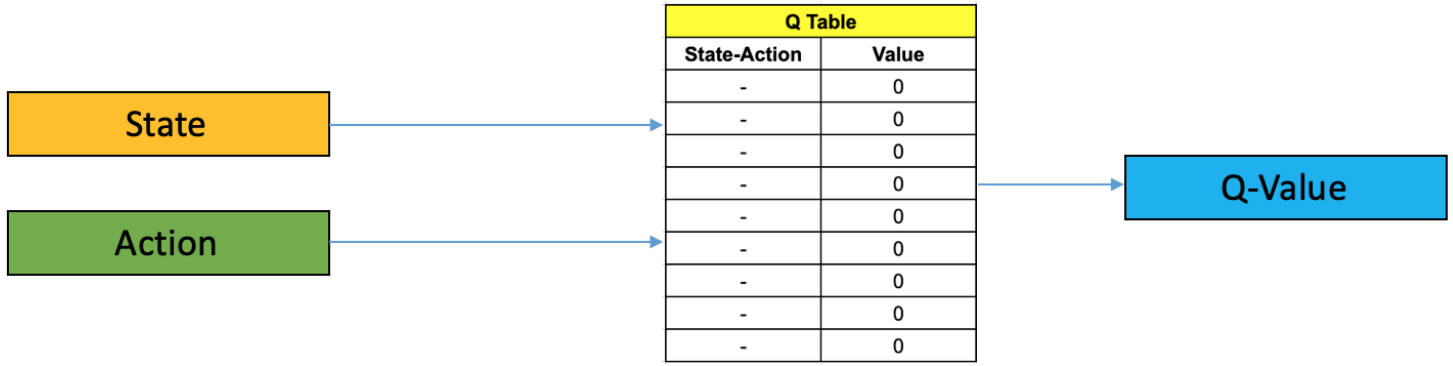


PODJELA

- *Model-based methods*
- *Model-free methods*
- Učenje s isključenom politikom
- Učenje s uključenom politikom

Q UČENJE

- Algoritam s isključenom politikom
- Procjena kvalitete određene akcije
- Korištenje Q tablica
- Kompromis između istraživanja i odabira akcije s najvećom Q vrijednošću

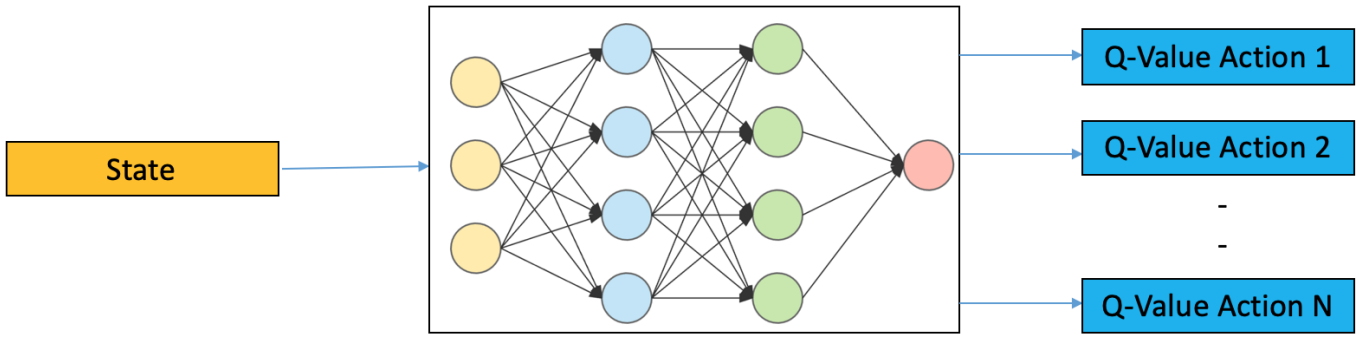


Q Learning

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha \left[r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right]$$

DUBOKO Q UČENJE

- Korištenje Q tablica nije pogodno za kompleksne okoline, okoline s velikim brojem stanja i akcija
- Ideja:
  - Naučiti duboki model da za određeno stanje okoline na ulazu, generira skup svih akcija i propadajućih vrijednosti funkcije akcije
  - Odabrati akciju s najvećom Q vrijednošću
- 2 neuronske mreže
  1. Prati trenutnu politiku
  2. Prati ciljanu politiku
- Korištenje spremnika za ponavljanje



Deep Q Learning

DVOSTRUKO DUBOKO Q UČENJE

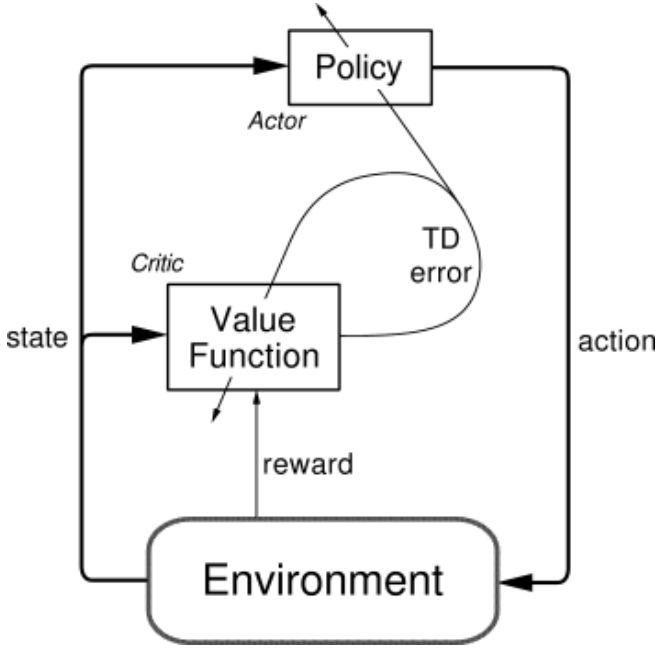
- Cilj smanjiti efekt precjenjivanja vrijednosti funkcije akcije bez mijenjanja osnovnog kostura algoritma

$$\max_a Q(s_{t+1}, a; \theta^-) = Q(s_{t+1}, \operatorname*{argmax}_a Q(s_{t+1}, a, \theta^-); \theta^-)$$

