

# Mašinsko učenje – Domaći 5




## Učenje podsticajem

Nemanja Saveski 2023/3163

4. januar 2024.

# 1 Simulator

Zadata mapa izgleda ovako:

	1	2	3	4	5
A	S				
B					

Internu mapu definišemo na sledeći način:

$$\begin{bmatrix} W & W & W & W & W & W & W \\ W & S & M & M & M & M & W \\ W & B & W & B & W & G & W \\ W & W & W & W & W & W & W \end{bmatrix},$$

gde su polja označena slovima:

- W - zid
- S - startna pozicija
- M - pozicija na koju agent može da stane
- B - loše terminalno stanje (nagrada je -1)
- G - dobro terminalno stanje (nagrada je 3),

a agent može da se kreće:

- (-1, 0) - gore (akcija 0)
- (1, 0) - dole (akcija 1)
- (0, -1) - levo (akcija 2)
- (0, 1) - desno (akcija 3)

Kao što se može primetiti, u odnosu na datu mapu, dodat je okolni zid kako bi se smanjio broj ispitivanja da li je pozicija koja je agentovo sledeće stanje validna, što u ovako malom okruženju i ne donosi neko ubrzanje, ali za veća okruženja donosi.

## 2 Q learning

U Q learningu imamo tabelu Q vrednosti za svaki par stanja i akcija. Politiku dobijamo tako što biramo onu akciju za koju imamo najveću Q vrednost u datom stanju:

$$\pi^*(s) = \underset{a'}{\operatorname{argmax}} Q^*(s, a')$$

Vrednosti u Q tabeli ažuriramo na sledeći način:

$$q(s, a) = R(s') + \gamma \max_{a'} Q(s', a')$$

$$Q(s, a) \leftarrow Q(s, a) + \alpha(q(s, a) - Q(s, a))$$

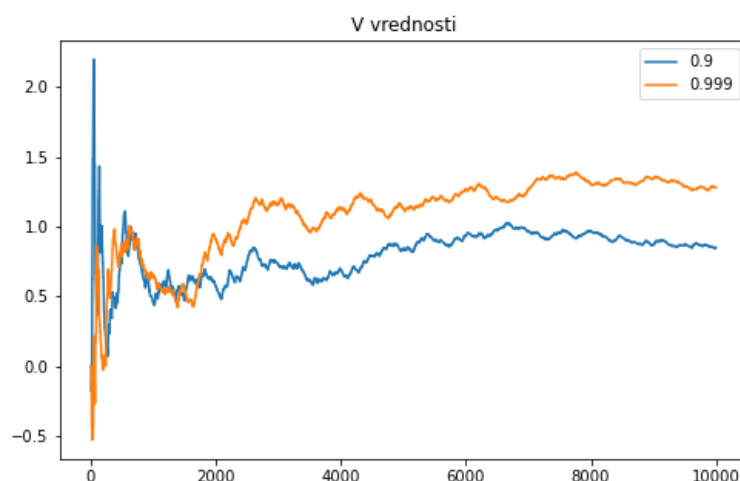
Ako uzmemo veće  $\gamma$ , dobili bismo agenta koji više daje na značaju nagradama koje su dalje u budućnosti, dok za manje  $\gamma$  agent se ipak više oslanja na trenutno stanje i trenutnu okolinu. Na svakom sledećem grafiku će biti prikazane V vrednosti za 10000 epizoda, za obe vrednosti  $\gamma$  (0.9 i 0.999):

$$V_t(s) = \max_a Q_t(s, a)$$

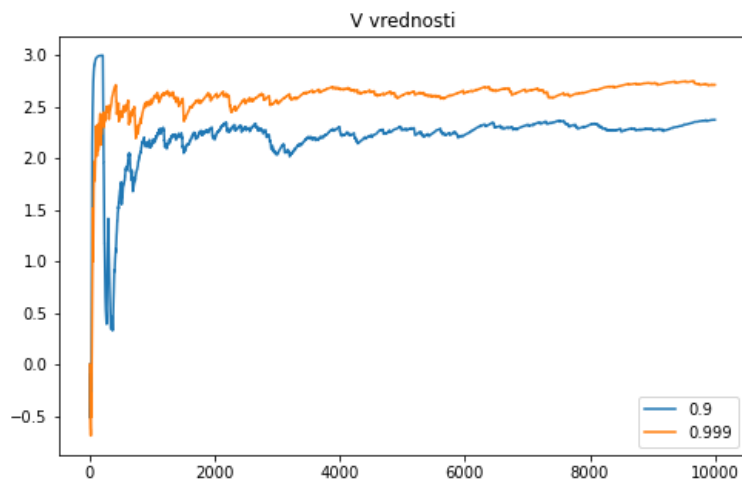
$\epsilon$  greedy strategija podrazumeva da sa verovatnoćom  $\epsilon$  istražujemo prostor (eksploatacija), dok sa verovatnoćom  $1 - \epsilon$  popravljamo odluke u već istraženom prostoru (eksploatacija). Prvo ćemo upotrebiti strategiju sa promenljivom stopom učenja koja je definisana na sledeći način:

$$\alpha_e = \frac{\ln(1 + e)}{1 + e}$$

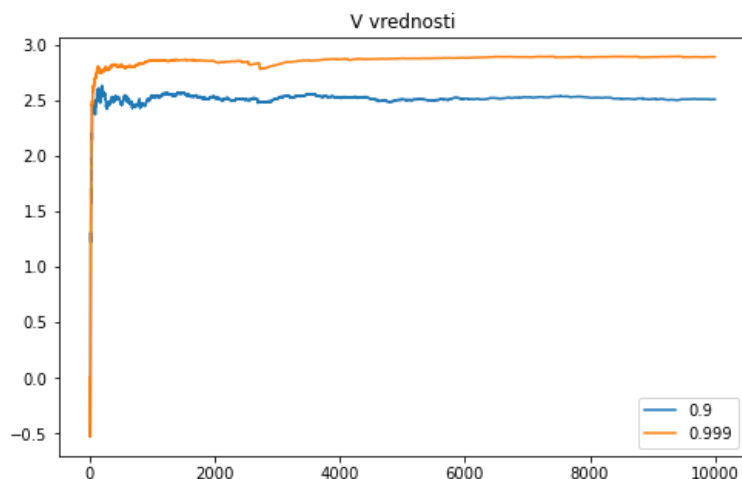
Dobijamo sledeći grafik za  $\epsilon = 0.9$  (za 10 test epizoda prosečna nagrada je 0.2):



$\epsilon = 0.2$  (za 10 test epizoda prosečna nagrada je 1.8):



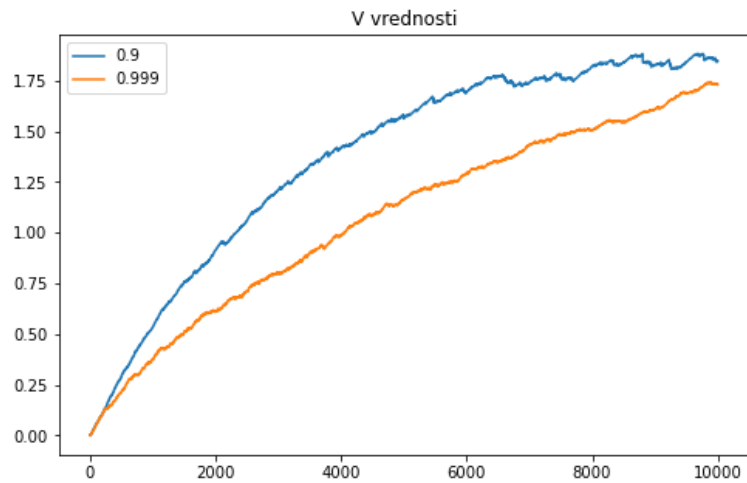
$\epsilon = 0.01$  (za 10 test epizoda prosečna nagrada je 3.0):



