Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.8
DOI: https://doi.org/10.24867/07BE44Mladenovic

PRIMENA UČENJA SA USLOVLJAVANJEM ZA OBUČAVANJE AGENTA ZA AUTONOMNU VOŽNJU AUTOMOBILA U SIMULATORU AIRSIM

USING REINFORCEMENT LEARNING TO TRAIN AN AGENT FOR AUTONOMOUS DRIVING IN THE AIRSIM SIMULATOR

Miloš Mladenović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Učenje uslovljavanjem je napretkom dubokog učenja i hardvera kao i razvojem novih naučno-tehnoloških izazova kao što su razvoj softvera za robote i samovozeće automobile postalo plodno istraživačko tle za sve naučnike i inženjere zainteresovane za ovu oblast. U ovom radu predstavljen je drugačiji pristup problemu autonomne vožnje u simulatoru - u kom se kombinuju tehnike računarskog vida i učenja sa uslovljavanjem da se napravi agent koji će se uspešno kretati u simuliranom okruženju. Evaluacija agenta urađena je poređenjem performansa modela u odnosu na postizanje zadatog cilja.*

Ključne reči: *Učenje uslovljavanjem, autonomna vožnja, simulacija, računarski vid, nagrade, neuronske mreže*

Abstract – *With the advancement of deep learning and hardware, along with emergence of new technological challenges like software for robots and self-driving cars – reinforcement learning has become a fertile ground for researchers interested in this field. In this paper a novel approach has been proposed to solving the problem of autonomous driving in simulator, which combines techniques of computer vision and reinforcement learning to create a software agent that can drive a car successfully in a simulation. Evaluation of the agent has been done by comparing results and expected/given goal.*

Keywords: *Reinforcement learning, autonomous driving, simulation, computer vision, rewards, neural networks*

1. UVOD

U poslednjih nekoliko godina veliki napredak na polju razvoja grafičkih čipova i njihove procesorske moći doveo je do toga da je veliku količinu podataka sada moguće lakše obraditi, pa su duboke neuronske mreže (eng. *Deep Neural Networks*) doživele veliku ekspanziju, a sa njima i duboko učenje (eng. *Deep Learning*) i time zavladale poljem veštačke inteligencije i mašinskog učenja.

Jedna od oblasti veštačke inteligencije kojoj je razvoj dubokog učenja doneo najviše napretka je učenje sa uslovljavanjem (eng. *Reinforcement learning)*, čime je omogućeno da se klasični pristup interakcije agenta i okruženja – kretanje uz dobijanje nagrada i kazni podigne na novi nivo i obradom podataka kroz neuronske mreže dođe do optimizovanijih rešenja i unapređenih algoritama.

Polje na kom poslednjih godina veštačka inteligencija pokazuje svoj puni potencijal i ima najveći uticaj jeste tehnologija samovozećih automobila. Kako bi se prikupila ogromna količina podataka potrebna za obučavanje samovozećih automobila potrebno je posebno opremljenim vozilima preći milione kilometara po svim uslovima vožnje, što je vremenski i po pitanju resursa veoma zahtevno. Često su podaci potrebni brže i u drugačijem obliku od onog koji je trenutno dostupan, pa su napredni računarski simulatori vožnje omogućili olakšavanje ovog zadatka. Jedan od njih je i Microsoft-ov „*AirSim*“ – simulator za testiranje i vožnju automobila i kvadkoptera.

Tema ovog rada je kombinacija istraživanja u oblastima navedenim iznad – upotreba algoritama učenja sa uslovljavanjem za obučavanje modela koji upravlja automobilom u okruženju „*AirSim*“ simulatora.

