Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.8
DOI: https://doi.org/10.24867/06BE40Marjanovic

ПРИМЕНА ДУБОКОГ УЧЕЊА УСЛОВЉАВАЊЕМ НА САМОВОЗЕЋИ АУТОМОБИЛ У СИМУЛИРАНОМ ОКРУЖЕЊУ *TORCS*

USING DEEP REINFORCEMENT LEARNING TO TRAIN AN AUTONOMOUS DRIVING AGENT IN A SIMULATED ENVIRONMENT *TORCS*

Ђорђе Марјановић, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – *У овом раду описан је систем у којем се дубоко учење условљавањем примењује на аутономно возилоу симулираном окружењу. Агент је трениран помоћу Deep deterministic policy gradient (DDPG) алгоритма, а окружење представља 3D тркачка видео игрaд TORCS. Након више од 200 епизода обучавања агент је успео да заврши цео круг без скретања ван стазе. DDPG алгоритам се показао веома успешно у окружењима са континуалним акцијама.*

Кључне речи: *дубоко учење условљавањем, DDPG, аутономна возила*

Abstract – *This paper presents an application of deep reinforcement learning for self-driving car in simulated environment. Agent is trained using Deep deterministic policy gradient (DDPG) algorithm and environment is 3D racing video game TORCS. After more than 200 episodes of training agent was able to finish a full lap without turning of the road. The results shot that the DDPG algorithm is very successfull in environments with continuous action space.*

Keywords: *deep reinforcement learning, DDPG, self-driving car*

1. УВОД

Значајан напредак машинког учења и дубоких неуронских мрежа (енг. *Deep Neural Networks*) омогућио је развој система критичних за безбедност као што су аутономни аутомобили. Скорашњи резултати показују да су самовозећи аутомобили постали веома ефикасни у пракси и већ прешли милионе километара без икакве људске контроле [1]. Двадесет америчких држава укључујући Калифорнију, Тексас и Њу Јорк усвојиле су закон који омогућава тестирање и развој аутономних возила [2]. Аутономна возила обећавају велику корист за човечанство, као што је значајно смањење повреда и смртних случајева у саобраћајним несрећама и ефикасније коришћење саобраћаја у сврху смањења загађености ваздуха и смањењем трошкова. Међутим, у овом тренутку ова технологија је још увек у развоју.

У овом раду описано је дубоко учење условљавањем (енг. *Deep Reinforcement Learning*) и његова примена на решавање проблема самовозећег аутомобила у симулираном окружењу. Као окружење користи се видео игрица *TORCS*. Овај систем има клијент-сервер структуру, при чему серверски део обухвата окружење, а део система који помоћу алгоритма контролише кретање аутомобила представља клијента. Допринос овог рада је имплементација *Deep Deterministic Policy Gradient* алгоритма који у овом окружењу са континуалним акцијама одређује угао скретања аутомобила. Резултати су показали да обучавањем алгоритам успешно контролише кретање аутомобила не дозвољавајући скретање аутомобила изван ивица стазе.

Рад је подељен у шест секција. У наредној секцији дат је преглед постојећих решења. У трећој секцији дате су теоријске основу. Четврта секција опису детаље и начин имплементације система. У петој секцији приказани су резултати, а у шестој изложен је закључак овог рада.

2. СРОДНИ РАДОВИ

Једна од првих успешних демонстрација аутономног возила датира од 1986. године [3]. У то време, симулиране слике пута коришћене су за обуку трослојне неуронске мреже чија је излазна вредност правац који аутомобил треба да прати на путу. Од тада, истражује се широк спектар аспеката аутономне вожње укључујући перцепцију, локализацију, планирање и контролу. Људи су постепено почели да се фокусирају на реалније урбане средине са неизвесним саобраћајем, што је и даље важан циљ. У пројекту *ALVINN* [4] 1989. год. коришћена је потпуно повезана неуронска мрежа (један скривени слој са 29 неурона) за предвиђање управљачких команди за аутомобил. Пројекат *DAVE* [5] из 2004. је учио на основу људске вожње. Инспирисан *DAVE* пројектом, *NVIDIA* тим је тренирао велику конволутивну неуронску мрежу мапирањем слика добијених из вожње правог аутомобила на команде за управљање аутономног возила – *behavioral cloning* [6].

