SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Diplomski sveučilišni studij računarstva

Primjena podržanog učenje za trening agenta u jednostavnim računalnim igrama

Strojno učenje

Antonio Berečić

Lovro Ružman

Osijek, 2025.

SADRŽAJ

1. Uvod

Podržano učenje (engl. Reinforcement Learning, RL) predstavlja granu strojnog učenja u kojoj agent uči optimalno ponašanje kroz interakciju s okolinom. Za razliku od nadziranog učenja, gdje se učenje odvija na temelju označenih podataka, u podržanom učenju agent dobiva povratnu informaciju u obliku nagrade ili kazne, ovisno o kvaliteti svojih akcija. Kroz pokušaje i pogreške, cilj agenta je maksimizirati ukupnu nagradu tijekom vremena.

U kontekstu računalnih igara, agent djeluje u virtualnom okruženju koje mu u svakom trenutku pruža određeno stanje. Na temelju tog stanja, agent bira akciju, a okolina zauzvrat vraća novo stanje i nagradu. Ovakav interaktivni proces omogućuje učenje strategije ponašanja kojom agent može uspješno savladati određenu igru.

Korišteni su standardizirani simulacijski okoliši iz paketa OpenAI Gymnasium, kao što su FrozenLake-v1, CartPole-v1 i LunarLander-v3. Cilj je primjenom različitih RL algoritama, poput Deep Q-learninga (DQN) i metoda temeljenih na policy gradientu (A2C i PPO), istrenirati agente koji će uspješno savladati zadane igre.

2. Opis korištenih skupova podataka, eksplorativna analiza i predobrada podataka

Za razliku od tradicionalnih pristupa strojnog učenja, u kojima se koristi unaprijed definiran i strukturiran skup podataka (npr. slike, tablice, tekst), u podržanom učenju skup podataka nastaje dinamički tijekom interakcije agenta s okolinom. Drugim riječima, podaci se generiraju sintetski u stvarnom vremenu na temelju ponašanja agenta.

Korištena su virtualna simulacijska okruženja iz paketa **OpenAI Gymnasium**, konkretno:

- **FrozenLake-v1** – diskretno okruženje s ciljem pronalaska puta do cilja izbjegavajući rupe u ledu.
- **CartPole-v1** – kontinuirano okruženje gdje agent mora balansirati štap na kolicima.
- **LunarLander-v3** – simulacija slijetanja letjelice na površinu s ciljem preciznog i sigurnog slijetanja.

Za sva okruženja primijenjeni su različiti algoritmi podržanog učenja:

- **Deep Q-learning (DQN)** – koristi memoriju prošlih iskustava (replay buffer) i neuronsku mrežu za predviđanje Q-vrijednosti.
- **Advantage Actor-Critic (A2C) –** kombinira pristup aktera (actor) koji odlučuje o akciji, i kritičara (critic) koji procjenjuje kvalitetu te akcije pomoću aproksimacije value funkcije. Uči i politiku i funkciju vrijednosti istovremeno.
- **Proximal Policy Optimization (PPO)** – napredniji policy gradient algoritam koji koristi klip funkciju za stabilnije ažuriranje politike i radi s mini-batch pristupom.

2.1 Skup podataka

U kontekstu RL-a, svaki podatkovni uzorak sastoji se od tzv. *tranzicije*:

(state, action, reward, next_state, done)

Ove tranzicije se pohranjuju u memoriju (replay buffer) i koriste kao iskustvo za učenje (kod DQN-a), ili se bilježe cijele epizode kao niz stanja i nagrada (kod A2C-a i PPO-a).

- Nema unaprijed definiranog skupa podataka – podaci nastaju tijekom interakcije.
- Kod DQN-a koristi se memorija od 10.000 uzoraka.
- Kod A2C-a i PPO-a koristi se čitava epizoda (ili više njih) kao batch za treniranje.

2.2 Predobrada podataka

S obzirom na različitu prirodu okolina, provedena je jednostavna predobrada:

- **FrozenLake-v1**: diskretna stanja (0–15) enkodirana su u *one-hot* vektore duljine 16.
- **CartPole-v1** i **LunarLander-v3**: stanja su predstavljena kao vektori kontinuiranih vrijednosti (npr. pozicija, brzina, kut...), koji se mogu koristiti direktno jer su već skalirani.
- Nema klasičnog označavanja klasa – agent putem nagrada sam uči koje akcije vode prema boljem ishodu.

2.3 Eksplorativna analiza

Iako RL ne koristi klasičnu analizu skupova podataka kao u nadziranom učenju, provedene su sljedeće analize radi boljeg razumijevanja ponašanja agenta i uspješnosti algoritama:

- Vizualizacija ukupne nagrade po epizodi
- Broj koraka do kraja epizode (uspjeh/pad)
- Praćenje brzine konvergencije i stabilnosti ponašanja
- Usporedba učenja s i bez skliskog terena u *FrozenLake-v1*
- Usporedba različitih algoritama u istom okruženju

Kroz analizu rezultata dobiveni su uvidi u to koliko brzo i stabilno svaki algoritam uči, te koliko je robusno njegovo ponašanje u različitim situacijama.