2. POSTOJEĆA REŠENJA

Postoji više rešenja koja su se poslednjih godina bavila problemom učenja sa uslovljavanjem i agenta koji bi ga koristio da vozi automobil u simulatoru. Jedna od najpopularnijih platformi za simulaciju kretanja i trke automobilima je *Torcs* [1], koja je korišćena za razvijanje brojnih autonomnih agenata. Neki od primera algoritama razvijenih u ovom simulatoru su *Monte Carlo tree search* [2], evolucioni algoritmi [3] i *Q-learning* [4]. Sem *Torcs-a*, projekat CARMA je bio inspirisan napretkom *DeepMind*-a sa DQN algoritmom u *Atari* okruženju i rešili su da skaliraju algoritam na okruženje *Vdrift* [5] gde su diskretizovali prostor akcija da bi se prilagodio DQN algoritmu. Korišćenjem ručno napravljene proste funkcije nagrade zajedno sa podacima sa senzora i slika uspeli su da dobiju rezultate koji su nadmašili ručno napravljen i programiran kontroler okruženja u tri kategorije: prosečna nagrada, prosečna brzina i maksimalna brzina.

Jedan *Microsoft*-ov istraživački tim se, u sklopu razvoja simulatora *AirSim* i promocije njegovog ekosistema i mogućnosti u kombinaciji sa treniranjem na *Azure cloud* infrastrukturi, bavio i problemom učenja sa uslovljavanjem i njegove primene na kreiranje agenta za autonomnu vožnju u simulacionom okruženju *Neighborhood*. [6] Oni su razvili model treniran distribuiranim dubokim učenjem sa uslovljavanjem, koji je koristio moć računara u oblaku. Model je zasnovan na *DQN* algoritmu *DeepMind-a*, ali je i koristio prednosti *transfera učenja*, odnosno težine u konvolutivnim slojevima neuronske mreže korišćene unutar *DQN-a* su već bile modifikovane i

---

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kovačević, vanr. prof.

unapređene nadgledanim učenjem, odnosno prethodnom vožnjom agenta po okruženju u *AirSim* simulatoru.

3. METODOLOGIJA

Ova sekcija je zadužena za opisivanje okruženja sa kojim agent vrši interakciju, alata i gotovih algoritama korišćenih u implementaciji. Takođe, biće dat i pristup problemu sa aspekta računarskog vida i napomena o korišćenim istog. Agent je implementiran u programskom jeziku *Python* [3].

3.1. Microsoft *AirSim* simulator

U ovom radu, kako bi se obezbedilo okruženje za treniranje korišćen je *Microsoft AirSim – open-source* simulator za automobile i dronove, baziran na *Unreal Engine*-u. Razvijen je kao više-platformski simulator otvorenog koda, koji podržava i hardversku kontrolu pomoću kontrolera kao što je PX4, za fizički i vizuelno realistične simulacije. Poseduje nekoliko API-ja za dobijanje podataka o stanju simulacije, kao i kontrolu parametara okruženja i načina kretanja i parametara vozila koje se kontroliše. [7] Na *slici 1* prikazano je *Hawaii* okruženje u *AirSim*-u.

Slika 1: *Okruženje Hawaii u AirSim simulatoru*

3.2. *OpenAI baselines* i *Gym*

Neprofitna organizacija *OpenAI* ponudila je kroz otvoreni kod skup visoko kvalitetnih implementacija algoritama za učenje sa uslovljavanjem, koji mogu da služe istraživačima u ovoj oblasti da lakše definišu i identifikuju nove ideje i koji služe kao dobra osnova za dalji razvoj i unapređenje oblasti učenja sa uslovljavanjem. Ovaj projekat, nazvan *baselines*, tj. *osnove*. Kako bi se novo okruženje, kao što je npr. korišćeni i ranije pomenuti *Hawaii* iz *AirSim-a* prilagodilo upotrebi kod gore-navedenih algoritama iz paketa *baselines*, potrebno je oblikovati ga na odgovarajući način, a to se postiže pomoću *gym* skupa alata i pravila. Glavni interfejs *Gym* seta je *Env*, koji predstavlja ujedinjeni interfejs okruženja, a njegove glavne metode, koje treba implementirati pri kreiranju sopstvenog modela okruženja su: *reset()* – resetuje okruženje na početno stanje; *step(action)* – izvršava zadatu akciju u okruženju I vraća sliku stanja, nagradu I sledeće stanje; *render(mode="human")* – iscrtava okruženje, tj. prikazuje sliku okruženja u prozoru. Ključni atribut koji treba zadati je *action_space* – određuje prostor akcija koje agent može da izvrši.