Учење условљавањем (енг. *Reinforcement learning*) такође налази примену у аутономној вожњи [7]. Учење условљавањем у почетку није било успешно у аутомобилским апликацијама, међутим скорашњи радови показују способност алгоритама учења условљавањем да стекну представу о окружењу у

НАПОМЕНА:
Овај рад је проистекао из мастер рада чији ментор је био др Александар Ковачевић, проф.

коме оперирају. *Google DeepMind* је ово демонстрирао у пројектима где је агент научио да игра игре као што су *Atari* и *Go* [8].

Захваљујући најновијим достигнућима вештачке интелигенције, *Automotive Driver Asistance Systems (ADAS)* системи остварују брз напредак у протеклих неколико година. Тренутно, бројне истраживачке групе и компаније напорно раде на довођењу ове технологије на тржиште. Један од лидера је *Waymo*, Гуглов пројекат аутомобила без возача. Од настанка 2009. године прешао је преко 4 милиона километара на јавним путевима. *Uber* врши тестирања и планира да опреми 75.000 самовозећих аутомобила у 13 америчких градова до 2022. године. *Tesla* је лансирао свој аутопилот 2014. и до краја 2019. године очекују демонстрацију потпуно самосталне вожње.

3. ТЕОРИЈСКЕ ОСНОВЕ

Учење условљавање (енг. *Reinforcement learning*) је грана машинског учења која се бави секвенцијалним одлучивањем. Проблем учења условљавањем састоји се од агента (енг. *agent*) и окружења (енг. *environment*). Агент у окружењу извршава акције из дозвољеног скупа акција і добија одговор окружења у виду награде (енг. *reward*). Награда, позитивна или негативна вредност, је начин да се агенту допусти да зна колико је била добра акција у том тренутку. Задатак агента је да научи оптималну политику – мапирање из стања (енг. *state*) у акцију, које максимизује очекивану суму награде.

3.1. *Q-Learning*

Q-Learning алгоритам је један од најпопуларнијих техника учења условљавањем, заснива се на учењу вредности стања. Циљ овог алгоритма је да агент научи оптималну функцију $\mathrm{Q(s,a)}$, при чему *Q* функција рачуна квалитет, односно *Q* вредност задатог пара (стање, акција). Ова функција представља итеративни поступак апроксимације, при чему се на сваком кораку ажурира тренутна процена *Q* вредности коришћењем процена оптималне вредности у будућности *(temporal difference)* [9].

3.2. *Policy gradient* методе

Методе градијента политике су приступ у учењу условљавањем јер директно оптимизују кумулативну награду и могу се лако користити са нелинеарним функцијама за апроксимацију као што су неуронске мреже [9].

3.3. *Actor-Critic* методе

Actor-Critic архитектура користи две структуре да оптимизује очекивани резултат. *Actor* и *Critic* раде заједно и посебно се обучавају. *Actor* дефинише политику и његова намена је да генерише акције у складу са тренутном политиком. Функција вредности (*Critic*) делује као критичар пошто процењује политику на основу грешке у временској разлици *(temporal difference error*). *Actor-Critic* методе пружају веома добра својства конвергенције и олакшавају израчунавање акције, посебно у простору са континуалним акцијама. Као функције за апроксимацију веома често се користе неуронске мреже [10].

3.4. *Deep deterministic policy gradient (DDPG)*

Deep deterministic policy gradient (DDPG) [11] је алгоритам учења условљавањем који комбинује *Q-learning*, *policy gradient* и *actor-critic* меотде и веома погодан за проблеме са континуалним или великим бројем акција. Заснива се на *Deterministic Policy Gradient (DPG)* [12] методи и користи неуронске мреже као апроксимационе функције детерминистичке политике.

Q-Learning није ефикасан у окружењима са континуалним акцијама, као решење користи се *Actor-Critic* структура у облику неуронских мрежа. *Actor* функцију $\mu(s|\theta^{\mu})$ која одређује политику која детерминистички за дато стање резултује конкретном акцијом која ће се извршити. *Actor* и *Critic* су међусобно повезани, тако да *Critic* користи тренутно стање и акцију добијену од *Actor*-а за то стање да одреди вредност акције $\mathrm{Q(s,a)}$.

