

Reinforcement learning za upravljanje 2D vozilom

Raspoznavanje uzoraka i strojno učenje



FERIT

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Luka Šimić

Pregled

Uvod u reinforcement learning

Implementacija okoline

Implementacija koristeći biblioteku tf-Agents

Pregled rezultata

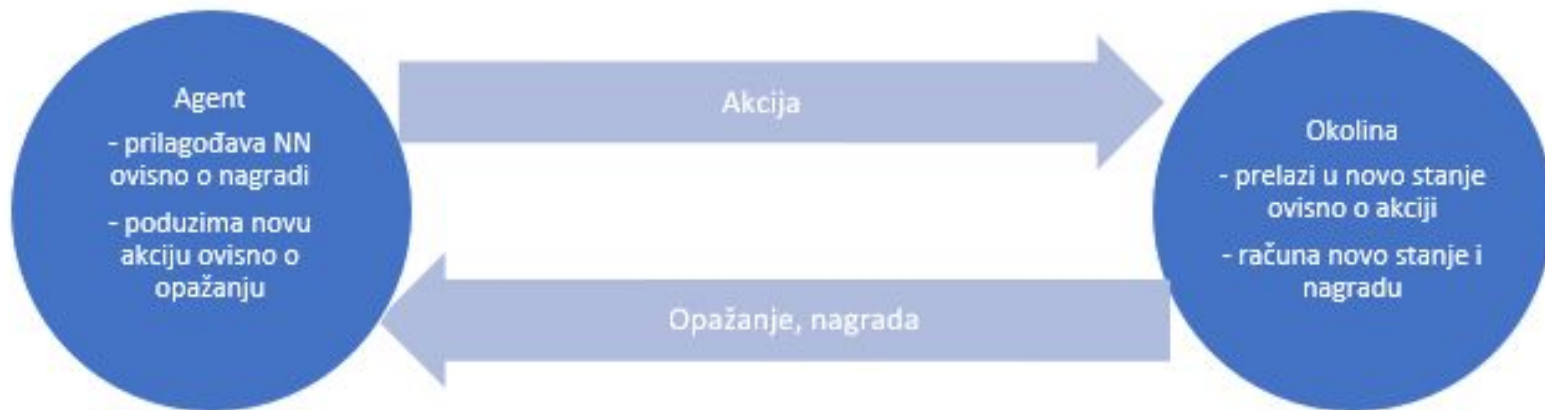
Reinforcement learning

Agent dobiva opažanja (engl. Observation) iz okoline (engl. Environment)

Na osnovu opažanja agent donosi odluku (engl. Action)

Akcije agenta donose određenu nagradu (engl. Reward)

Cilj je donositi odluke koje daju veću nagradu



Agent

Najčešće se temelji na neuronskim mrežama

Neki od algoritama su:

- Brute Force
- Value Function
- Q learning
- Monte Carlo metode

Okolina

Načelno, svaka okolina (model) sadrži step() metodu

step()

- prima akciju koju agent poduzima
- računa novo stanje okoline
- provjerava uvjet završenosti
- računa nagradu
- kao rezultat vraća nagradu i opažanje

Funkcija nagrade (engl. Reward Function)

Računa nagradu ovisno o promjeni stanja okoline

Govori koliko je neka akcija dobra (ili loša)

Nagrada se može dodjeljivati na svakom koraku, ili kada je okolina u prihvatljivom stanju