3.3. Pristup problemu iz ugla računarskog vida – detekcija središnje linije puta

Dobra nagrada je glavni pokretač i osnov uspešnosti agenta za učenje sa uslovljavanjem, pa je stoga njena optimizacija ključ dobrih rezultata treniranja. U nekim radovima, kao što je [8], osnov za računanje nagrade, kod automobila koji se autonomno kreću po nekoj stazi u simuliranom okruženju a koriste učenje sa uslovljavanjem kao algoritam koji ih „pogoni", bila je udaljenost vozila od linija na putu, odnosno od središnje linije puta ili krajnje desne, odnosno leve linije koje ograničavaju traku. Ovaj pristup se pokazao kao korektan, međutim veliki problem da se algoritam sa takvom vrstom nagrade upotrebi bilo gde drugde, na drugoj stazi, okruženju ili čak i u realnim uslovima, jeste to što okruženje i staza/put moraju biti mapirani i da se poznaju koordinate svih tačaka na putu, ali i širina puta, ili pak lokacija središnje linije puta, zavisno od pristupa.

Rešenje opisanog problema je upotreba tehnika *računarskog vida* kojima se vrši detekcija središnje linije na putu na slici pribavljenoj sa kamere postavljene na prednjoj strani vozila i onda optimizovala nagrada da se prati detektovana linija. U narednih nekoliko rečenica, biće opisani koraci korišćeni da se dođe do detektovane linije:

1. Primeniti *Gausov blur* filter na sliku pribavljenu sa kamere.
2. Promena modela boja rezultujuće slike iz prethodnog koraka – prebacivanje iz RGB u HSV spektar, kako bi se lakše definisao raspon žute boje, koja čini središnju liniju u traci na putu.
3. „Maskiranje slike" – uklanjanje svih boja koje ne spadaju u definisani raspon datih vrednosti za HSV sprektar.
4. Korišćenje *Canny edge detetection* [9] za detekciju ivica na maskiranoj slici. Primer rezultata nakon ove transformacije dat je na *slici 2*

Slika 2: *Slika puta i okoline nakon maskiranja HSV spektra za žutu i detekcije ivica*

5. Nakon detekcije ivica, konačno se primenjuje klasična *Hafova* transformacija za detekciju linija na slici, kojom se dobija niz x, y koordinata linija detektovanih na slici, odnosno njihove početne i krajnje tačke. Na s*lici 3* dat je rezultat nakon ovog koraka u stazi *Hawaii*, korišćenoj u ovom radu.

Slika 3: *Središnja linija na putu detektovana Hafovom transformacijom*

3.4. Implementacija rešenja

Implementaciju *Gym* okruženja je moguće izvesti tako da podržava i kontinualne i diskretne akcije. Za različite isprobane algoritme, korišćen je drugačiji tip akcija, odnosno kod DQN algoritma je korišćen diskretan skup akcija, a kod *policy gradient* kontinualan. Diskretan skup akcija podrazumeva određen krajnji broj vrednosti za

uglove skretanja u rasponu [-1, 1]. Kod DQN algoritma sa najboljim dosadašnjim rezultatima npr. korišćen je sledeći skup akcija: [-0.65, -0.5, -0.25, -0.1, 0.0, 0.1, 0.25, 0.5, 0.65]. Zbog unapred poznatih detalja o okruženju - stazi *Hawaii*, odnosno da staza ne sadrži raskrsnice i neke isuviše oštre krivine, upotrebljen je skup akcija koji ne sadrži neke ekstremne vrednosti za uglove skretanja kao što su -1 i 1 - kako bi kretanje automobila po stazi bilo uglađeno i bez previše naglih pokreta.

Pored navedenog skupa akcija, po *Gym* specifikaciji, potrebno je implementirati i prostor stanja okruženja. Pod ovim se podrazumeva oblik strukture podataka koja će biti korišćena za predstavljanje okruženja, tip podataka, kao i najveća i najmanja vrednost koja se može javiti.