- Postupak odabira akcije provodi neuronska mreža koja slijedi politiku
- Izračun funkcije akcije provodi ciljana neuronska mreža

AKTER-KRITIČAR

- Aproksimacija funkcije politike i funkcije vrijednosti
- Akter – uči parametriziranu politiku
- Kritičar – procjenjuje funkciju vrijednosti
- Algoritam s uključenom politikom



PREDNOSNI AKTER-KRITIČAR

- Procjena trenutne akcije korištenjem funkcije prednosti
- Funkcija prednosti - koliko je određena akcija bolja od one koja prati politiku

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$$

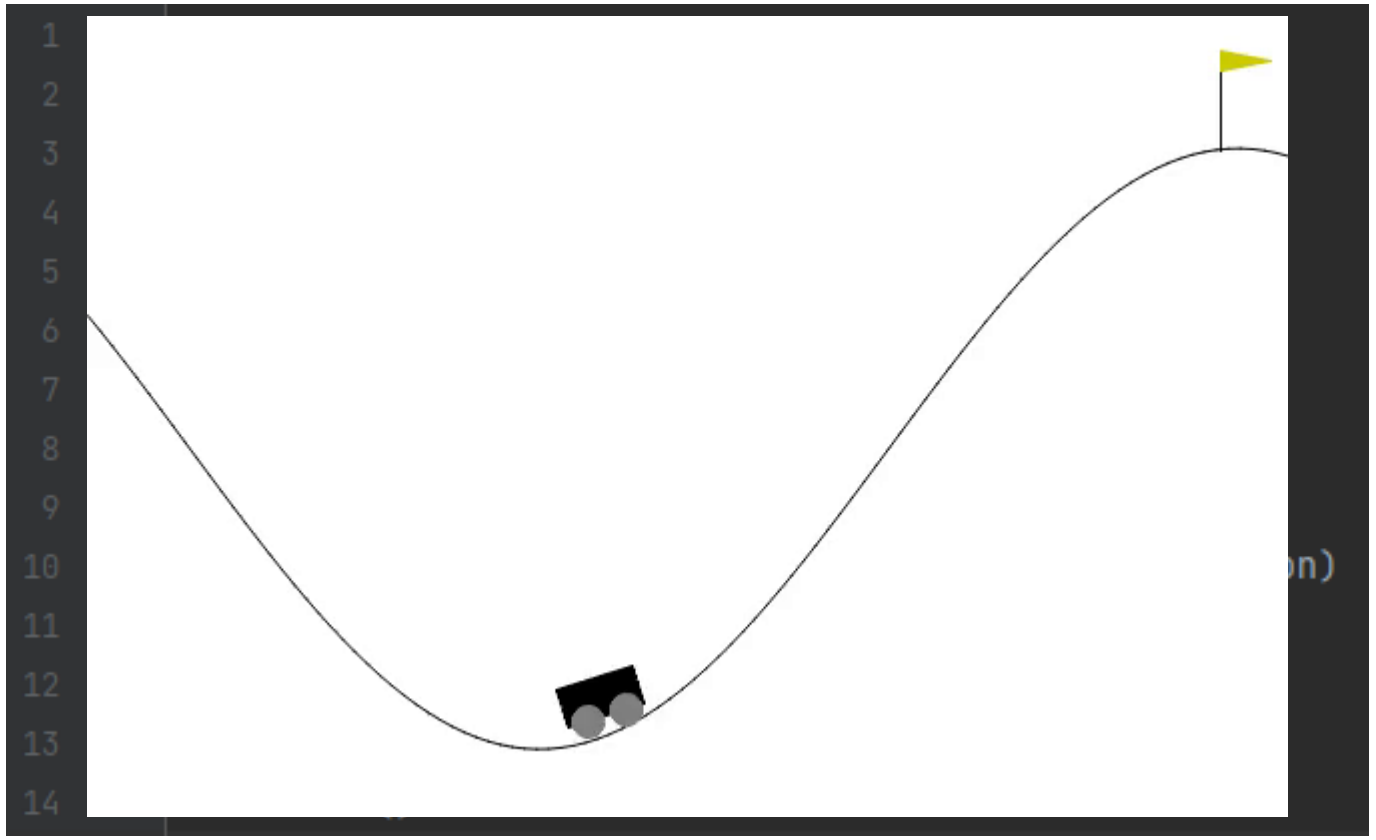

OpenAI Gym

OpenAI Gym

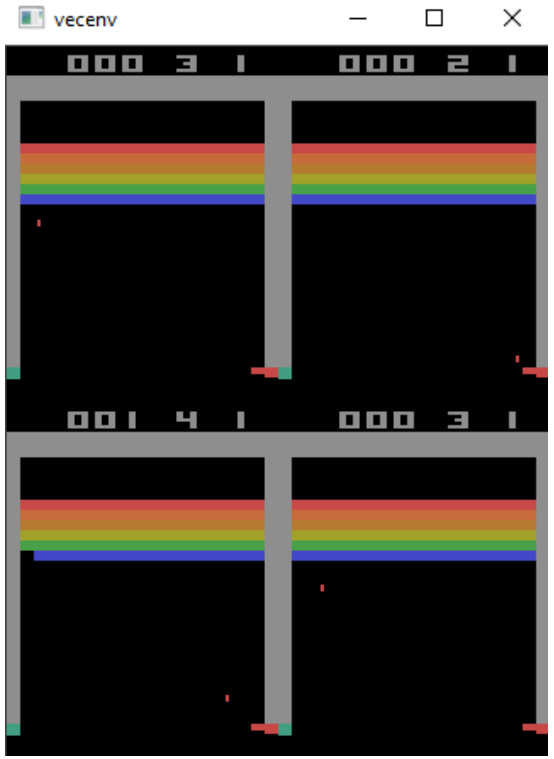
- *Python* biblioteka otvorenog koda
- Razvijanje i usporedba agenata u odabranim okolinama

