Autonóm járműirányítási feladatok megoldása „FastRL” megerősítéses tanulási algoritmusok segítségével

Szőke László*, Aradi Szilárd **

**Robert Bosch Kft., Budapest,*
e-mail: laszlo.szoke@hu.bosch.com.
***Budapesti Műszaki és Gazdaságtudományi Egyetem,*
e-mail: aradi.szilard@kjk.bme.hu

Absztrakt: Ebben a cikkben a megerősítéses tanulás egyik legfrissebb eredményén alapuló metodológiát és egy autonóm járműirányításban alkalmazható ágens mutatunk be. Először az alapvető megerősítéses tanulás (RL) fogalmait tárgyaljuk, folytatva a Markov Döntési folyamatok (MDP) rövid magyarázatával. Továbbá megvizsgáljuk a Successor Features (SF) rendelkezésre álló szakirodalmát és felhasználásának legújabb eredményeit. Motivációnk az, hogy egy komplex és/vagy egymásnak ellentmondó követelményeket tartalmazó feladatot a jutalom felbontásával kezeljük, mivel a vezetés közbeni összetett feladatok sikertelen tanítást okozhatnak és ezért kihívást is jelenthetnek. Az alkalmazott módszertan elmagyarázása és működésének bemutatása után a legkorszerűbb ötleteket vizsgáljuk meg és bemutatjuk a alkalmazását a járműirányítás területén. A FastRL autonóm járműirányítási területen történő alkalmazhatóságának bizonyítása mellett egy fejlett tanító környezetet is bemutatunk, ami az ágensek szimuláción keresztüli tanítására használható. Ezenkívül a cikk részletezi, hogyan lehet az SF-eket az autópálya-vezetési szcenáriókhoz igazítani, és mi a titka annak, hogy képes növelni az RL ágensek teljesítményét. Annak érdekében, hogy hitelesen megvizsgálhassunk néhány javasolt problémát, környezetként nagy megbízhatóságú forgalomszimulátort (SUMO) alkalmaztunk, és különböző tanításokat végeztünk változatos szituációkban. Eredményeink azt sugallják, hogy a tanult készségek segíthetnek az összetett jutalmazási problémákban, és változó jutalmazási rendszerekben, így az alkalmazott ágensek gyorsan tudnak alkalmazkodni az új feladatokhoz. Az egyetlen dolog, amit meg kell találni, az az SF-ek helyes felbontása és kiválasztása.

1. BEVEZETŐ

Az utóbbi időben minden életterület egyre nagyobb figyelmet fordít a mesterséges intelligencia (AI) hatalmas világára. Az AI lassan beépül a mindennapi életünkbe. Megjelenik az utcán, az otthonainkban, az okoseszközeinkben és a minket körülvevő környezetünkben. Emellett az autonóm járművek (AV) kutatása gyorsan fejlődő terület, amelyre mind akadémiai mind pedig ipari szektorban hatalmas erőforrásokat fordítanak. A gépjárművek AI alkalmazásainak fő hajtóereje a közlekedésbiztonság és az utaskényelem. (Rajasekhar & Jaswal, 2015; Y. Wang et al., 2021). Napjainkban nagy kihívást jelent a kutatók számára az önvezetés megoldása, amit valószínűleg nem lehet megoldani kizárólag klasszikus, szabályalapú irányítással. A nagyvállalatok legutóbbi ígéretes eredményei azonban arra engednek következtetni, hogy az önvezető autók a közeljövőben valósággá válhatnak. A fejlesztések többsége felügyelt tanulást használ, így teljesítményük korlátozott a rendelkezésre álló tanulási adatok és a címkézési költségek korlátozása miatt (Favarò et al., 2018; Navarro et al., 2017; Tuncali et al., 2018).

Ebben a munkában arra törekszünk, hogy kihasználjuk a megerősítéses tanulás (RL) előnyeit egy AV irányításához autópályás szituációkban. Továbbá, megvizsgáljuk a legújabb algoritmusok alkalmazhatóságát, mint például a Successor Features (SF), a General Policy Improvement és Evaluation.

Célunk, hogy a Google DeepMind FastRL (Barreto et al., 2020) nevű új algoritmusa alapján egy olyan ágenst alkossunk meg, amely autópályás környezetben képes bizonyos járműirányítási funkciók ellátására.

Javaslatot teszünk az SF-ek lehetséges meghatározására autópályán előforduló forgalmi helyezetekben. További értéket adunk a létező munkákhoz a tanítási környezet átalakításával, amellyel lehetővé tesszük az SF-ek használatát az ágensek tanulása során. Ezenkívül összehasonlítjuk algoritmusunkat a Q-Learninggel, és kiértékeljük, hogy mennyire illeszkedik a módszer az autonóm járműirányítási területhez.

