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

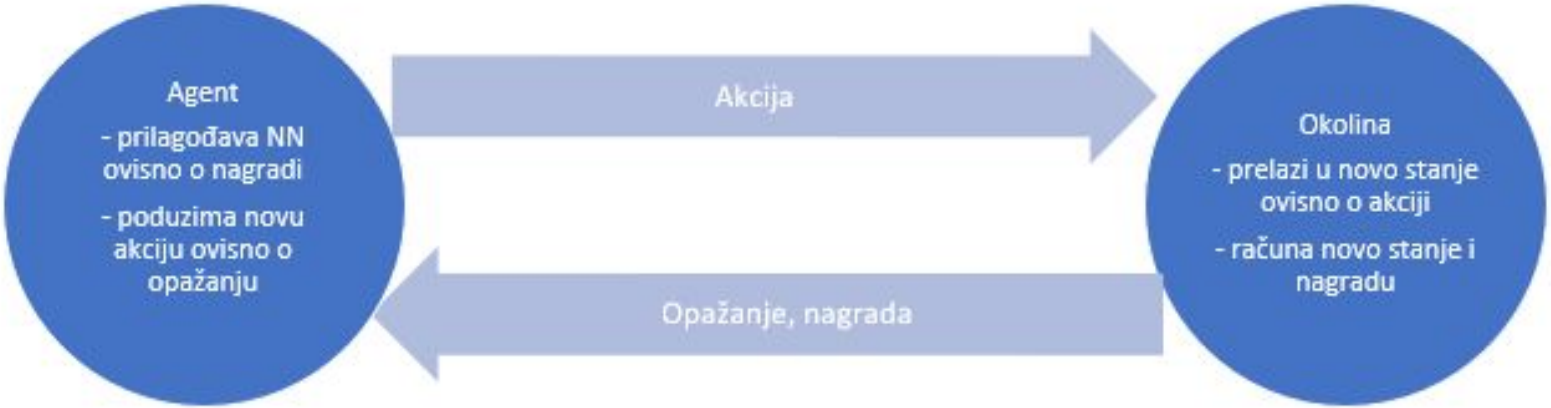
Implementacija okoline

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

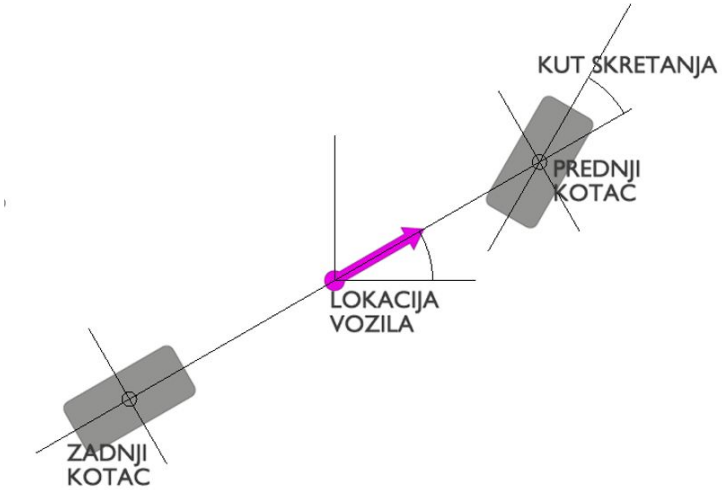
Situacija kada vraćamo nagradu u samo jednom (ili malom broju) koraka

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

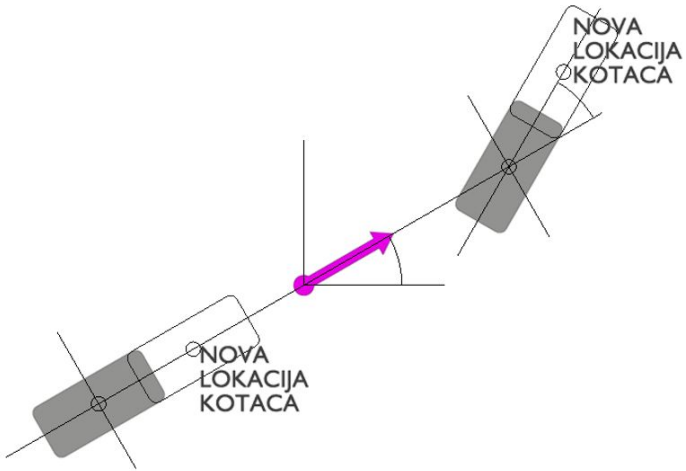
Velika vjerojatnost da agent nikada ne dođe u prihvatljivo stanje