Показало се да директна примена неуронских мрежа у *DPG* алгоритму не доводи до стабилног обучавања. Као решење овог проблема користе се *target networks* које представљају копије *Actor* и *Critic* мрежа. Тежине ових мрежа се ажурирају веома споро тако да прате оригиналне мреже. Рачунају се по формули (1)

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \qquad (1)$$

Мања вредност параметра $\tau \ll 1$ доводи до споријег праћења оригиналних тежина, али увелико побољшава стабилност обучавања. Као још један начин побољшања перформанси, *DDPG* алгоритам користи *experience replay*.

Циљ процеса обучавања *DDPG* алгоритма је да максимизује вредност функције Q(s,a) коју користи actor за модификацију свог параметра θ. *Critic* се обучава користећи *Q-Learning* над узорком података из *replay-buffer*-а при чему се *TD-error* рачуна као сума непосредне награде и излазне вредности *target actor* и *target critic* мрежа.

Actor мрежа учи директно из посматраног простора користећи детерминистички *policy gradient* метод *(DPG)*. Детерминистичка политика може се посматрати као посебан случај стокхастичне политике у чијој расподели вероватноћа постоји једна екстремна не нула вредност за једну акцију. Заправо, доказано је да стокхастична политика $\pi_{\mu_\theta,\sigma}$ је еквивалентна детерминистичкој политици μ_θ где је параметар $= 0$.

Применивши ову теорему у методи *policy gradient* долази се до закључка да је стокхастични градијент политике еквивалентан детерминистичком градијенту политике. Пошто политика μ није зависна од акције, за обучавање *Actor* мреже довољан је градијент резлутата *Critic* мреже помножен градијентом резултата *Actor* мреже над узорком података из *replay-buffer*-а [11][12].

Главни изазов учења у окружењима са континуалним акцијама представља експлорација. Предност *off-policy* алгоритама, као што је *DDPG*, је у томе што се проблем експлорације може третирати независно од учења алгоритма. Овај проблем решава се додавањем шума (енг. *noise*) $\mathcal{N}$ на политику *actor*-а, тј. на саму акцију (2).

$$\mu'(s) = \mu(s_t|\theta^{\mu}) + \mathcal{N} \qquad (2)$$

4. ИМПЛЕМЕНТАЦИЈА

Систем је структуиран као клијент-сервер апликација, при чему *TORCS*[1] видео игра представља серверски део, а за клијентски део користи се *SnakeOil* [2] библиотека помоћу које се контролише аутомобил. Ова библиотека омогућава добављање 21 параметра (сензора) аутомобила који описују стање агента, опсервацију, а то су:

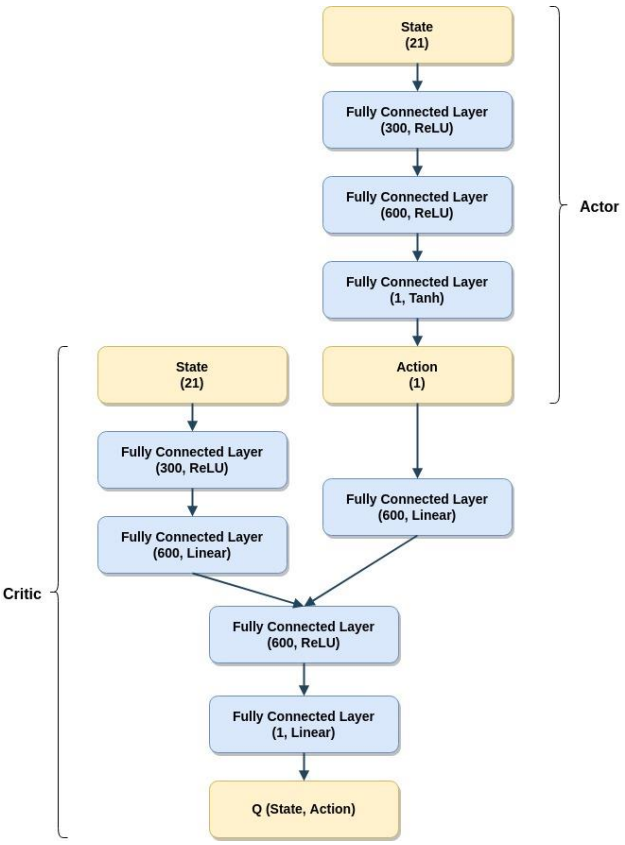
- ***angle*** [-π, π](rad) – угао између правца аутомобила и правца стазе
- ***track*** [0, 200](m) – низ од 19 сензора чије вредности представљају различита растојања аутомобила од ивица стазе
- ***trackPos*** (-∞, ∞) - растојање аутомобила и средишта стазе