Először egy rövid szakirodalmi áttekintést (2. fejezet) adunk a kapcsolódó kutatásokról, amelyet a Markov Döntési Folyamatok, SF-ek és használatuk ismertetése követ, valamint a General Policy Update (3. fejezet). A 4.1 szakaszban bemutatjuk tanító környezetünket, a vizsgálatainkat, és tapasztalatainkat (4.2 szakasz). Munkánk befejezéseként értékeljük az eredményeket (5. fejezet), és levonjuk a következtetéseket (6. fejezet).

2. IRODALMI ÁTTEKINTÉS

2.1 A megerősítéses tanulás a járműirányításban

Az RL már néhány éve jelen van a szakirodalomban, de még a mesterséges intelligencia és a gépi tanulás fiatal területeihez soroljuk. (Sutton & Barto, 2018) bemutatja az RL alapjait, az alkalmazott matematikai megközelítést, néhány lehetséges megoldást és praktikát a különböző feladatok megfogalmazásához és megoldásához.

Az RL algoritmusait egyre gyakrabban használják a járműirányításban, hiszen alkalmazásánál nincs szükség előre felcímkézett adatokra illetve a tanulási folyamatok konkrét interakciókon alapszanak, ami egy ilyen rendszernél elengedhetetlen.

(Aradi, 2020) egy átfogó körképet mutat a mély RL hierarchikus mozgástervezési problémáiról és azoknak publikált megoldásáról. A témában többen fejlesztenek és mutatnak be különböző algoritmusokat és megközelítéseket az autonóm járművek területén. Többek között komplex városi szituációkat oldanak meg kettős mély Q-hálózattal (DDQN), Soft Actor-Critic (SAC) algoritmussal és TD3 módszerrel (Chen et al., 2019), változó autópálya-szcenáriókat vizsgálnak módosított Deep-Q-hálózatokkal (DQN) (Toromanoff et al., 2019), illetve mély determinisztikus policy gradiens (DDPG) módszert használnak autópályás sávváltáshoz (P. Wang et al., 2019). Az RL által megoldott autonóm irányítási feladatokra további példákért a következő cikkeket ajánljuk: az autó követésére, a sávtartásra, a pálya követésére, valamint besorolásra: (Gu et al., 2017; Kővári et al., 2020; Sallab et al., 2017).

A fent említett munkák azt példázzák, hogyan lehet egy ágenst egy adott környezetben meghatározott jutalommal és céllal tanítani. Ha azonban a jutalomfüggvények (célok) változnak, vagy a környezet változik meg, az ágensek általában nem tudnak jól teljesíteni. Úgy tűnik, hogy az RL egyik legnagyobb hiányossága abból fakad, hogy az ágensek, ellentétben az emberekkel; a nulláról tanulják meg viselkedésüket. Véleményünk szerint erre a problémára egy lehetséges megoldást kínál az SF-ek használata, amelyekről rengeteg munka található a legfrissebb szakirodalomban. Ezek közül néhányra a 2.2 alfejezetben mutatunk rá.

A kulcsfontosságú gondolat az, hogy szétbontjuk az állapotváltozásokat készségekké (feature-ökké). Ezzel egy lépéssel közelebb kerülünk a különböző jutalmazási függvényekkel rendelkező feladatok megoldásához a már elsajátított készségek kombinálásával. Ilyen absztrakciókat használva a szakirodalomban számos munka mutatja meg a képességek (skillek) újra felhasználását és így a megfelelő teljesítmény gyors elérését. A következő alfejezet a témával kapcsolatos fontosabb munkákat mutatja be.

2.2 Successor Features

Dayan említette először a „Successor representation” fogalmat (Dayan, 1993). Kijelenti, hogy a „successors” hasonlósága határozza meg, hogy mennyire megfelelő az állapotok közötti általánosítás. (Kulkarni et al., 2016) bontja fel az értékfüggvényeket két összetevőre, az egyik a jutalom előrejelzése, a másik a „successorok” térképe. (Van Niekerk et al., 2019) cikke, amelyet a International Conference of Machine Learning (ICML) mutattak be, azt állítja, hogy az RL-ben az optimális értékfüggvény összeállítható a regularizált entrópia segítségével. Az első kapcsolódó cikk az SF -ekről (Barreto et al., 2017) újrafeldolgozza Dayan ötletét, és kiterjeszti azt a „General Policy Improvement” (GPI) fogalmával. Ennek köszönhetően a dinamikus programozás policy update műveletének általánosításával nem egy, hanem egy sor policyt vesznek figyelembe a legjobb döntés kiválasztásánál. A módszer RL-be történő zökkenőmentes integrációjának levezetése után a szerzők teljesítménygaranciát is bizonyítanak az új feladatokhoz választott megfelelő döntésekre további tanulás nélkül. Szerencsére nem álltak meg a GPI alkalmazásánál, de egy később megjelent cikkükben kibővítették az ötletet az úgynevezett „General Policy Evaluation” - nel (GPE).