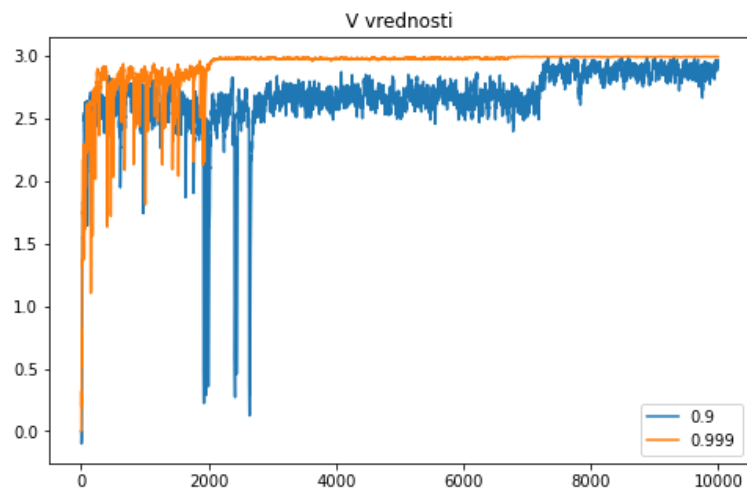
Vidimo da se u svakom slučaju od ova tri bolje agent ponaša bolje sa  $\gamma = 0.999$ , što je i očekivano (agent više gleda u budućnost, a pošto je malo okruženje, uspešnije prati put do cilja). Takođe možemo primetiti da se smanjuju fluktuacije u V vrednostima sa smanjenjem  $\epsilon$ . Zaključujemo i da su za agenta povoljnije manje  $\epsilon$  vrednosti, što i ima smisla jer u početku kada su sve Q vrednosti na 0 i posle par koraka u vrednostima oko 0, agent, iako mu je malo  $\epsilon$  (po definiciji retko kad vrši eksploraciju, a često eksploataciju) uzima na slučajan način akcije, pa ipak istražuje više nego što eksploatiše.

U slučaju konstantne stope učenja iskoristićemo  $\epsilon = 0.01$ , koje smo dobili u obučavanju agenta sa promenljivom stopom obučavanja.

Dobijamo za  $\alpha = 0.0005$  (za 10 test epizoda prosečna nagrada je 2.2):



$\alpha = 0.1$  (za 10 test epizoda prosečna nagrada je 3.0):



Dobijamo da je za veoma malu konstantnu stopu učenja, agent ne stiže da iskonvergira (kao i da brže uči za manje  $\gamma$ ), dok za stopu učenja 0.1, agent ( $\gamma = 0.999$ ) uspeva da konvergira u 3, dok je agent ( $\gamma = 0.9$ ), veoma blizu 3, što je ujedno i nagrada u dobrom terminalnom stanju.

### 3 REINFORCE

Ovaj algoritam predstavlja jedan od metoda gradijenta politike, u kom koristimo podatke iz jedne epizode da bismo "navukli" parametre politike, tako da težimo ka lokalnom/globalnom maksimumu kriterijumske funkcije koja je u stvari očekivana vrednost nagrade pod politikom  $\pi$ , sa parametrima  $\theta$ . Želimo da povećamo verovatnoće onih akcija koje su u prošlosti dovodile do dobrih nagrada, a smanjimo verovatnoće onih akcija koje su rezultovale u epizodama sa lošom nagradom.

Dakle naše parametre  $\theta$  ćemo menjati na sledeći način:

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} J(\theta)$$

Gradijent  $\nabla_{\theta} J(\theta)$  je po teoremi o gradijentu politike jednak:

$$\nabla_{\theta} J(\theta) = E_{\pi_{\theta}}(\nabla_{\theta} \ln \pi_{\theta}(a|s) Q^{\pi_{\theta}}(s, a))$$

Pošto koristimo samo jednu epizodu za računanje gradijenta gubi se očekivanje (imamo samo jedan uzorak, jednu epizodu):

$$\nabla_{\theta} J(\theta) = \nabla_{\theta} \ln \pi_{\theta}(a|s) Q^{\pi_{\theta}}(s, a)$$

Pošto imamo diskretno okruženje koristićemo softmax parametrizaciju, pa skor  $\nabla_{\theta} \ln \pi_{\theta}(a|s)$  postaje:

$$\nabla_{\theta} \ln \pi_{\theta}(a|s) = \phi(s, a) - E_{\pi_{\theta}} \phi(s, \cdot)$$

Funkcija  $\phi(s, a)$  ne predstavlja ništa nego one hot vektor koji ima jedinicu na poziciji  $a$ :

$$\phi(s, a) = \mathbb{1}_a,$$

dok očekivanje predstavlja u stvari predstavlja vrednosti koju nam je izbacila softmax funkcija na osnovu parametara  $\theta$  politike  $\pi_{\theta}$ , to su verovatnoće za izbor svake akcije pod politikom  $\pi_{\theta}$ . Softmax funkcija (softmax politika) je definisana na sledeći način:

$$\pi_{\theta}(s, a) = \frac{e^{\phi(s, a)^T \theta}}{\sum_{a'} e^{\phi(s, a')^T \theta}},$$

a vrednosti u eksponentu  $\phi(s, a)^T \theta$  postaju upravo parametri politike  $\theta$ , zbog prirode vektora  $\phi$ .