За примену алгоритама учења условљавањем користи се *Gym-TORCS* [3] емулатор који је потпуно компатибилан са *OpenAI Gym* [4] пакетом алата. Убрзање и кочење аутомобила контролисаће се аутоматски од стране клијента а *DDPG* алгоритам одређује акцију агента – **угао скретања** у вредности [-1, 1] при чему вредност -1 представља максимално скретање удесно, а 1 максимално скретање улево.

У решавању проблема учења условљавањем веома је битно добро дефинисати функцију награде како би учење било ефикасно. У овом пројекту циљ је да аутомобил научи да се креће на стази, да не стоји и да не скреће ван стазе. Према томе дефинише се функција награде која максимизује брзину аутомобила у правцу стазе, минимизује попречну брзину и додељује негативну вредност на већу удаљеност од средишта стазе. За ефикасније и брже обучавање додат је услов да при скретању ван стазе агент добија награду -200 и сигнал за прекид епизоде.

Actor и *Critic* конструисани су помоћу вишеслојних неуронских мрежа. Улазни податак у actor мрежу је стање агента у облику вектора са 21 елементом, а излаз је акција агента, тј угао скретања дефинисан активационом фунцкијом хиперболични тангенс која ограничава резултат на опсег [-1, 1].

Critic неуронска мрежа као улаз прима два податка: стање агента и акцију агента за то стање које производи *Actor* мрежа. Архитектура *Actor* и *Critic* мрежа са активационим функцијама и бројем неурона приказана је на слици 4.1.



Slika 4.1. *Actor-Critic* архитектура

Структура *Actor* и *Critic* мрежа и избор хиперпараметара *DDPG* алгоритма формирани су по моделу описаном у раду [11] где је показано да алгоритам са истим овим параметрима примењен на различита окружења даје веома добре резултате. Комплетан код овог пројекта, документација и упутства налазе се на *GitHub* репозиторијуму[5].

5. РЕЗУЛТАТИ

Циљ обучавања агента је да оствари што већу кумулативну награду, односно да се креће стазом у што дужем временском периоду. Скретање ван стазе сматра се лошом акцијом при чему агент добија негативну награду -200 и сигнал за завршетак епизоде.

Након више од 200 епизода обучавања аутомобил је успео да, без скретања ван стазе, константно вози више од два круга. У овој епизоди агент је извршио више од 6.000 акција и остварио највећу кумулативну награду до тог тренутка у вредности преко 130.000. Процес обучавања алгоритма, посматрајући кроз кумулативну награду по епизоди приказан је на графику 5.1.

[1] http://torcs.sourceforge.net/
[2] http://xed.ch/project/snakeoil/
[3] https://github.com/ugo-nama-kun/gym_torcs
[4] https://gym.openai.com/