2.3 GPE és GPI

A GPE és a GPI, vagy általában „general policy update”-nek nevezett fogalom a policy-k értékelésének és javításának kiterjesztése, általánosítása, ahol egy policy kiértékelése/javítása helyett egyidőben többet is képesek vagyunk elvégezni.

Amint fentebb említettük, (Barreto et al., 2018) bevezet egy algoritmust, amelyet RL és SF-k esetén használhatunk. A további munka (Borsa et al., 2018) ötvözi az univerzális értékfüggvény approximációk skálázhatóságának előnyeit, az SF-ek azonnali teljesítményét és a GPI erős általánosítását. Ehhez a munkához kapcsolódóan a következő cikk (Barreto et al., 2019) leírja, hogyan lehet a legjobban kombinálni a készségeket az RL-ben.

A GPI-t és GPE-t, SF-ek is használó egy másik hatékony alkalmazást a DeepMind egyik legújabb kiadványa (Hansen et al., 2019) részletezi. A szerzők meghatározzák a Variational Intrinsic Successor featuRes-t (VISR), amely felülmúlja a legkorszerűbb RL modelleket. Először felügyelet nélküli előtanítással tisztán állapotváltozásokból határozzák meg az SF-eket, majd hozzáadnak jutalmakat RL-el. Ez jelentős lépés az általános megerősítéses tanulás (GRL) felé. A fentiek alapján kijelenthető, hogy az SF-ek sok kiaknázható lehetőségek kínálnak. Az SF+GPE+GPI teljes kihasználását (Barreto et al., 2020) cikke írja FastRL néven. Barreto és társai bemutatják, hogyan kell alkalmazni az „oszd meg és uralkodj” szemléletet az RL-re, és az elméletet rengeteg kísérlettel támasztják alá. Mi ebben a munkában a FastRL használhatóságára összpontosítunk az autonóm járművek területén, amelyet a legjobb tudásunk szerint a jelen cikk megfogalmazásakor nem alkalmaznak aktívan az ilyen területeken. (Barreto et al., 2020) ötlete alapján, kidolgoztunk egy ágenst az autópályán történő döntéshozás problémájának kezelésére.. Ez magában foglalja a jutalom definálást és a terminálási feltételek meghatározását is.

3. METODOLÓGIA

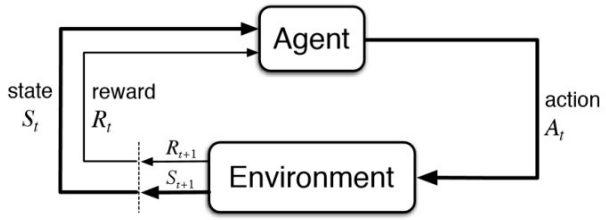
3.1 Megerősítéses tanulás

Az önvezető járművek irányítási és döntéshozási kérdései a kihívások mellett izgalmas lehetőségeket is nyújtanak mesterséges intelligencia kutatói számára. Korábbi munkáink jó példákat mutatnak az RL ágensekre, amelyek egyszerű neurális hálózatokon alapulnak, és autópálya szcenáriókban működnek (Szoke et al., 2020; Szőke et al., 2020). Az alábbiakban áttekintjük a megerősítéses tanulás matematikai alapjait. Általánosságban elmondható, hogy egy RL probléma modellezhető a Markov Döntési folyamattal (Puterman, 1990). Lényegében az MDP-ket rendezett n-esként lehet definiálni,

$$\mathrm{M} \equiv \{\mathrm{S}, \mathrm{A}, \mathrm{P}_{\{\mathrm{S,A}\}}, \mathrm{R}\} \tag{1}$$

ahol, S az állapottér, A az akciótér, $P_{S,A}$ az állapotváltozás valószínűsége, és R a jutalomtér.