Sparse reward setting

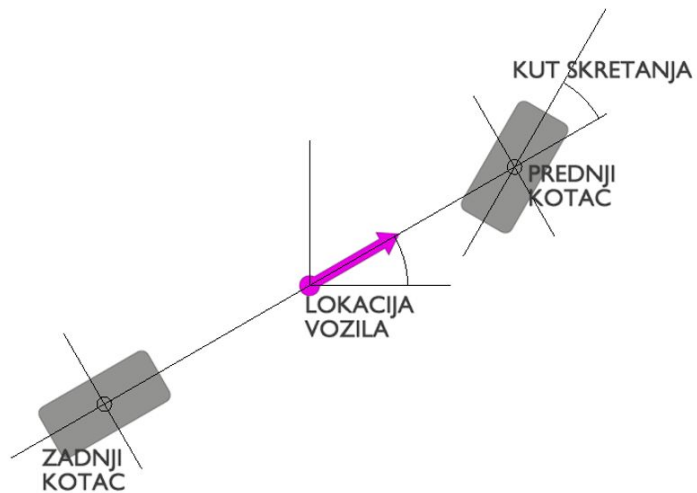
Situacija kada vraćamo nagradu u samo jednom (ili malom broju) koraka

Agent mora poduzeti velik broj ispravnih akcija kako bi došao do nagrade

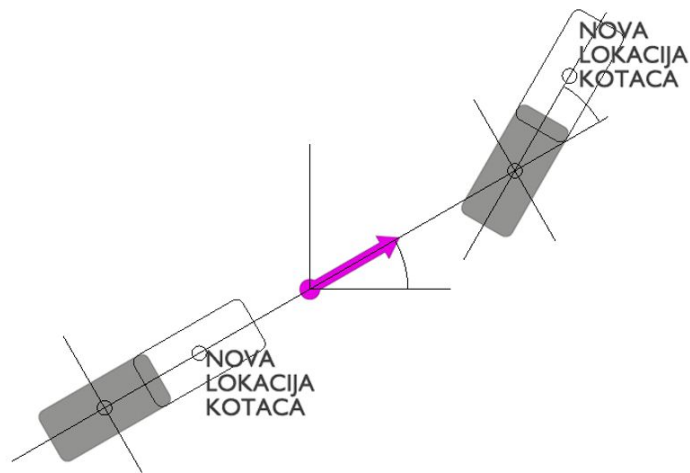
Velika vjerojatnost da agent nikada ne dođe u prihvatljivo stanje

Rješenje jako sporo (ili nikada) konvergira

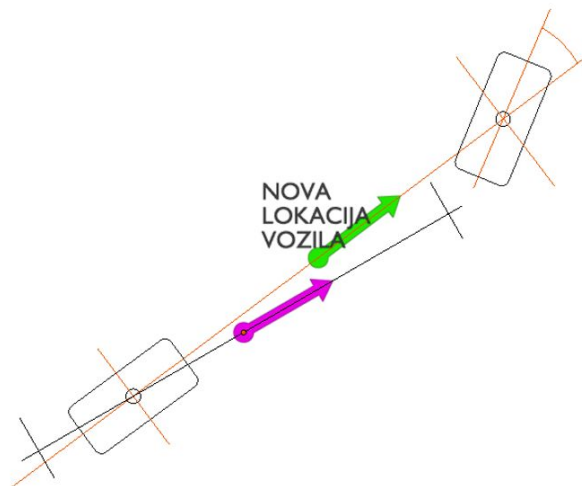
Model okoline



Model okoline



Model okoline



Implementacija - Okolina

Opažanja (engl. Observation) - vektor od 7 vrijednosti, normaliziran u $[-1, 1]$

- X, Y koordinate cilja
- X, Y koordinate vozila
- X, Y vektor kretanja vozila
- Kut između cilja i vektora kretanja

Akcije - Diskretne akcije

- 0 - vozilo ne skreće
- 1 - vozilo skreće lijevo
- 2 - vozilo skreće desno

Implementacija - Funkcija nagrade

$$\frac{\text{početni kut} - \text{krajnji kut}}{\text{max. kut skretanja}}$$

- za svaki korak

$$25 + 75 \cdot \left(1 - \frac{\text{tren. korak}}{\text{max korak}}\right)$$

- kada vozilo dođe do cilja

$$- 100$$

- ako vozilo izađe izvan definiranih granica

$$25 \cdot \left(1 - \frac{\text{trenutna udaljenost}}{\text{početna udaljenost}}\right)$$