[5] https://github.com/djo10/deep-rl-ddpg-self-driving-car

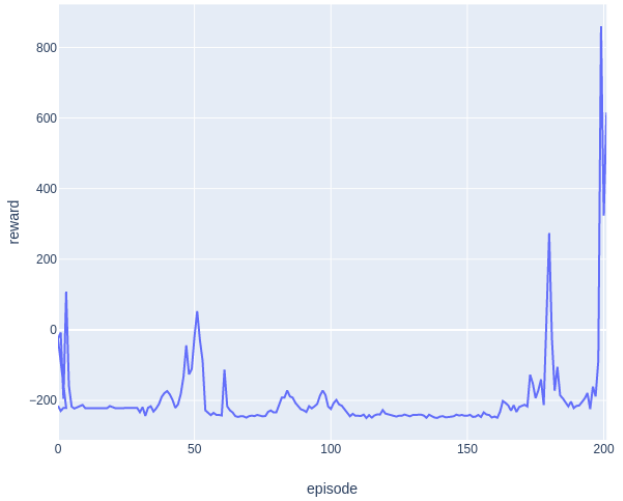


График 5.1. Кумулативна награда по епизоди

На основу графикона може се приметити да агент у раној фази обучавања извршава лоше акције које доводе до негативне кумулативне награде. Пошто алгоритам учења условљавањем учи на основу прикупљања искуства, после 200 епизода агент постаје способан да извршава исправне акције и максимизује кумулативну награду. Процес обучавања *DDPG* алгоритма у *TORCS* видео игри забележен је видео снимком[6].

6. ЗАКЉУЧАК

У раду је представљен је самовозећи аутомобил у симулираном окружењу применом учења условљавањем. За управљање агента коришћен је *Deep Deterministic Policy Gradient (DDPG)* алгоритам, а тркачка видео игра *TORCS* је представљала окружење. Резултати су показали да је овај алгоритам веома ефикасан у окружењима са континуалним акцијама и његова примена у контроли кретања аутомобила показала се веома успешном. Предност алгоритама учења условљавањем је да константно могу да се усавршавају и тиме достигну, а чак и надмаше резултате постигнуте од стране човека.

Како су ове технологије и аутономни аутомобили још увек у развоју, овај пројекат има доста простора за унапређивање и усавршавање. Пре свега, може се модификовати алгоритам да поред скретања контролише брзину и кочење аутомобила. Већина савремених решења за самовозеће аутомобиле користе камере, јер камере представљају најбољу врсту сензора. Употреба конволутивних неуронских мрежа при обради стања агента (слике) представља корак ближе реалним системима аутономних возила. Такође у систем се може имплементирати посебан модул за препознавање препрека као што су пешаци, остали аутомобили у саобраћају, знакови, семафори итд. Заправо, главни циљ оваквих система је усавршавање окружења и алгоритма да обучавање у симулираном систему може успешно да се искористи у реалном свету аутономних возила.

[6] https://www.youtube.com/watch?v=XyvOfroVIwg

7. LITERATURA

[1] 2017. The Numbers Don’t Lie: Self-Driving Cars Are Getting Good. https://www.wired.com/2017/02/california-dmv-autonomouscardisengagement. (2017).
[2] 2017. Autonomous Vehicles Enacted Legislation. http://www.ncsl.org/research/transportation/autonomous-vehiclesselfdriving-vehicles-enacted-legislation. Aspx. (2017).
[3] Takeo Kanade, Chuck Thorpe, and William Whittaker. Autonomous land vehicle project at cmu. In Proceedings of the 1986 ACM fourteenth annual conference on Computer science, pages 71–80. ACM, 1986. 1.1
[4] Pomerleau, D. A. Alvinn, an autonomous land vehicle in a neural network. Technical report, Carnegie Mellon University, Computer Science Department, 1989
[5] Net-Scale Technologies. Autonomous off-road vehicle control using end-to-end learning. Technical report, 2004. Available at: http://netscale.com/ doc/net-scale-dave-report.pdf. [Accessed 17 March 2017]
[6] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J. et al. 2016. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316
[7] A. El Sallab, M. Abdou, E. Perot, and S. Yogamani. Deep reinforcement learning framework for autonomous driving. Autonomous Vehicles and Machines, Electronic Imaging, 2017
[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013
[9] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018
[10] V. R. Konda and J. N. Tsitsiklis, “On Actor-Critic Algorithms,” SIAM Journal on Control and Optimization, vol. 42, pp. 1143–1166, Jan 2003.
[11] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” arXiv preprint arXiv:1509.02971, pp. 1–14, 2015.
[12] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, “Deterministic Policy Gradient Algorithms,” Proceedings of the 31st International Conference on Machine Learning (ICML-14), pp. 387– 395, 2014

Кратка биографија:

Ђорђе Марјановић рођен је 16.3.1994. године у Ужицу. Основне студије уписао 2013. године на Факултету техничких наука, смер Софтверско инжењерство и информационе технологије. Основне студије завршио 2017. Након чега уписује мастер студије, смер Интелигентни системи.