- Početno stanje se nasumično generira iz distribucije
- Interakcija do terminalnog stanja

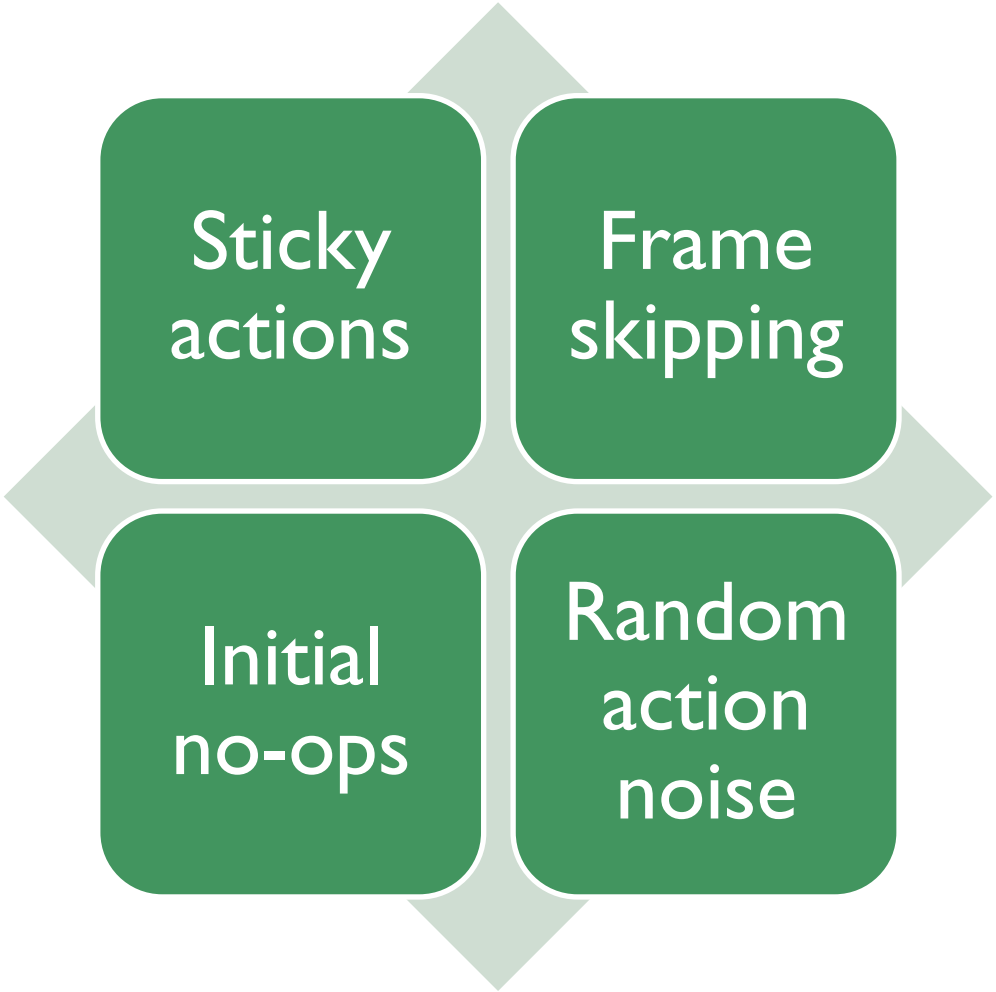
Run program



STRUKTURA



- Okolina
- Interakcija
- Prostor akcija
- Prostor stanja
- Omotači



DETERMINIZAM ATARI OKRUŽENJA

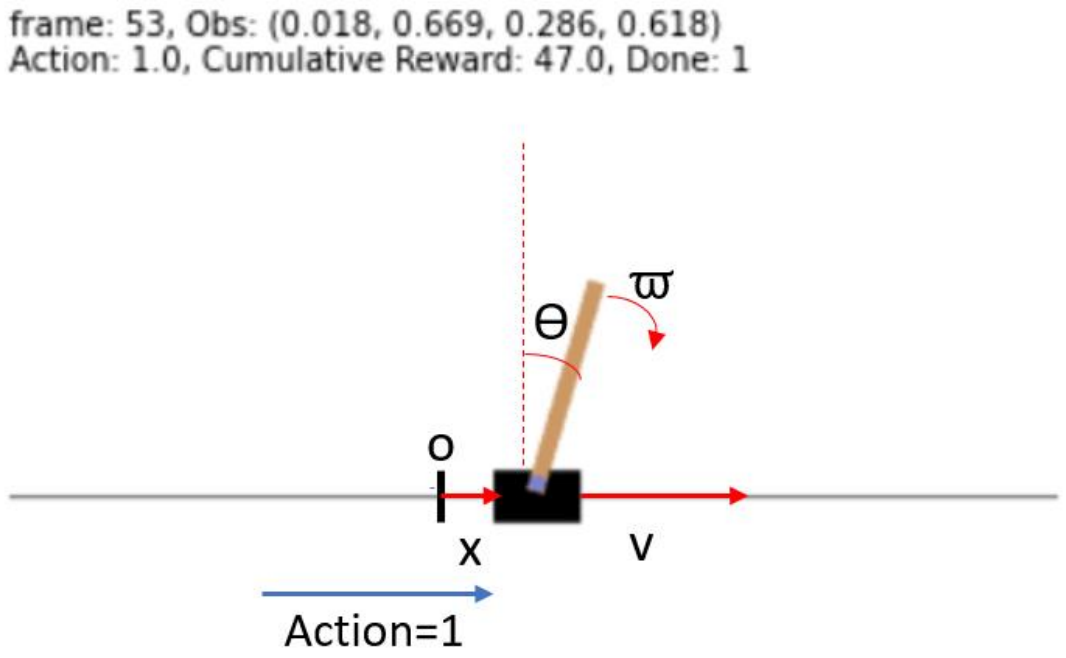
- Originalna Atari 2600 konzola nije imala izvor entropije za generiranje pseudoslučajnih brojeva
- Okruženja u potpunosti deterministička