3. Opis korištenih metoda / algoritama

Za rješavanje problema učenja optimalne politike djelovanja u simuliranim okolišima, koristili smo više algoritama iz područja reinforcement learninga (RL), odnosno učenja pojačanjem. Testirali smo i usporedili tri različita pristupa: Deep Q-learning (DQN), Proximal Policy Optimization (PPO) i Advantage Actor-Critic (A2C).

3.1 Deep Q-learning (DQN)

DQN je unaprijeđena verzija klasičnog Q-learning algoritma, koja koristi duboku neuronsku mrežu za aproksimaciju Q-funkcije (funkcije akcije-vrijednosti). Na taj način, može se primijeniti i na okoliše s kontinuiranim ili velikim diskretnim prostorom stanja, gdje bi standardni Q-table pristup bio nepraktičan.

Struktura modela:

- Ulazni sloj prima stanje okoliša (npr. pozicija i brzina objekta).
- Dva skrivena sloja koriste ReLU aktivaciju.
- Izlazni sloj daje Q-vrijednosti za svaku moguću akciju.

Tehnike stabilizacije učenja:

- Replay buffer: pohranjuje prošla iskustva i omogućuje treniranje na slučajno uzorkovanim podacima radi smanjenja korelacija.
- Target mreža: kopija glavne mreže koja se periodički ažurira kako bi se stabilizirale Q-vrijednosti.
- Epsilon-greedy strategija: koristi se za balansiranje između istraživanja i eksploatacije.

Trenirali smo DQN agente na igrama CartPole-v1, FrozenLake-v1 i LunarLander-v3, prateći ukupnu nagradu po epizodi. U svrhu analize performansi, snimljeni su grafovi koji prikazuju:

- Prosječnu nagradu kroz prozor od 50 epizoda.
- Standardnu devijaciju nagrade kao indikator stabilnosti.

3.2 Proximal Policy Optimization (PPO)

PPO je napredna policy-gradient metoda koja koristi optimizaciju s ograničenjem na promjene politike (klipiranjem omjera novih i starih politika). PPO je stabilniji i učinkovitiji od standardnih policy-gradient algoritama.

Prednosti PPO algoritma:

- Robustnost na velike promjene u politici.
- Efikasno korištenje podataka iz simulacije (više epoha po uzorku).
- Manja potreba za finim podešavanjem hiperparametara.

PPO smo implementirali koristeći postojeće biblioteke (npr. stable-baselines3) i testirali na istim igrama kao i DQN. Dobiveni rezultati pokazali su znatno stabilnije učenje, osobito u igri LunarLander, gdje PPO postiže konzistentno dobre nagrade kroz veći broj epizoda.

3.3 Advantage Actor-Critic (A2C)

A2C je još jedna policy-based metoda koja kombinira policy (actor) i value (critic) pristupe. Actor odlučuje koju akciju poduzeti, dok critic procjenjuje koliko je ta akcija dobra (tj. koristi aproksimaciju value funkcije).

Prednosti A2C algoritma:

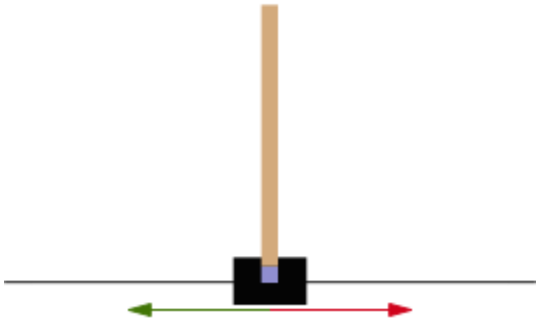
- Brže konvergira od DQN u nekim igrama.
- Uči politiku i funkciju vrijednosti istovremeno.

Korištenjem A2C u okolišima poput CartPole i FrozenLake, dobivene su konzistentne politike koje brzo konvergiraju i zadržavaju visoku nagradu kroz epizode.

3.4 Rezultati treniranja na igri CartPole-v1

Igra CartPole-v1 sastoji se od dvije moguće akcije (0,1) koje indiciraju smjer fiksne sile koja gura kolica: 0 – guraj kolica desno, 1 – guraj kolica lijevo. Pošto je cilj držati štap uspravno što duže,

nagrada od +1 se daje za svaki korak napravljen, epizoda se računa riješenom ako ukupna nagrada dosegne 500.