Rješenje jako sporo (ili nikada) konvergira

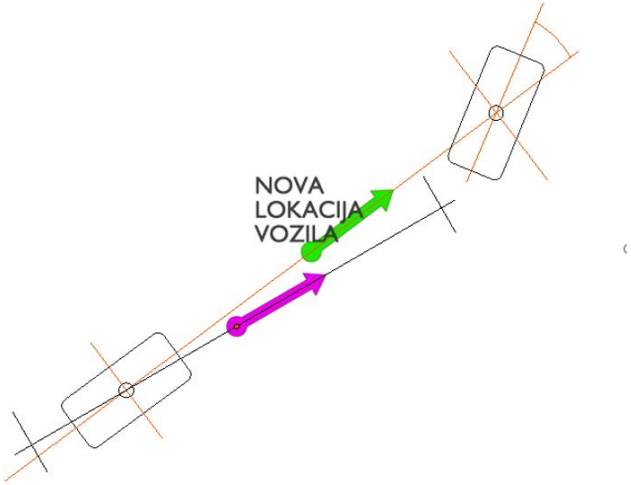
Implementacija - Okolina



Implementacija - Okolina



Implementacija - Okolina



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 ackije

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

Implementacija - Funkcija nagrade

$$\frac{\textit{početni kut} - \textit{krajnji kut}}{\textit{max. kut skretanja}}$$ - za svaki korak

$$25 + 75 \cdot \left(1 - \frac{\textit{tren. korak}}{\textit{max korak}}\right)$$ - kada vozilo dođe do cilja

$$-100$$ - ako vozilo izađe izvan definiranih granica

$$25 \cdot \left(1 - \frac{\textit{trenutna udaljenost}}{\textit{početna udaljenost}}\right)$$ - na posljednjem koraku