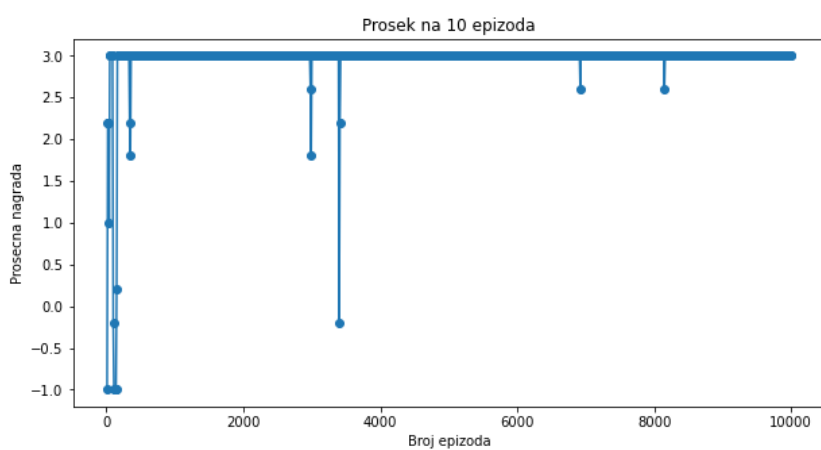
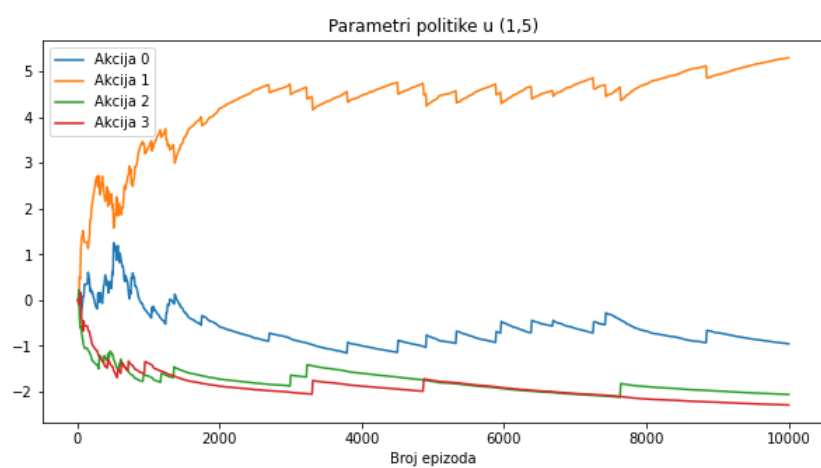
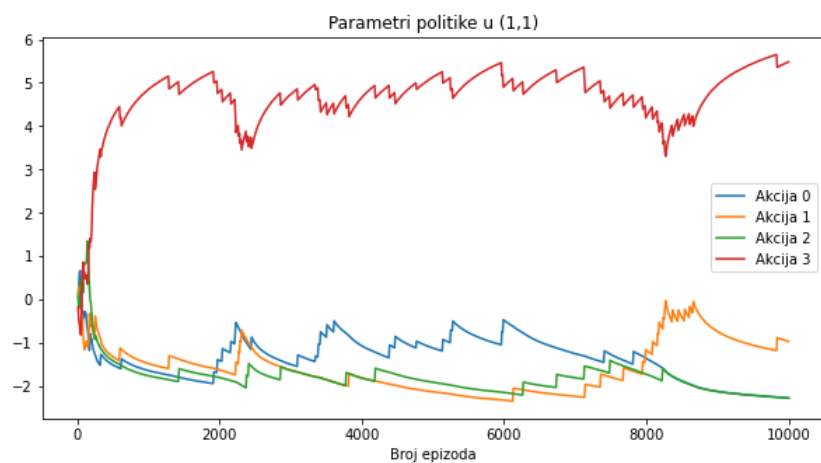
Sada samo još treba da aproksimiramo vrednost  $Q^{\pi_{\theta}}(s, a)$  na sledeći način:

$$Q^{\pi_{\theta}}(s, a) \approx v_t = \sum_{\tau=t}^T \gamma^{\tau-t} R_{\tau}$$