Akcija	Opis akcije
0	Pomak kolica ulijevo
1	Pomak kolica udesno

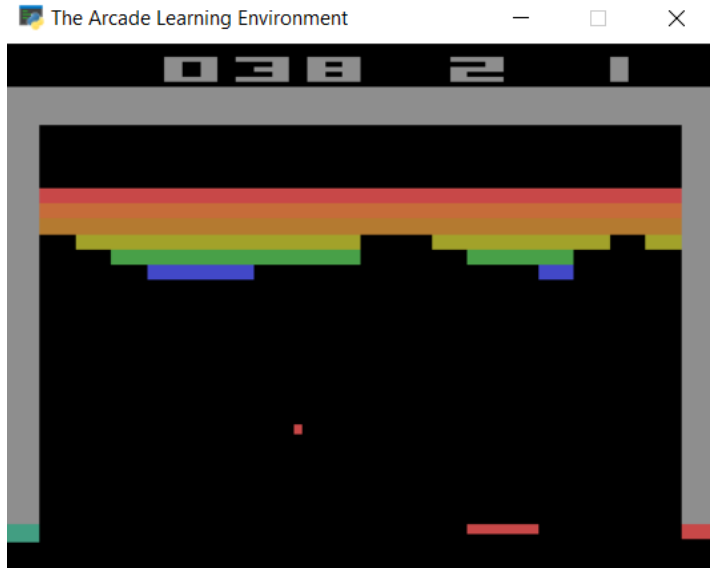
Indeks	Opis	Donja granica	Gornja granica
0	Pozicija kolica	-4.8	4.8
1	Brzina kolica	$-\infty$	∞
2	Nagib štapića i kolica	-0.418rad	0.418rad
3	Brzina štapića na vrhu	$-\infty$	∞

Okruženje CartPole

- Fizikalni problem održavanja ravnoteže
- Cilj održavati ravnotežu stupa primjenom sile i pomicanjem kolica u lijevom / desnom smjeru



Akcija	Opis akcije	Detaljniji opis akcije
0	NOOP	Ne poduzima se nikakva akcija
1	FIRE	Akcija koja pokreće igru
2	RIGHT	Platforma se pomiče udesno
3	LEFT	Platforma se pomiče ulijevo

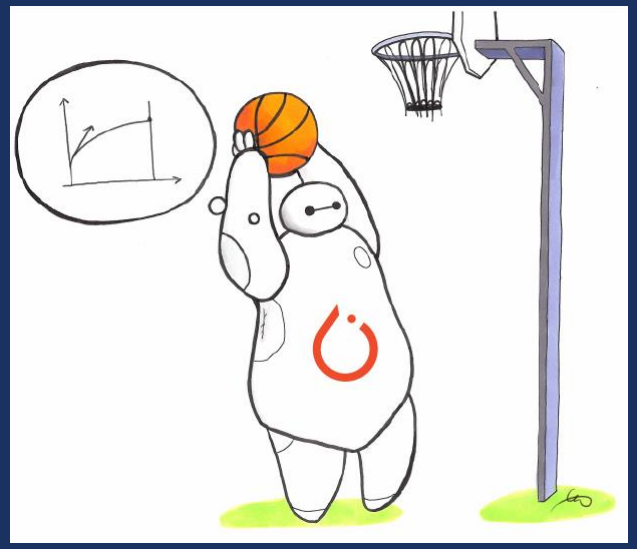


Okruženje Breakout

- Cilj sakupiti što više bodova pomičući platformu i održavajući lopticu na ekranu
- Igra završava kada igrač potroši 5 života

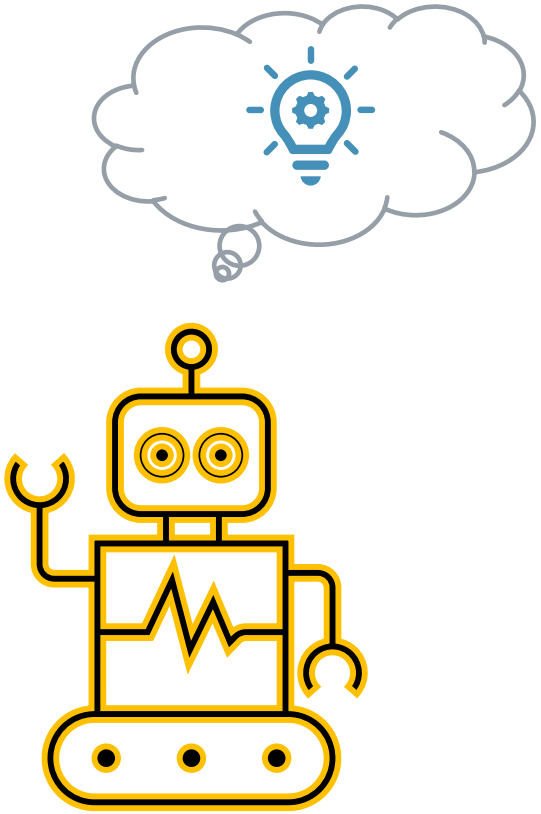
Indeks	Struktura
RAM vrijednosti	`Box(0, 255, (128,), uint8)`
Vrijednosti RGB slike	`Box(0, 255, (210, 160, 3), uint8)`