$$2.5$$ - ako se vozilo stvori unutar cilja

Treniranje

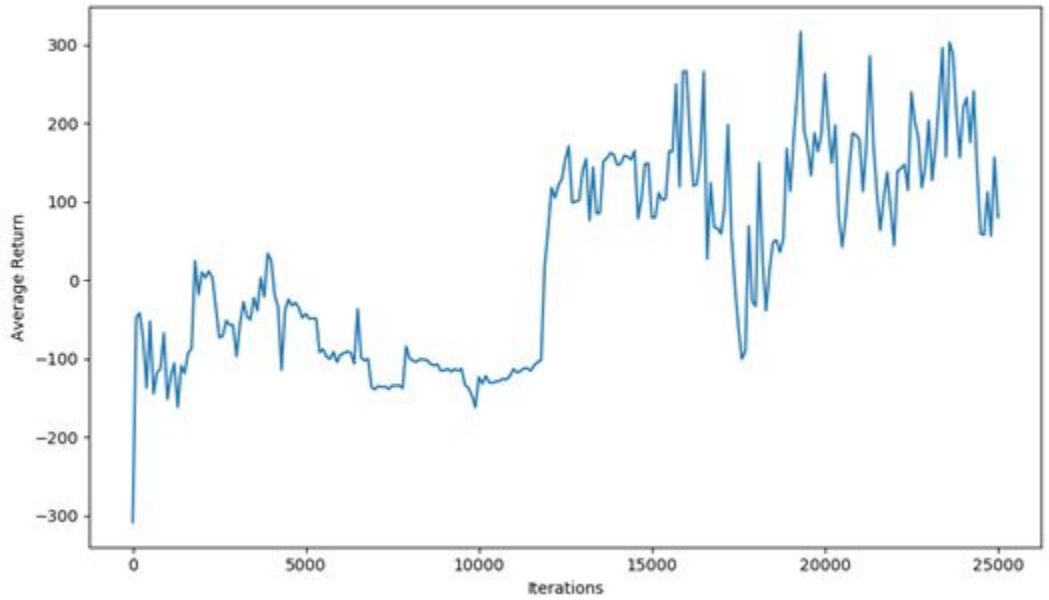
Koristeći TF-Agents biblioteku, Python

TF-Agents - pruža implementaciju uobičajenih algoritama za machine learning

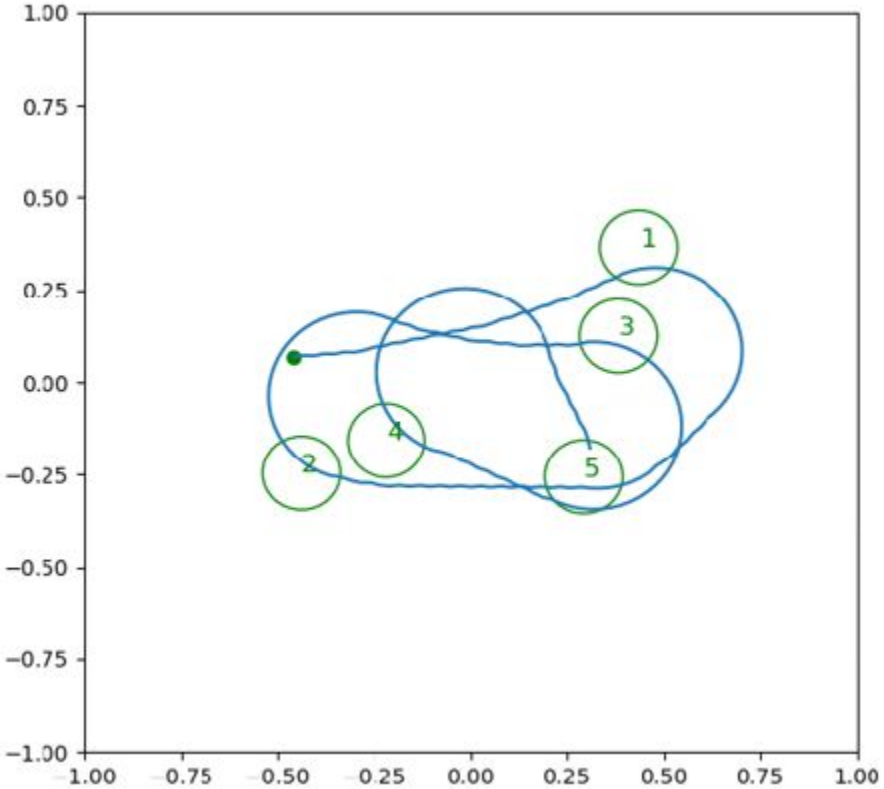
Koristi se mreža sa 2 Fully Connected sloja

Svaki sloj sadrži 32 neurona

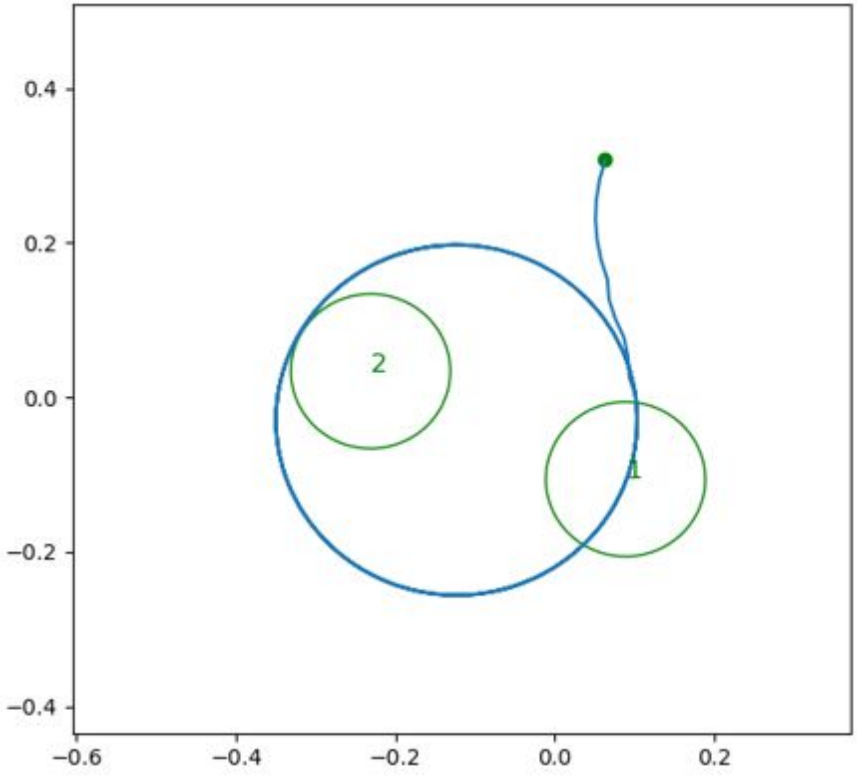
Treniranje


Average Return
300
200
100
0
−100
−200
−300
0
5000
10000
15000
20000
25000
Iterations

Rezultati


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
1.00
0.75
0.50
0.25
0.00
0.25
0.50
0.75
1.00
1
2
3
4
5

Rezultati



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