Ágensünk az epizód minden egyes diszkrét időpontjában kölcsönhatásba lép a környezettel. Ennek eredményeként minden egyes $\mathrm{t} \in \{1,2,3,4,\ldots,\mathrm{n}\}$ értéken figyeli az őt körülvevő környezetet $\mathrm{s} \in \mathrm{S}$ állapotban, majd az $a \in A$ akciót választja, hogy maximalizálja a várt jutalmat $r \in R$. E folyamat során a környezet s állapotból s' állapotba változik. Ezt a dinamikát a $\mathrm{P}_{S,A}$ valószínűség adja meg. A folyamatot az 1. ábra mutatja be. A folyamatábra egy lépést szemléltet a környezetben, de hosszú távon az ágens megpróbálja megtalálni a jutalomfüggvény maximumát.



1. ábra – A megerősítéses tanulás folyamatábrája

(Barreto et al., 2017) a jutalomfüggvényeket különböző feladatokként azonosítja, így a tanítás célja egy olyan policy megtalálása $\pi: \mathrm{S} \mapsto \mathrm{A}$, amely maximalizálja minden állapot-akciópár értékét:

$$\mathrm{Q}_{\mathrm{r}}^{\pi}(\mathrm{s},\mathrm{a}) \equiv$$

$$\mathrm{E}^{\pi}\left[\sum_{i=1}^{\infty} \gamma^{\mathrm{i}}\, \mathrm{r}\left(\mathrm{S}_{\{\mathrm{t+i}\}}, \mathrm{A}_{\{\mathrm{t+i}\}}, \mathrm{S}_{\{\mathrm{t+i+1}\}} \,\middle|\, \mathrm{S}_{\mathrm{t}} = \mathrm{s}, \mathrm{A}_{\mathrm{t}} = \mathrm{a}\right)\right], \tag{2}$$

ahol $E^{\pi}[\cdot]$ a π policy által javasolt akciók várható értékét jelöli, és $\gamma \in [0,1]$ a visszaterjesztési tényező, amelyet a jövőbeli jutalmak súlyozására használnak.

A (2) használatával folytathatjuk a GPE és a GPI SF-ekkel való megfogalmazását.

3.2 SF, GPE, GPI

Ezeknek a fogalmaknak az alkalmazása azt eredményezi, hogy "az ágens képes egyidejűleg megismerni a bonyolult jutalmazási funkciókat, és további előnyökkel járhat, ha az összetett feladatokat egyszerűbbekre bontja, a feladatok között információcserét és a készségeket újra felhasználja." (Barreto et al., 2020)

Teljes levezetés nélkül a következő kifejezések szükségesek a módszer bevezetéshez: $\phi: S \times A \times S \mapsto R^d$, mint "features" és bármilyen d *dimenziójú* $vektor\ w \in R^d$.

Ezek használatával a jutalom függvény (feladatok) a következőkkel jellemezhetők:

$$r_w(s,a,s') = \phi(s,a,s')^{\top} w, \tag{3}$$

A következő (Barreto et al., 2020) után az SF a következőképpen van definiálva:

$$\psi^{\pi}(s,a) \equiv$$

$$E^{\pi}\left[\sum_{\{i=0\}} \gamma^i \phi\left(S_{\{t+i\}}, A_{\{t+i\}}, S_{\{t+i+1\}}\right) \mid S_t = s, A_t = a\right] \tag{4}$$

Ha a (2) egyenletbe a (3)-at helyettesítjük, a következő egyenletet kaphatjuk:

$$\begin{aligned} \psi^{\pi}(s,a)^{\top}\mathbf{w} &= \mathbb{E}^{\pi}\left[\textstyle\sum_{i=0}^{\infty} \gamma^i \phi\,(S_{t+i}, A_{t+i}, S_{t+i+1})^{\top}\,\mathbf{w} | S_t = s, A_t = a\right] \\ &= \mathbb{E}^{\pi}\left[\textstyle\sum_{i=0}^{\infty} \gamma^i \mathbf{r}_{\mathbf{w}}\,(S_{t+i}, A_{t+i}, S_{t+i+1}) | S_t = s, A_t = a\right] \\ &= Q_{\mathbf{r}_{\mathbf{w}}}^{\pi}(s,a) \equiv Q_{\mathbf{w}}^{\pi}(s,a) \end{aligned} \tag{5}$$

Ez egyszerűsíti az adott policy akció értékfüggvényének kiszámítását a $\psi^{\pi}(s,a)^{\top} \times \mathrm{w}$ belső szorzatra, amely a GPE kiszámításának nagyon hatékony formája. Az (5) egyenlet levezetéséről és bizonyításáról további részleteket a (Barreto et al., 2020) tartalmaz.