Slika 1. Prikaz igre CartPole-v1

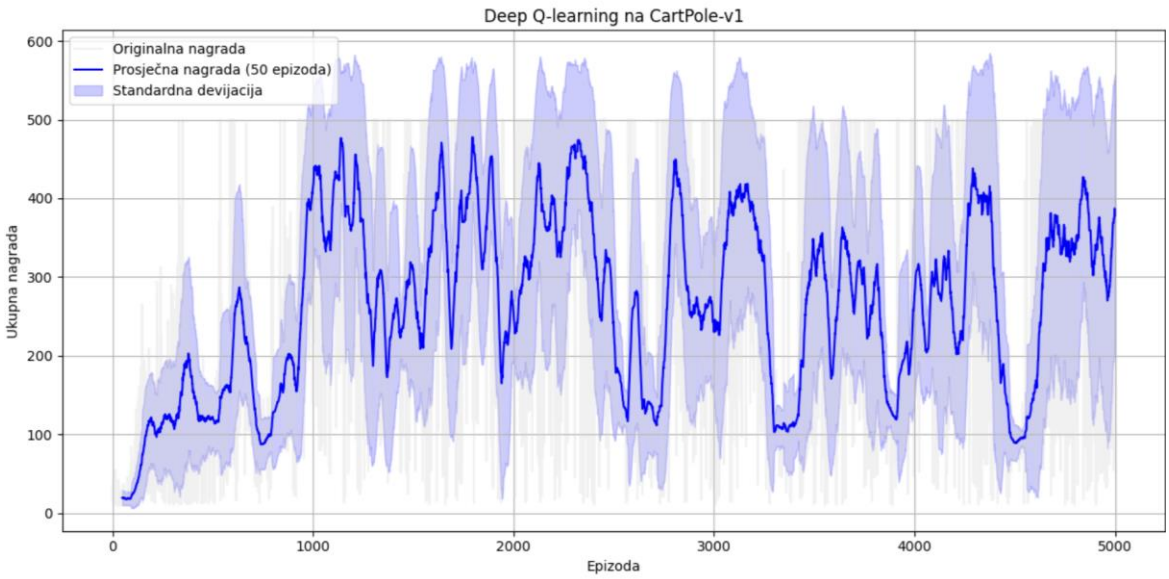
Početno stanje kreće od uniformne nasumične vrijednosti iz polja (-0.05, 0.05). Epizoda završava ukoliko se dogodi jedan od 3 scenarija:

- Kut štapa veći od ±12°
- Pozicija kolica je veća od ±2.4 (Centar kolica je dosegao rub prikaza)
- Dužina epizode je veća od 500

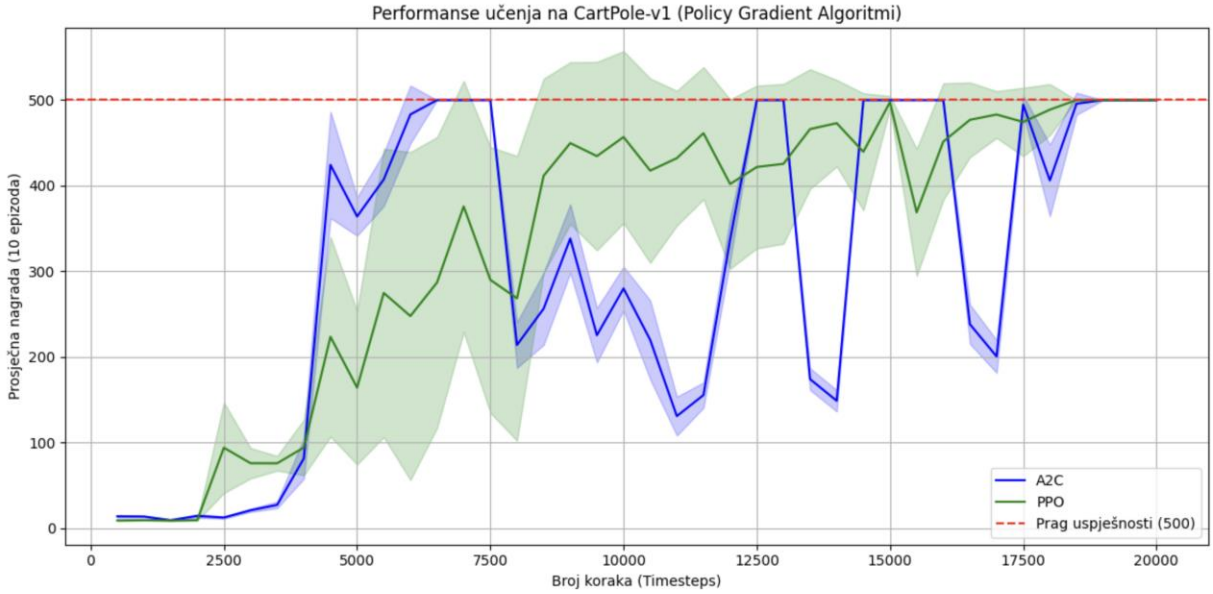
Kako bismo objektivno usporedili performanse svakog algoritma, trenirali smo sve agente u istom okolišu CartPole-v1, no s različitim brojem epizoda zbog razlika u implementaciji i prirodi algoritama:

Metoda	Broj epizoda / koraka	Vrijeme treniranja
DQN	5000 epizoda	14 minuta i 53 sekunde
PPO	20 000 koraka (timesteps)	1 minuta i 27 sekundi
A2C	20 000 koraka (timesteps)	1 minuta i 24 sekunde