Implementirano okruženje za prostor stanja koristi preprocesiranu sliku koju dobija iz okruženja, sa kamere koja se nalazi na prednjem braniku automobila i usmerena je ka putu, tako da snima trake u kojima se vozilo kreće i malo stvari u okolini. Pod preprocesiranjem se podrazumeva sečenje slike po širini i dužini na neke eksperimentalno utvrđene idealne dimenzije .

3.5. Računanje nagrade agenta

Računanje nagrade je ključna stvar koja određuje uspešnost treniranja agenta za učenje sa uslovljavanjem. Ono se izvršava u svakom vremenskom koraku, što je kod implementiranog agenta svaki frejm.

Nagrada je kod implementiranog agenta spoj nekoliko komponenti, odnosno sadrži ciljeve koje prilikom vožnje mora da ispuni - da vozilo stigne od tačke A do tačke B, usput se držeći središnje linije puta, prateći put željenom brzinom i izbegavajući sve prepreke na tom putu. Zbir ovih komponenti sačinjava nagradu agenta, a najvažnija je ona koja ima ključnu ulogu u kretanju – praćenje središnje linije puta.

Kako bi se držanje središnje linije puta uračunalo u nagradu, potrebno je odrediti trenutnu udaljenost automobila od nje, a to je učinjeno tako što su detektovane sve linije algoritmom opisanim u *3.3*, nakon čega je određivano najbliže rastojanje između detektovanih linija i trenutne pozicije vozila (koja je uvek na sredini slike). Prilikom detekcije linije, eksperimentalno je utvrđeno da je zbog senki i položaja sunca u nekom trenutku moguće menjanje boja na slici tako da nije moguće inicijalno detektovati središnju liniju jer je zašla u previše tamni spektar žute. Kada bi se ovo desilo, da treniranje ne bi bilo prekinuto odmah i bilo računato da je auto krenuo u „*off-road*“ režim kretanja, odnosno van puta, povećavala bi se osvetljenost slike za određeni stepen, da bi se na taj način probala detekcija linije ponovo. Ukoliko i nakon toga nije bilo detektovane središnje linije, znači da je udaljenost od nje maksimalna i da je automobil krenuo u kretanje van puta, što mu je dozvoljeno određen broj vremenskih koraka.

Nakon što je određeno koja je najbliža detektovana linija trenutnoj liniji kretanja vozila, to rastojanje se podeli najvećim mogućim rastojanjem između dve linije (nešto manje od širine slike) i odradi eksponencijalna funkcija, kako bi se nagrada skalirala u rasponu od 0 do 1.

U formuli *1* dat je način računanja nagrade. Primenom ove tehnike računarskog vida, agent može da se kreće i trenira u okruženju bez bilo kakve ljudske pomoći.

$$R = \begin{cases} e^{-d_{min}} + R_v + R_{cg}, \ za\ d_{min} > -1 \\ x * n, za\ d_{min} = -1\ i\ n < n_{max} \end{cases} \quad (1)$$

Parametri u gore navedenoj formuli označavaju sledeće:

- Parametar *d* je udaljenost od najbliže detektovane linije na putu
- Parametar *n* je broj frejmova (koraka) u kojima je automobil proveo van puta
- Parametar n_{max} je maksimalni broj frejmova dozvoljen agentu da se kreće van puta (tj. kada središnja linija nije detektovana) i zadat je programski
- Parametar x predstavlja nagradu (kaznu) za kretanje agenta van puta
- R_{cg} je parametar koji karakteriše uticaj približavanja cilju, odnosno ukoliko je vozilo u trenutnom koraku izvršavanja bliže cilju nego što je bilo u prethodnom koraku, dobiće nagradu. Ovo je zadato formulom *2*

$$R_{cg} = \begin{cases} R_g, & za\ dist_{curr} < dist_{prev} \\ 0, & za\ dist_{curr} \geq dist_{prev} \end{cases} \quad (2)$$

- Parametar R_s je nagrada, odnosno kazna za kretanje brzinom većom od maksimalne ili manjom od minimalne

Postoje dva specijalna slučaja nagrade, koji omogućavaju završetak epizode treniranja pre dostizanja zadatog broja koraka: R_g kada se dostigne cilj zadat GPS koordinatama i kada se automobil sudari sa nekom preprekom – R_c.