Stable Baselines3



- Skup implementacija algoritama podržanog učenja
- Korištenje *PyTorch* biblioteke
- Dobra dokumentacija, jednostavnost, 95% koda pokriveno testovima

```
1  from stable_baselines3 import A2C
2
3  env = A2C('MlpPolicy', 'CartPole-v1', verbose=1)
4  model = env.learn(total_timesteps=100_000)
```



IMPLEMENTACIJA

OMOTAČI

Skaliranje i pretvaranje slike iz RGB vrijednosti piksela u nijanse sive boje

Prebacivanje u PyTorch dimenzije

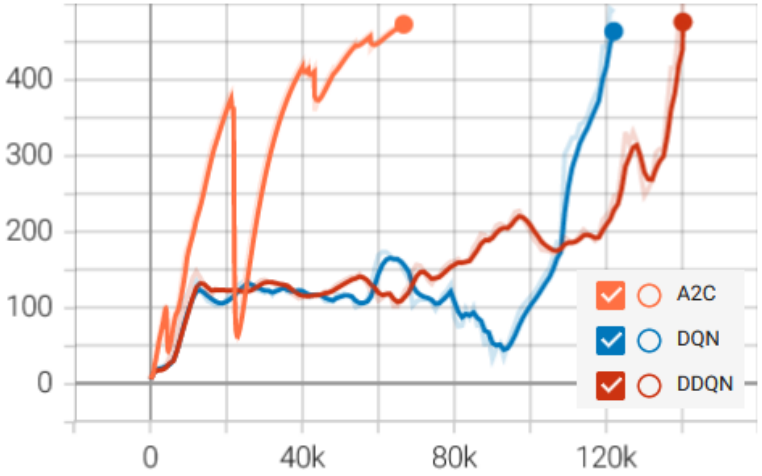
Tehnika *initial no-ops*

Podrezivanje vrijednosti nagrade

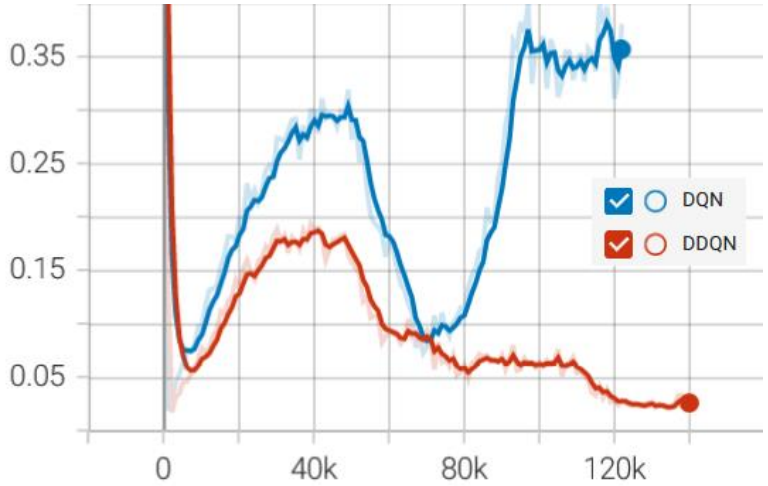
Izvršavanje operacije *FIRE* na početku epizode

Istovremeno izvođenje nekoliko instanci okruženja

REZULTATI AGENATA CARTPOLE OKOLINE



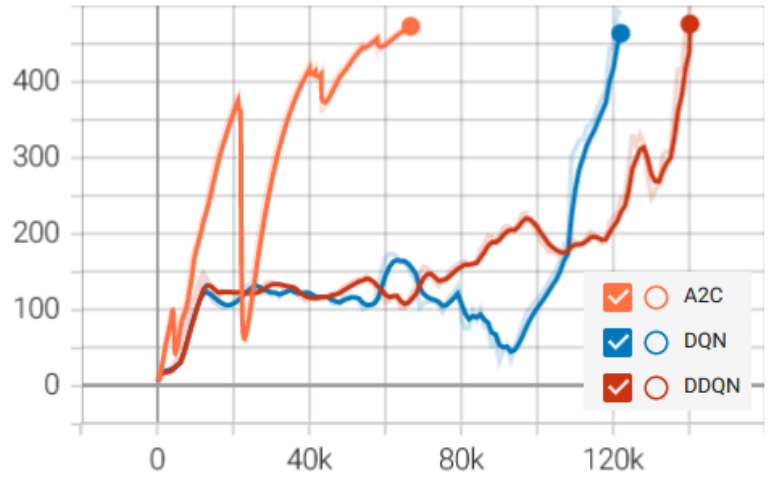
Kumulativna nagrada po broju iteracija



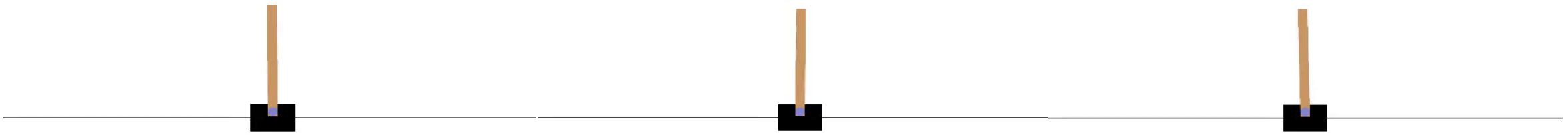
Prosječna vrijednost funkcije gubitka po broju iteracija

Algoritam	Iteracija	Kumulativna nagrada	Vrijeme učenja
DQN	122000	489.5	4m 50s
DDQN	140000	473	6m 46s
A2C	66870	476.2	48s