Tablica 1. Parametri treniranja



Slika 2. Performance treninga za DQN



Slika 3. Performance treninga za A2C i PPO

Rezultati:

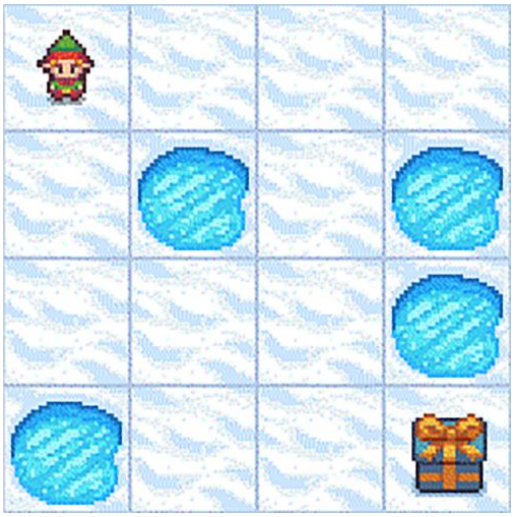
- PPO je vrlo brzo dosegao maksimalnu nagradu od 500 i održavao ju gotovo bez oscilacija kroz većinu učenja.

- A2C je pokazao sličnu dinamiku kao PPO, uz nešto veću varijabilnost i nešto sporiju konvergenciju.
- DQN je također postizao maksimalne nagrade, ali je tijekom cijelog treniranja imao izražene oscilacije, što ukazuje na manju stabilnost i osjetljivost na hiperparametre.

Grafovi prosječne nagrade i standardne devijacije jasno pokazuju da su PPO i A2C znatno stabilniji tijekom učenja i brže dolaze do optimalne politike.

3.5 Rezultati treniranja na igri FrozenLake-v1

Igra FrozenLake-v1 započinje sa igračem sa lokaciji [0,0] mape „frozen lake“ gdje je cilj lociran na samom suprotnom kraju mape [3,3]. Rupe u ledu su raspoređene u određena mjesta kada se koristi unaprijed određena mapa ili u nasumične lokacije kada je nasumična mapa generirana. Igrač dobija nagradu od +1 ukoliko dođe do cilja ili 0 ukoliko dođe do rupe ili zaleđenog.



Slika 4. Prikaz igre FrozenLake-v1

Prostor akcija je definiran kao : 0 – pomakni se u lijevo, 1 - pomakni se prema dolje, 2 – pomakni se u desno, 3 – pomakni se prema gore. Igrač radi poteze dok ne upadnu u rupu ili ne dođe do cilja. Uz to pod je sklizak pa se uz svaki potez igrača postoji 1/3 šansa u bilo kojem okomitom smjeru te 1/3 vjerojatnosti u nasumičnoj kretnji u oba smjera. Epizoda završava ukoliko:

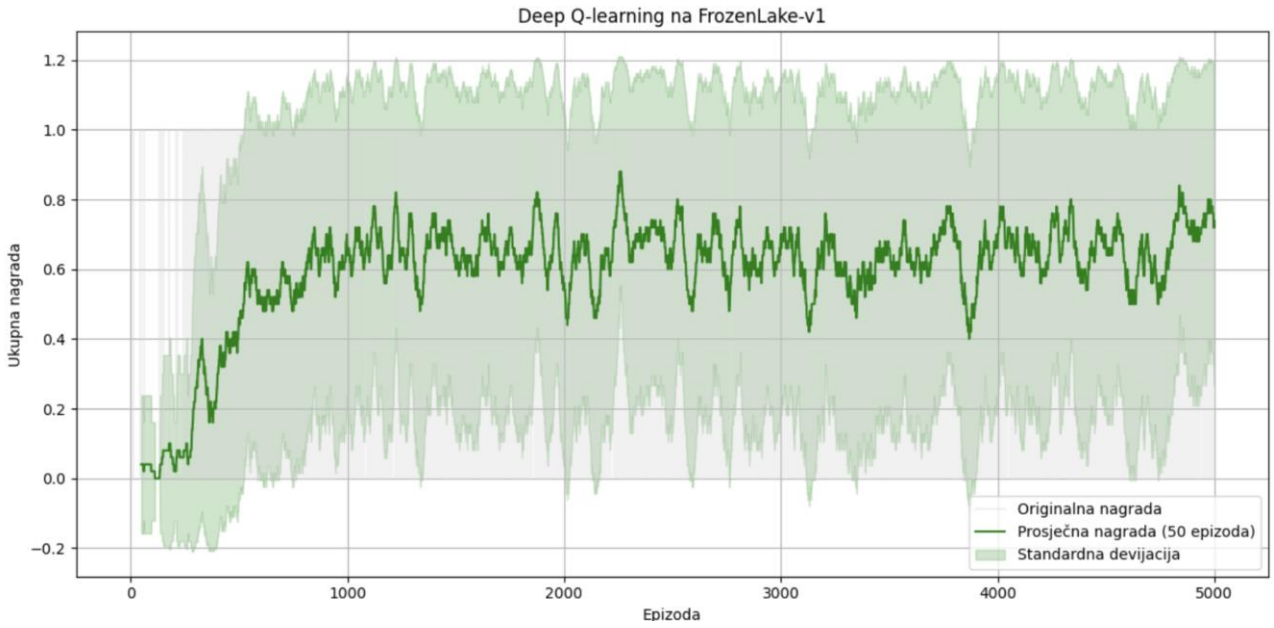
- Igrač završi u rupi.
- Igrač dođe do cilja.