3.6. Primenjeni algoritmi

Algoritmi koji su korišćeni za treniranje agenta učenja sa uslovljavanjem dolaze iz paketa *baselines*[1] i isprobani su algoritmi DQN i unapređene verzije *Policy gradient* algoritma. *DQN* sa svojim poboljšanjima, kao i *Policy gradient*. S obzirom na to da je DQN agent ranije opisivan i u ovom slučaju pokazao bolje rezultate u *tabeli* 1 će biti predstavljeni parametri treniranja za DQN algoritam, zajedno sa kratkim opisom i vrednošću korišćenom u verziji algoritma koji je pokazao najbolje rezultate.

Tabela 1: *Parametri DQN algoritma*

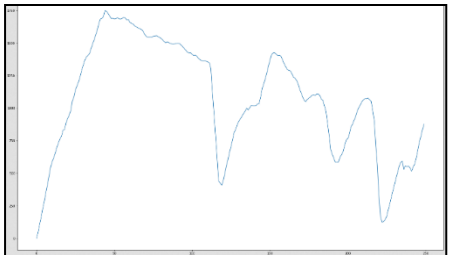
Parametar	Značenje	Korišćena vrednost
network	neuronska mreža koja se koristi kao aproksimator funkcija. Može biti *cnn, conv_only* ili *mpl*	*cnn*
lr	stopa učenja za Adam optimizator	0.0011
buffer_size	veličina bafera za reprodukciju	75000
exp_fraq	*exploration-exploitation rate*	0.1
exploration_final_eps	konačna vrednost verovatnoće nasumičnog odabira akcija	0.015

4. EKSPERIMENTALNI REZULTATI I DISKUSIJA

Zbog kompleksnosti problema i mnogo parametara prisutnih na slici, treniranje agenta za upravljanje auto-

[1] https://github.com/openai/baselines.

mobilom u *AirSim*-u sa primenjenim tehnikama i implementacijama opisanim u prethodnim poglavljima, trajalo je i po dvadesetak sati, da bi se dobio model koji stiže do cilja i pokazuje solidnu voznu dinamiku pritom, držeći se glavne vodilje – držanje srednje linije puta. Na *slici* 4 prikazana je oscilacija, rast i ponašanje nagrade za vreme treniranja, na primeru od 300 000 vremenskih koraka.



Slika 4: *Nagrada po broju vremenskih koraka*

Kao što možemo da vidimo sa slike, nagrada stabilno raste, kako prolazi vreme, dok ne dođe do neke maksimalne vrednosti od ~1750, kada otprilike agent stiže blizu cilja kretanja, pa bi naredni koraci pri treniranju i poboljšavanju nagrade trebalo da budu da se kreće što brže i što više drži središnje linije. Međutim, vidimo da kasnije nagrada opada, čemu je najverovatniji uzrok promena parametara okoline, kao što su pad nivoa osvetljenosti od sunca i promena vremena u danu, koji je u simulatoru nešto sporiji nego u realnom vremenu, ali opet jedan od glavnih faktora realističnosti *AirSim-a*. Pošto dolazi do promene osvetljenja i boja u okolini i na putu, agent više ne može da detektuje središnju liniju i počinje loše da se kreće i sudara sa okolinom, tako da nagrada drastično pada. Uprkos tome što nagrada vremenom počne da opada, najbolji model se kroz *callback* ipak čuva i moguće je proveriti njegove performanse. Snimak procesa treniranja, kao i ponašanja i vožnje nekih od treniranih modela moguće je naći na sledećem linku:
https://drive.google.com/open?id=1pFkB9v0DRPdPfEfH6lZwDcxU2DPV1gKg
Osmišljavanje nagrade izložene u ovom radu je dobra početna tačka, ali poboljšanja bi moglo da bude i sa aspekta kretanja automobila u traci, da se detektuju i linije na stazi koje ograničavaju traku sa leve i desne strane i da se vozilo, umesto duž središnje linije kreće u svojoj traci pravilno. Ovo je i bila inicijalna namera, međutim zbog nedostatka dobrih staza i besplatnih okruženja za *AirSim* simulator, *Hawaii* je ostala jedina sa adekvatnim putnim oznakama, koje zadovoljavaju inicijalnu ideju.