Statistike agenata pri najvećoj vrijednosti kumulativne nagrade



Kumulativna nagrada po broju iteracija

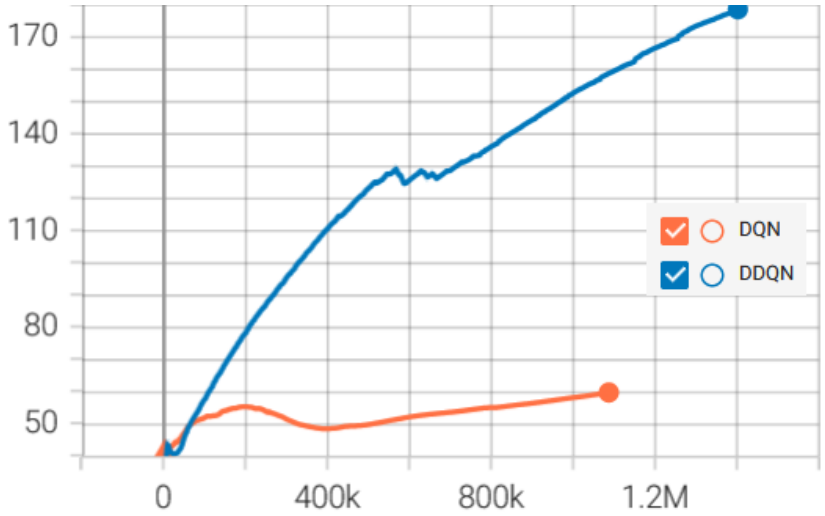


Duboko Q učenje

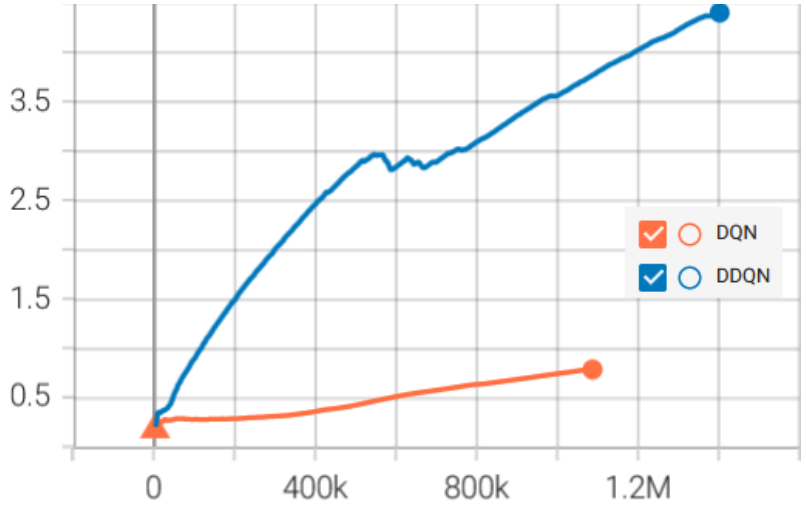
Dvostruko duboko Q učenje

Prednosni akter-kritičar

REZULTATI AGENATA BREAKOUT OKOLINE



Prosječna duljina života agenta po epizodi



Prosječna vrijednost skalirane nagrade po epizodi

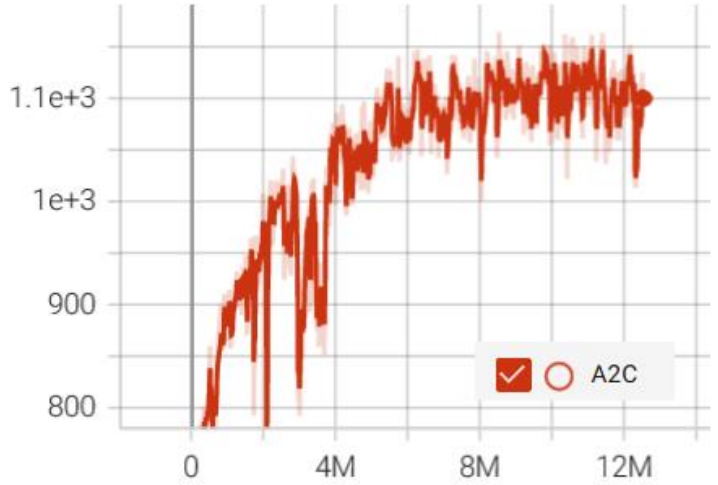
Algoritam	Iteracija	Kumulativna nagrada	Vrijeme učenja
DQN	1086000	59.42	12h 4m 20s
DDQN	1401000	178.8	10h 57m 24s

Statistike agenata pri najvećoj vrijednosti prosječne duljine života

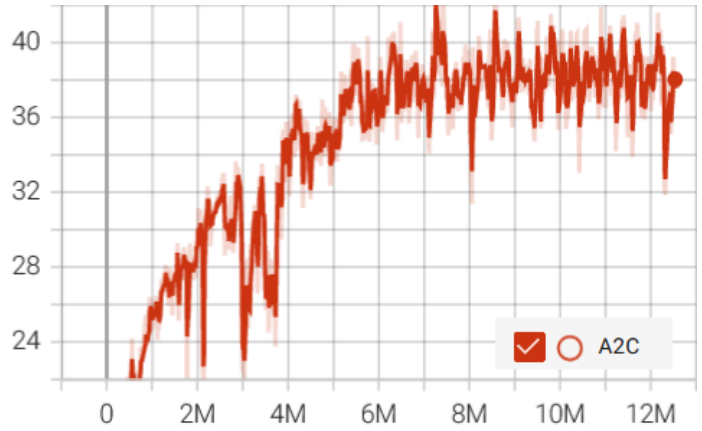
REZULTATI AGENATA BREAKOUT OKOLINE (*A2C*)