- Iskoristi maksimalan broj koraka (100).

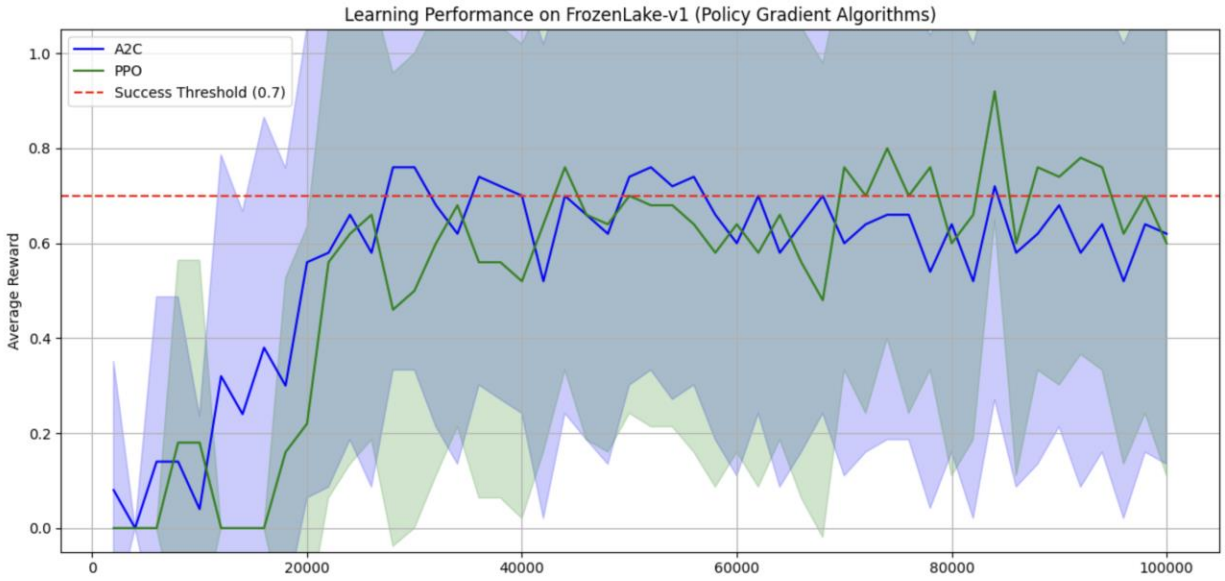
Kako bismo usporedili ponašanje različitih algoritama u stohastičnim okolišima, trenirali smo agente u igri FrozenLake-v1 s uključenim klizanjem (is_slippery=True). Cilj je bio razviti politiku koja maksimizira vjerojatnost uspješnog dolaska do cilja (nagrada = 1), gdje je prag uspjeha postavljen na prosječnu nagradu ≥ 0.7 (dok je uključeno is_slippery nije moguće imati perfektnu prosječnu nagradu od 1).

Metoda	Broj epizoda / koraka	Vrijeme treniranja
DQN	5000 epizoda	1 minuta i 31 sekunda
PPO	100 000 koraka (timesteps)	2 minute i 51 sekunda
A2C	100 000 koraka (timesteps)	3 minute i 9 sekundi

Tablica 2. Parametri treniranja



Slika 5. Performance treninga za DQN



Slika 6. Performance treninga za A2C i PPO

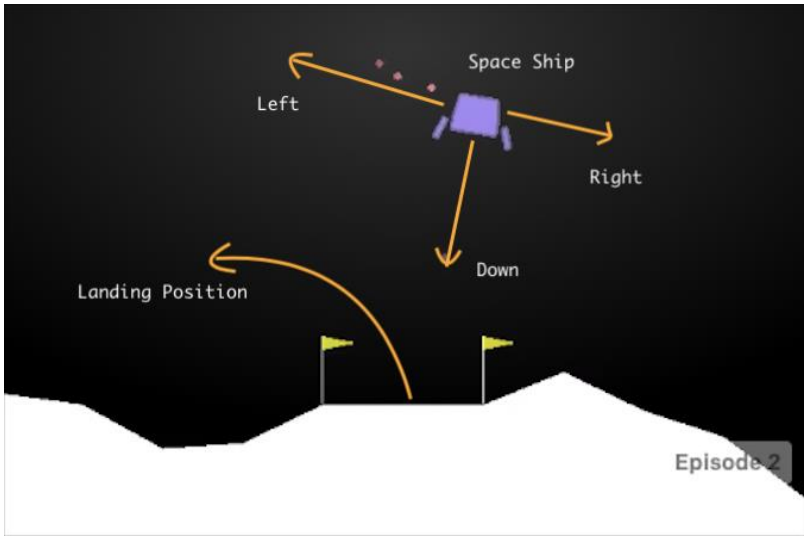
Rezultati:

- PPO je u ranom stadiju treniranja dostigao zadovoljavajuću prosječnu nagradu (>0.7), uz visoku stabilnost kroz nastavak epizoda.
- A2C je imao sličnu krivulju, no s većom varijabilnošću i ponešto manjom stabilnošću.
- DQN je pokazivao najveće fluktuacije, ali je povremeno uspijevao postići maksimalnu nagradu. Međutim, nedostatak konzistencije ukazuje na teže učenje u stohastičnom okruženju.