5. ZAKLJUČAK

Oblast mašinskog učenja sa uslovljavanjem, zahvaljujući sve boljem hardveru na kom se mogu obučavati algoritmi koji obrađuju veliku količinu podataka visokih dimenzija i napretku u simulatorima koji sve više elemenata realnog sveta prenose u virtuelni, ima veliku mogućnost napredovanja i uticaja na svet oko nas. U ovom radu dat je kratak pregled oblasti i njenih najvažnijih modernih algoritama, koji su na kraju i primenjeni na rastuću i sve značajniju oblast autonomne vožnje.

Što se tiče samog agenta razvijenog i predstavljenog u ovom radu, postoji mnogo prostora za poboljšanja i unapređenja, pre svega u vidu smanjenja uticaja svetlosti i promene doba dana na detektovanu liniju, pa samim tim i na nagradu i performanse modela. Druga stvar koju bi trebalo poboljšati je i neujednačeno kretanje agenta koji često „oscilira“ levo-desno oko centralne linije, na šta verovatno utiče diskretizovani skup akcija, koji bi moglo povećati ili staviti da bude kontinualan, ali i primeniti druge algoritme kao što je *DDPG* i sl.
Bez obzira na to što počeci učenja sa uslovljavanjem dosežu do ranih 60-ih godina prošlog veka, skorašnji napredak na polju dubokog mašinskog učenja i razvitak hardvera omogućili su procvat ove oblasti, koja pretenduje da u skorijoj budućnosti utaba put ka generalnoj veštačkoj inteligenciji, više nego oblasti nadgledanog i nenadgledanog učenja, ukoliko se dovoljno istraživačkog napora uloži u to.

6. LITERATURA

[1] Loaiacono i Cardamone, „Simulated car racing championship: Competition software manual,“ 2013.

[2] F. J, F. N, Vielwerth i T. J, „ Monte-Carlo Tree Search for Simulated Car Racing,“ 2015.

[3] Koutnik, Cuccu i Schmidhuber, „Evolving large-scale neural networks for vision-based reinforcement learning“.

[4] Loiacono, Prete, L. P. i C. L, „Learning to overtake in torcs using simple reinforcement leraning“.

[5] M. V. A. N, „Carma: A deep reinforcement learning approach to autonomous driving“.

[6] M. Spryn, S. Aditya i P. Dhawal. [Na mreži]. Available: https://github.com/microsoft/AutonomousDrivingCookbook/tree/master/DistributedRL.

[7] Microsoft. [Na mreži]. Available: https://microsoft.github.io/AirSim/docs/apis/.

[8] K. M i K. S, „Autonomous vehicle control via deep reinforcement learning,“ Master's thesis, 2017.

[9] R. Szelinski, Computer Vision: Algorithms and Applications, Springer, 2011.

Kratka biografija:

Miloš Mladenović je rođen 24.05.1994. u Gnjilanu, Republika Srbija. Osnovnu školu „Desanka Maksimović“ završio je 2009. godine u Kosovskoj Kamenici. Gimnaziju u istom gradu završava 2013. godine i upisuje osnovne akademske studije Elektronskom fakultetu u Nišu. Zvanje diplomirani inženjer elektrotehnike i računarstva stiče 2017. godine, sa prosečnom ocenom 9.71, uz specijalizaciju računarske nauke i informatika. Nakon toga, iste godine, upisuje master akademske studije na Fakultetu tehničkih nauka u Novom Sadu, odsek Elektrotehnika i računarstvo, smer Računarstvo i automatika, modul Inteligentni sistemi. Položio je sve ispite predviđene planom i programom master studija uz prosečnu ocenu 9.71.