- Implementacija i učenje pomoću biblioteke *Stable Baselines3*

Prosječna vrijednost duljine epizode po epizodi

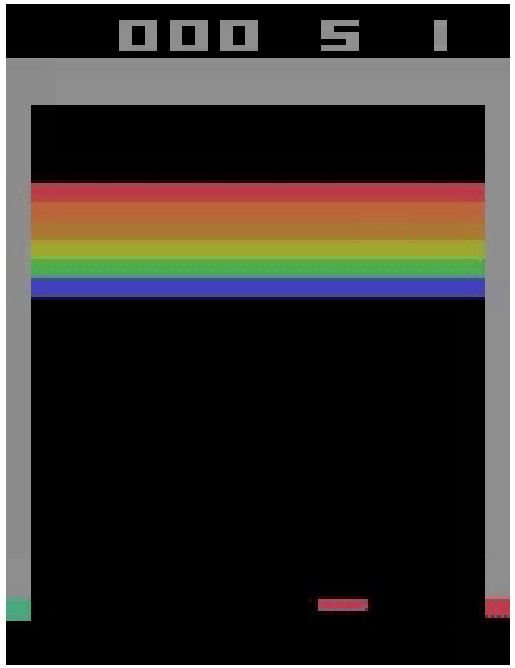


Prosječna vrijednost nagrade po epizodi

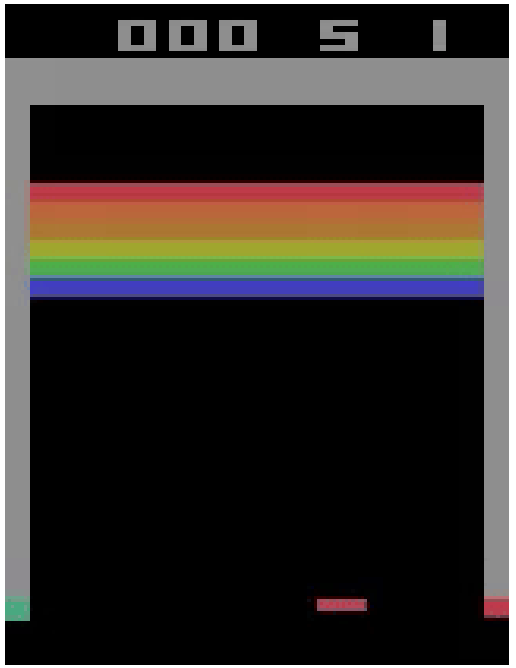


Algoritam	Iteracija po epizodi	Rezultat po epizodi
DQN	81.25 ± 13.311	5.87 ± 1.452
DDQN	385.5 ± 29.004	80.75 ± 8.584
A2C	277.57 ± 63.716	47.92 ± 16.024

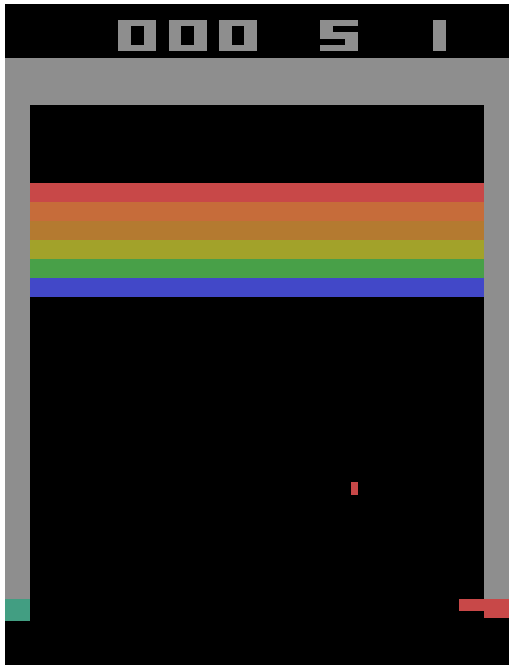
Evaluacija agenata Breakout okoline



Duboko Q učenje



Dvostruko duboko Q učenje



Prednosni akter-kritičar

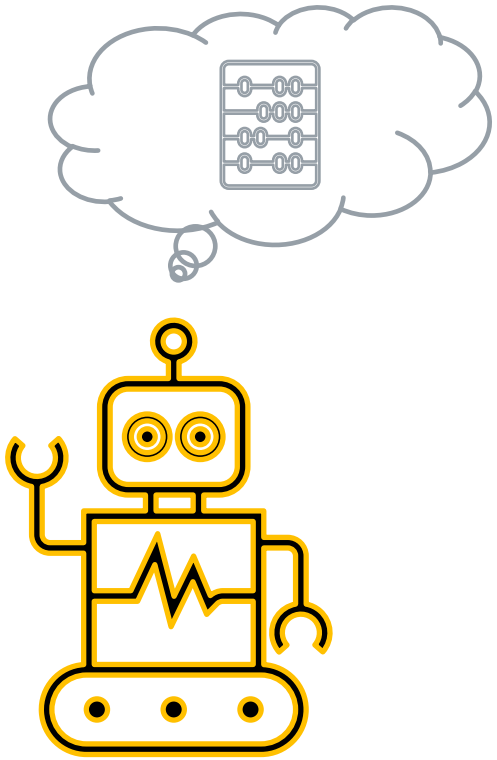
DALJNJI RAD

Implementacija dodatnih algoritama:

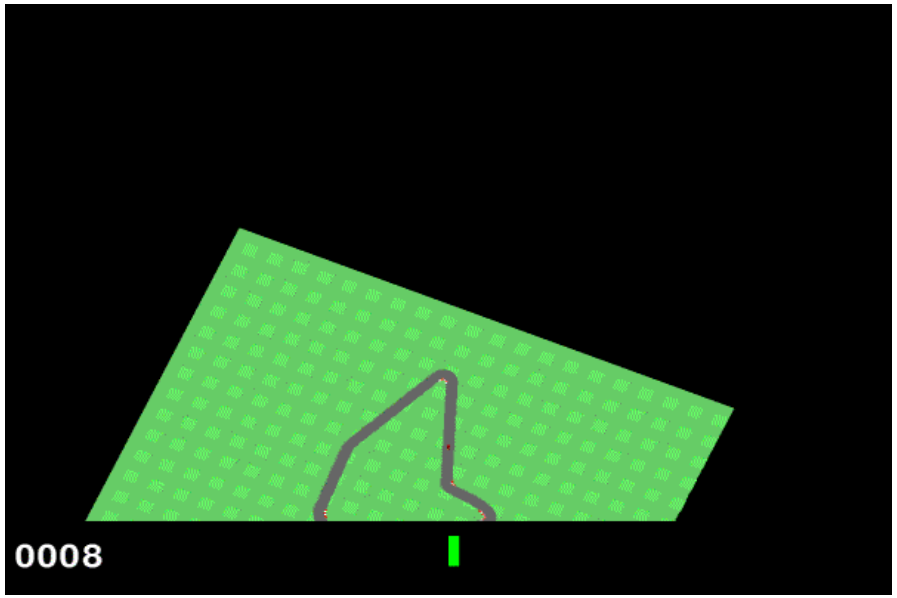
- Suparničko dvostruko Q učenje (*Dueling Double Deep Q Network*)
- Asinkroni prednosni akter-kritičar (*Asynchronous Advantage Actor-Critic*)
- Meki akter-kritičar (*Soft Actor-Critic*)
- *Trust Region Policy Optimization*
- *Proximal Policy Optimization*

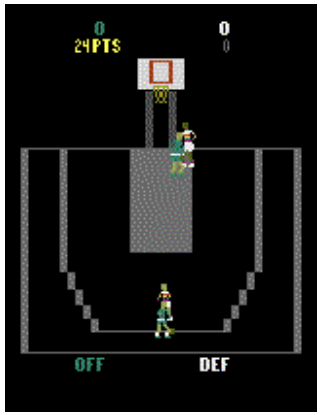
Pronalazak optimalnih hiperparametara i duže učenje modela

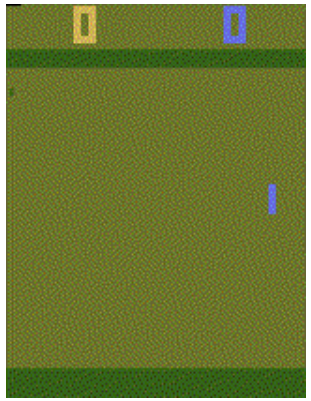
Validacijska epizoda



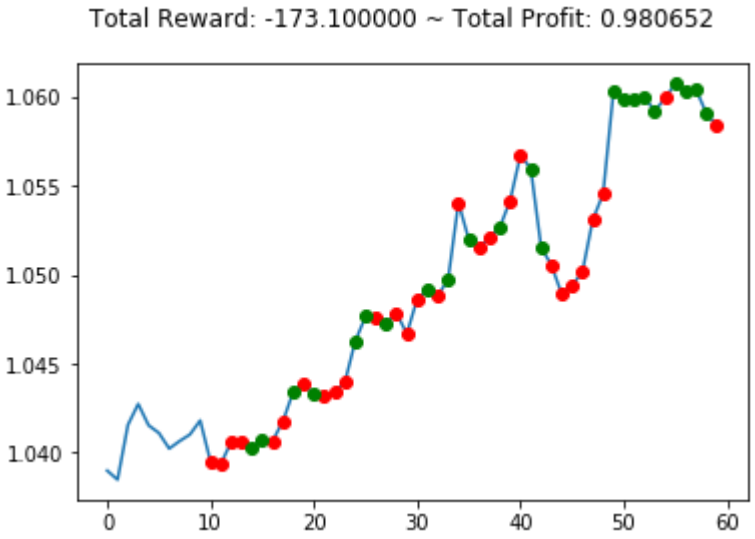
DALJNJI RAD – NOVE OKOLINE







- *Double Dunk, Space Invaders, Car Racing, Pong*
- *Gym AnyTrading, TensorTrade*



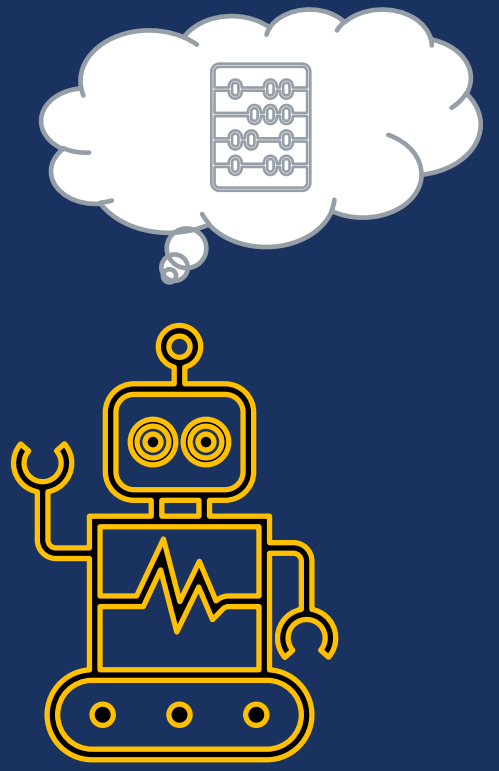
Snimke preuzete s https://www.gymlibrary.dev/

ZAKLJUČAK

Postignuti rezultati bliski onima *state-of-the-art* algoritama i implementacija

Inspiracija u načinima na koje ljudi i ostala živa bića uče iz iskustva pri interakciji s okolinom

Veliki broj algoritama, inačica, implementacija i dodataka čine podržano učenje izrazito velikim i iscrpnim područjem s velikim brojem primjena u stvarnom životu



HVALA NA PAŽNJI