- na posljednjem koraku

$$2.5$$

- ako se vozilo stvori unutar cilja

Treniranje

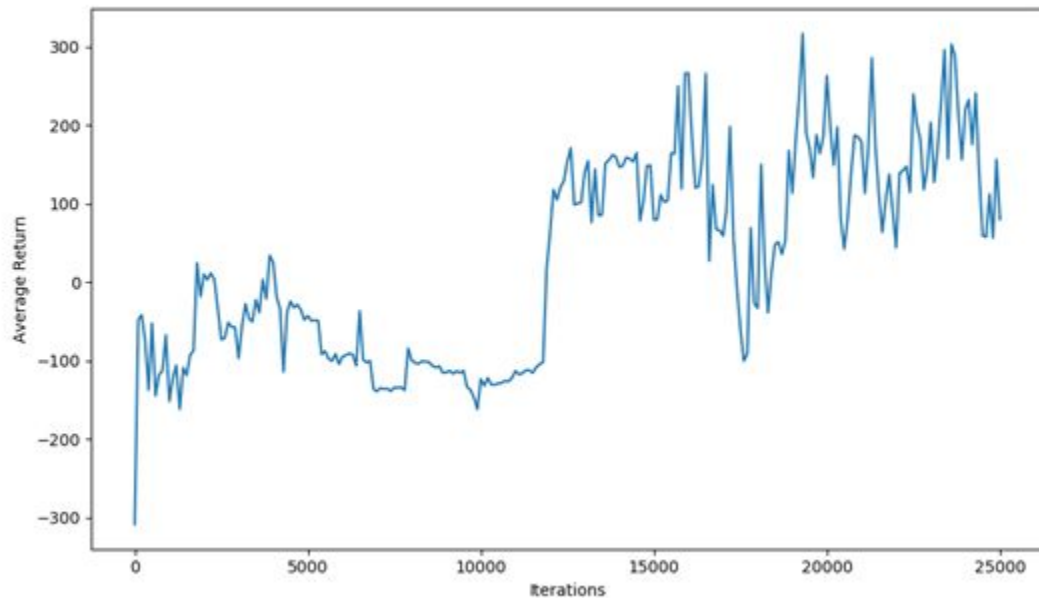
Koristeći TF-Agents biblioteku, Python

TF-Agents - pruža implementaciju uobičajenih algoritama za machine learning

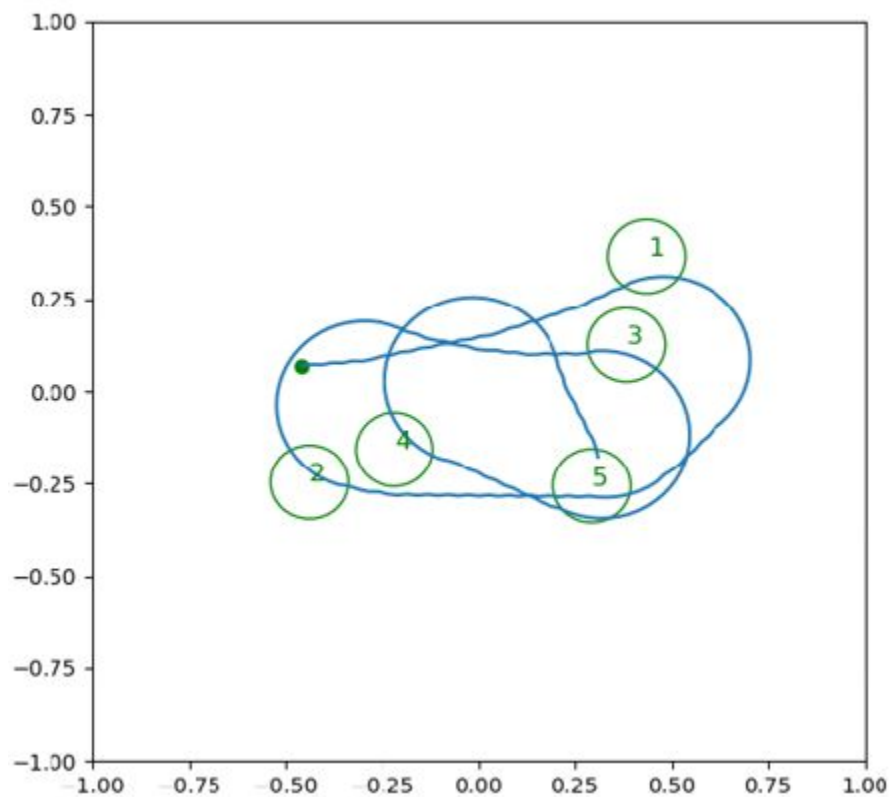
Koristi se mreža sa 2 Fully Connected sloja