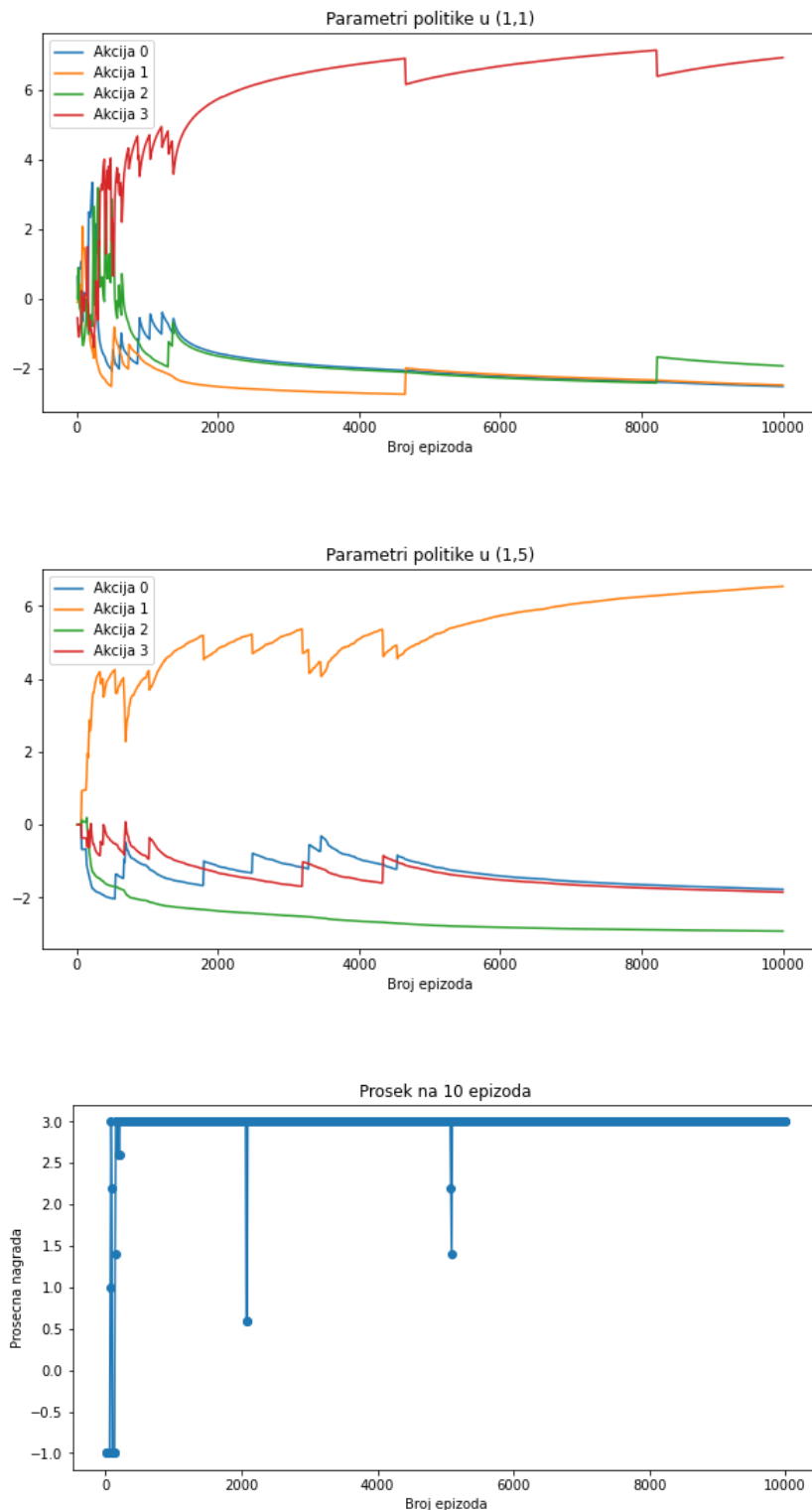
Konačno možemo pisati ažuriranje parametara na sledeći način:

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \ln \pi_{\theta}(a_t, s_t) v_t$$

U REINFORCE algoritmu dobijamo sledeće vrednosti, za stopu učenja  $\alpha = 0.1$ :



U REINFORCE algoritmu dobijamo sledeće vrednosti, za stopu učenja  $\alpha = 0.25$ :



Vidimo da se agent veoma dobro obučava za obe vrednosti stope obučavanja.