Grafovi prosječne nagrade i standardne devijacije jasno pokazuju da su policy-based algoritmi (PPO, A2C) bolje prilagođeni za okoliše sa slučajnim prijelazima stanja, za razliku od DQN-a, koji zahtijeva dodatno podešavanje i više epizoda za stabilan učinak.

3.6 Rezultati treniranja na igri LunarLander-v3

Igra LunarLander-v3 nastoji sletjeti letjelicu unutar zastavica predefiniranih na nasumično generiranoj mapi.



Slika 7. Primjer igre LunarLander-v3

Postoje četiri diskretne radnje koje agent može obaviti: 0 – radi ništa, 1 – upali lijevi orijentacijski motor, 2 – upali glavni motor, 3 – upali desni orijentacijski motor. Stanje je 8-dimeznionalni vektor koji opisuje x i y poziciju letjelice, linearne brzine u x i y smjeru, kutnu brzinu i dvije bool vrijednosti koje opisuju je li ijedna noga letjelice u kontaktu sa zemljom.

Letjelica kreće epizodu na vrhu sredine prikaza sa nasumičnom silom primijenjenom na njezin centar mase. Nakon svakog koraka odobrena je nagrada, a ukupna nagrada je suma nagrada svih koraka u epizodi. Za svaki korak nagrada je:

- Povećana/smanjena što je letjelica bliže/dalje do podloge za slijetanje.
- Povećana/smanjena što sporije/brže se letjelica kreće.
- Smanjena što je više letjelica nakrenuta (kut nije horizontalan).
- Povećana je za 10 za svaku nogu koja je u kontaktu sa tlom.
- Smanjena je za 0.03 za svaki kadar u kojem se bočni motor pali.
- Smanjena za 0.03 za svaki kadar u kojem se glavni motor pali.

Epizoda ima dodatnu nagradu od -100 ili +100 za uspješno/neuspješno slijetanje. Epizoda se smatra riješenom ukoliko je ukupna nagrada barem 200 bodova.

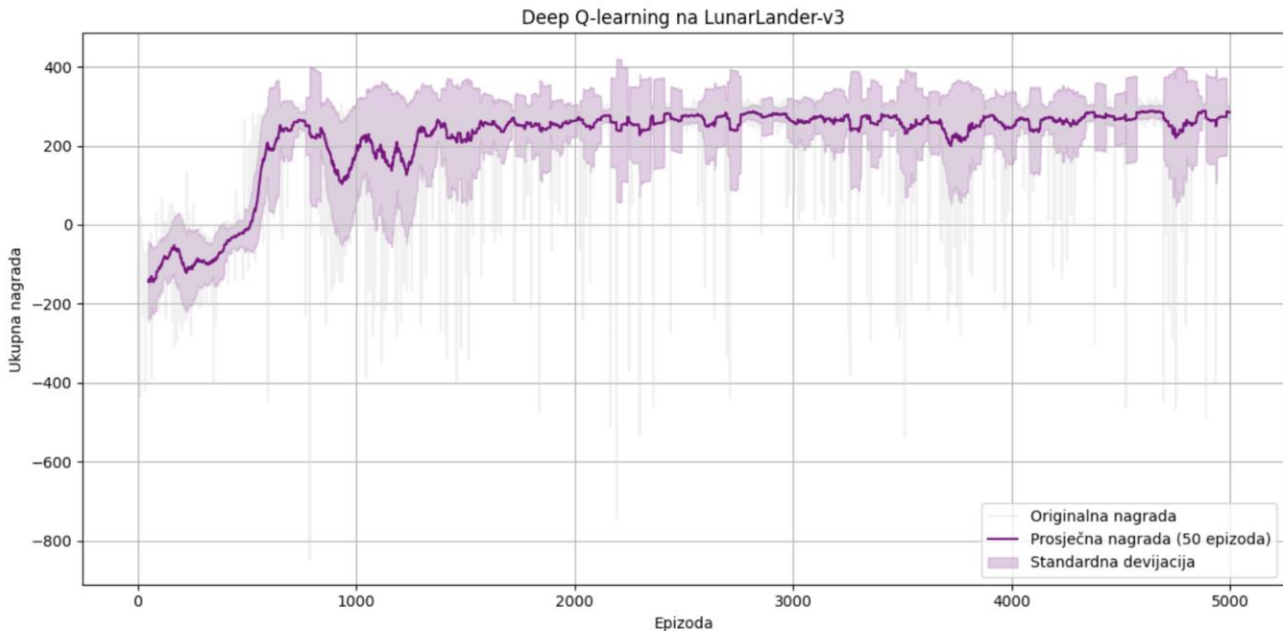
- Epizoda se raskida ukoliko:
- Letjelica se sruši
- Letjelica izlazi izvan vidnog polja

Letjelica nije budna (tijelo koje nije budno je tijelo koje se ne kreće i ne sudara se sa niti jednim drugim tijelom).