Svaki sloj sadrži 32 neurona

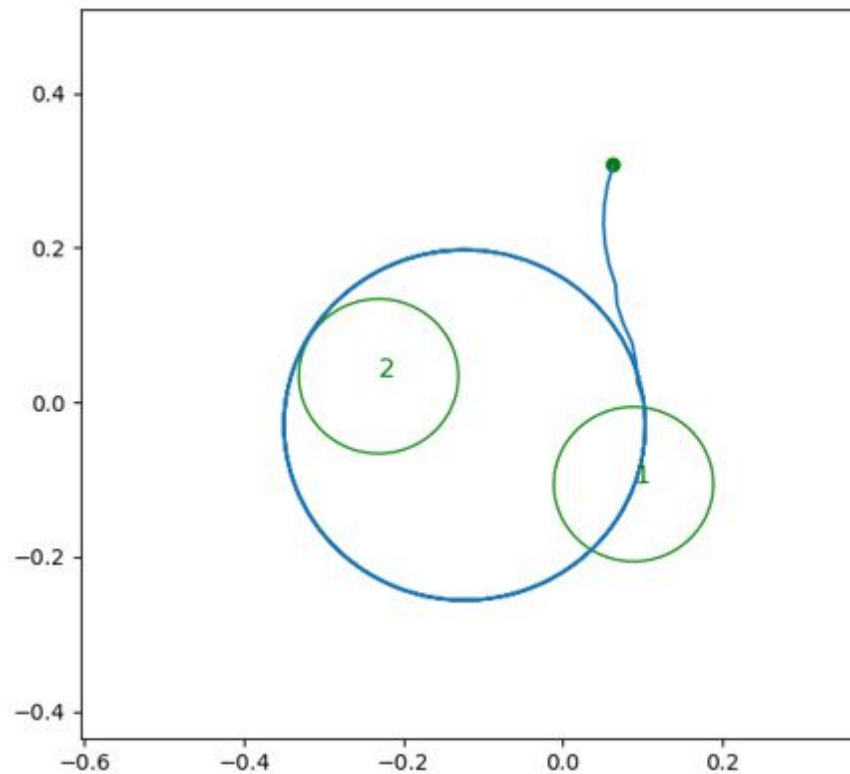
Treniranje



Rezultati



Rezultati



Rezultati

Trenirano na:	Broj ciljeva	1 cilj	2 cilja	5 ciljeva	10 ciljeva
5k iteracija	uspjeh	100/100	37/100	0/100	0/100
	pr. vrijeme	3.39s	4.66s	N/A	N/A
	st. dev.	2.1021	1.8621	N/A	N/A
15k iteracija	uspjeh	100/100	55/100	16/100	0/100
	pr. vrijeme	5.44s	9.46s	18.54s	N/A
	st. dev.	3.6152	4.0826	4.2720	N/A
25k iteracija	uspjeh	100/100	91/100	79/100	59/100
	pr. vrijeme	3.10s	5.37s	12.83s	25.95s
	st. dev.	2.2897	2.1890	3.0395	4.9754

Zaključak

Prednosti

- Tretiramo okolinu kao “*Black Box*”
- Pronalazi rješenja do kojih nije moguće doći analitički

Nedostatci

- Oblikovanje funkcije nagrade je teško
- Rubni slučajevi

Pitanja

Hvala na pažnji