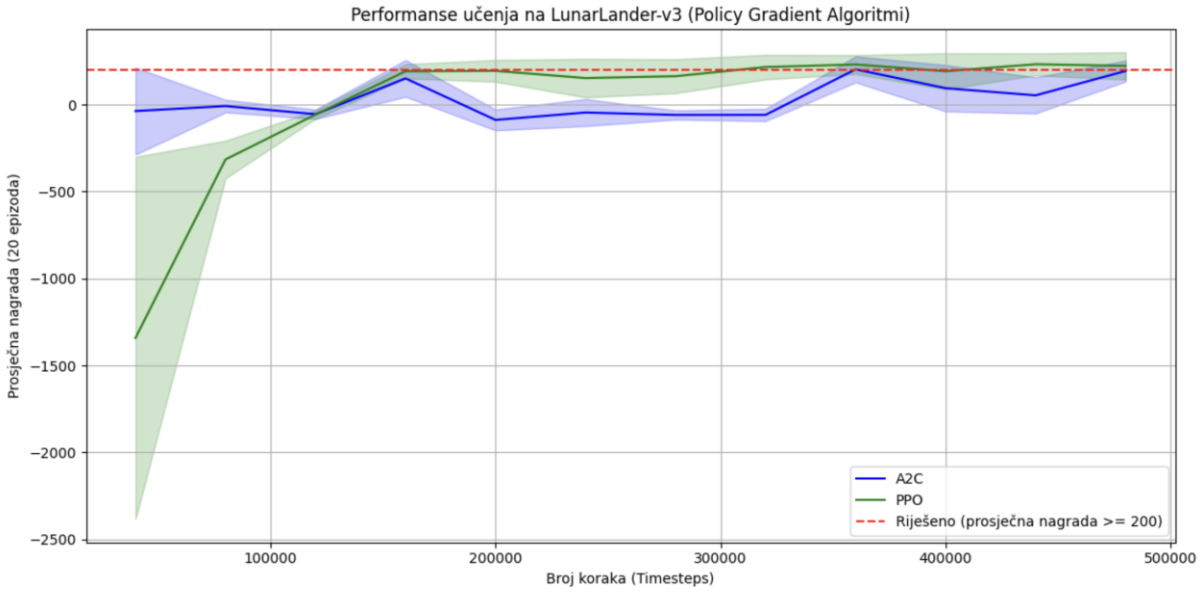
Okoliš LunarLander-v3 predstavlja složeniji zadatak s kontinuiranim prostorom stanja i diskretnim akcijama. Cilj je uspješno sletjeti svemirsku letjelicu, uz izbjegavanje udara i maksimiziranje ukupne nagrade. Prag uspjeha postavljen je na prosječnu nagradu ≥ 200.

Algoritam	**Broj epizoda / koraka**	**Vrijeme treniranja**
DQN	5000 epizoda	17 minuta i 6 sekundi
PPO	480 000 koraka (timesteps)	13 minuta i 2 sekunde
A2C	480 000 koraka (timesteps)	13 minuta i 5 sekundi

Tablica 3. Parametri treniranja



Slika 8. Performance treninga za DQN



Slika 9. Performance treninga za A2C i PPO

Rezultati:

- PPO je u ovom zahtjevnijem okolišu postigao najvišu prosječnu nagradu (~246) i održavao je iznad praga uspješnosti kroz veći dio treniranja.
- A2C je također dosegnuo prag uspješnosti (~200), ali s nešto većom varijabilnošću i osjetljivosti na promjene tijekom treniranja.
- DQN, iako sposoban za učenje uspješne politike, imao je sporiju konvergenciju i osjetljivost na stohastičnost okoliša, ali je u kasnijim epizodama postizao nagrade veće od 250.

4. Evaluacija izgrađenih modela / algoritama

Za mjeru evaluacije koristili smo metriku prosječne nagrade po epizodi kao osnovni pokazatelj kvalitete naučene politike. Za svaki model provedena je evaluacija na 100 epizoda, uz računanje srednje vrijednosti i standardne devijacije.

4.1 Metrike evaluacije

- Prosječna nagrada: prosjek ukupnih nagrada po epizodi.
- Standardna devijacija nagrada: pokazatelj stabilnosti ponašanja agenta.
- Vrijeme treniranja svakog modela također je zabilježeno kao mjera računalne efikasnosti.

4.2 Rezultati evaluacije

Algoritam	Okoliš	Prosječna nagrada ± StdDev	Vrijeme izvođenja
DQN	CartPole-v1	449.65 ± 127.25	14 minuta i 53 sekunde
PPO	CartPole-v1	499.50 ± 3.34	1 minuta i 27 sekundi
A2C	CartPole-v1	500.00 ± 0.00	1 minuta i 4 sekunde
DQN	FrozenLake-v1	0.75 ± 0.43	1 minuta i 13 sekundi
PPO	FrozenLake-v1	0.76 ± 0.43	2 minute i 51 sekunda
A2C	FrozenLake-v1	0.74 ± 0.44	3 minute i 9 sekundi
DQN	LunarLander-v3	250.61 ± 123.12	17 minuta i 6 sekundi
PPO	LunarLander-v3	246.85 ± 38.71	13 minuta i 2 sekunde
A2C	LunarLander-v3	200.75 ± 107.35	13 minuta i 5 sekundi

Tablica 4. Rezultati evaluacije

Rezultati evaluacije pokazuju da različiti algoritmi postižu različite performanse u zavisnosti od okruženja. U okruženju CartPole-v1, najbolji rezultat je postigao A2C algoritam s prosječnom nagradom od 500.00 ± 0.00, što ukazuje na stabilnost i efikasnost ovog algoritma u jednostavnijem okruženju. PPO je također pokazao vrlo dobre rezultate, dok je DQN imao nešto nižu nagradu i veću varijancu te znatno veće vrijeme izvođenja u odnosu na ostala dva algoritma.

U složenijem i stohastičkom okruženju FrozenLake-v1, svi algoritmi imaju vrlo slične rezultate, s nagradama između 0.70 i 0.76, što nam govori da je ovo okruženje izazovno za sve testirane metode, bez jasnog pobjednika.

Kod zadatka LunarLander-v3, DQN je pokazao najbolju prosječnu nagradu (250.61 ± 123.12), iako s visokom varijancom, što upućuje na nestabilnost u učenju. PPO je postigao sličan rezultat s manjom varijancom, dok je A2C imao najnižu prosječnu nagradu, ali također s visokom standardnom devijacijom.

Ukupno gledano, može se zaključiti da ne postoji univerzalno najbolji algoritam – rezultati zavise od specifičnosti okruženja. A2C se najbolje pokazao u deterministčkim okruženjima poput CartPole, dok DQN ima prednost u složenijim zadacima poput LunarLander, unatoč većoj varijabilnosti.

4.3 Dodatna evaluacija

Kako bi se dodatno procijenila kvaliteta ponašanja agenata, moguće je provesti i subjektivnu evaluaciju, posebno korisnu u situacijama gdje:

- ponašanje nije u potpunosti kvantificirano kroz nagradu (npr. prirodnost pokreta),
- ponašanje se treba ocijeniti ljudskom percepcijom.

Subjektivna evaluacija se može provesti putem vizualne inspekcije ponašanja agenata u simulacijama ili kroz anketiranje korisnika koji ocjenjuju kvalitetu ponašanja na temelju unaprijed definiranih kriterija (npr. efikasnost, greške).

5. Zaključak

U okviru ovog projektnog zadatka istražena je i implementirana primjena algoritama podržanog učenja (Reinforcement Learning) u standardiziranim simulacijskim igrama. Korištenjem okruženja iz OpenAI Gymnasium biblioteke, uspješno su istrenirani agenti u trima različitim okolišima – CartPole-v1, FrozenLake-v1 i LunarLander-v3 – koristeći tri različita algoritma: Deep Q-learning (DQN), Proximal Policy Optimization (PPO) i Advantage Actor-Critic (A2C).

Kroz provedenu analizu rezultata i evaluaciju modela, uočeno je sljedeće:

- PPO i A2C su pokazali veću stabilnost, bržu konvergenciju i manju osjetljivost na hiperparametre u odnosu na DQN.
- DQN je, iako sposoban za postizanje visokih nagrada, imao izražene oscilacije i veću varijabilnost, osobito u stohastičnim okruženjima poput FrozenLake.
- U zahtjevnijem okolišu LunarLander-v3, PPO se pokazao najefikasnijim, dok je A2C zaostajao, a DQN zahtijevao više epizoda za stabilne rezultate.

Evaluacija na 100 epizoda po modelu omogućila je kvantitativnu usporedbu pomoću prosječnih nagrada i standardne devijacije. Osim toga, integrirane su i vizualne simulacije te razrađen plan za buduću subjektivnu evaluaciju na temelju ljudske percepcije ponašanja agenta.

Moguća unaprjeđenja uključuju:

- Implementaciju naprednijih algoritama poput DDPG, SAC ili TD3 za kontinuirane akcijske prostore.
- Proširenje evaluacije kroz stvarnu ljudsku procjenu i anketne podatke.
- Trening u vlastitim, kompleksnijim ili realističnijim okolišima razvijenima u Unity-ju ili drugim simulatorima.

Projekt je uspješno demonstrirao snagu i fleksibilnost podržanog učenja te pružio čvrstu osnovu za daljnje istraživanje i razvoj složenijih inteligentnih agenata u simuliranim i stvarnim uvjetima.