A kísérleteinkhez egyetlen fennmaradó még ismeretlen kifejezés a GPI, szintén Barreto és társai után. A π' továbbfejlesztett policy-t a következő egyenlet határozza meg:

$$\pi'(s) \in arg\max_{a \in A} \max_{\pi \in \Pi} Q_r^{\pi}(s,a) \tag{6}$$

Alapvetően minden állapot felett értékeljük az állapot-akciópár értékfüggvényét, és kiválasztjuk a legmagasabb értékű akciót.

A (Barreto et al., 2017) cikkben az 1. tétel azt mutatja, hogy ez a számolás a GPI helyes formája, és így felhasználható munkánk során.

Miután megvizsgáltuk a módszer fő előfeltételeit, a következő részben gyorsan áttekintjük a környezetet, és megadjuk az alkalmazott neurális hálózatokat és azok tanítási beállításait is részletezzük.

4. A JAVASOLT ÁGENS

4.1 Környezeti beállítások

A járműirányításban többféle szimulátor áll rendelkezésre, ilyenek például a Carla, a PreSCAN vagy a CarSim szoftverek. Áttekintve a funkciókat és a szükséges paramétereket úgy döntöttünk, hogy az autópálya-szcenárióinkat a Simulation of Urban MObility (SUMO) szoftverrel készítjük el (Lopez et al., 2018), mert a SUMO egy nagy pontosságú forgalomszimulátor, rengeteg funkcióval és személyre szabható forgalmi forgatókönyvekkel. A szoftver mélyebb megismeréséhez javasoljuk a SUMO működését részletező dokumentumokat: (Behrisch et al., 2011; Daniel Krajzewicz, Georg Hertkorn, Peter Wagner, 2002).

A környezetünket eredetileg a Python-ban hozták létre az OpenAIGym felület képében, hogy könnyen integrálható legyen a Pytorch-ba. (Szoke et al., 2020) Ezt átalakítottuk, hogy képes legyen bontott, vektorizált jutalomfüggvénnyel működni. Epizódikus RL beállításokkal dolgozunk, ahol a terminál események az ütközés, túl lassú haladás ($v_{ego} < 60\ km/h$ vagy az autópálya elhagyása.

Egy egyenes autópályát alakítottunk ki 1000 méteren és 3 sávon. Az autók véletlenszerűen generálódnak, egyenletesen elosztva a kezdő sávok között (3600 jármű/óra). Az autók mindegyike randomizált viselkedés- és sebességprofillal rendelkezik. Minden autót a SUMO beépített szimulátora irányít, kivéve az EGO járművet, amit az ágensünk vezet. Az EGO-t véletlenszerűen választjuk ki a szimulációban már létező autók közül. A maximális sebesség 50 m/s, az EGO elvárt sebessége pedig 28-43 m/s változik. A környező járművek átlagos sebességét véletlenszerű 18-25 m/s között. Ez biztosítja, hogy az EGO-nak előznie, sávot váltania kell, ha meg akarja tartani az elvárt sebességet. Minden szimuláció 100 ms időlépéssel fut, ahol az ágens dönthet, hogy sávot vált, illetve gyorsít vagy lassít.

Az akciótér 3 oldalirányú és 3 hosszirányú irányításból áll, beleértve a semleges műveleteket is, 9 diszkrét választással dekódolva:

$$a_i \in A\ \{0, 1, \ldots, 7, 8\}$$

$$a_{\{acc\}} \in [-0{,}7,\quad 0,\quad 0{,}3],$$

$$a_{\{steer\}} \in [jobbra, marad, balra]$$

A laterális kontroll csak sávváltási parancsokat tartalmaz mindkét oldalra, amelyek azonnal érvényesülnek. Az ágens állapottere magában foglal olyan információkat, mint például a relatív sebesség és távolság, az elöl és a hátul haladó járművekről mindkét szomszédos sávban. Az EGO kétoldalán azonban 1 vagy 0 jelzi a járművek jelenlétét. További állapottér elemek az EGO sebessége, iránya, az elvárt sebesség és a sáv indexe, így az állapottér egy 18 elemű vektorként reprezentálható. Az állapottér minden értéke elemenként normalizált, és [-1, 1] közé skálázott.

Az SF-ek miatt a környezet támogatja a vektoros jutalmakat a skaláris visszacsatolás helyett. Létrehoztunk 3 különböző jutalmazási funkciót. Az egyik r_1 jelzi a termináló eseményeket, értéke -10 ha bármilyen terminálás bekövetkezik, egyébként 0. r_2 figyelembe veszi az elvárt sebességtől való eltérést, maximális értéke 1. Az $r_3 = 1$ azonnali jutalmat jelent minden sikeres sávváltásért.

Jutalmi függvényeink eltérnek a (Barreto et al., 2020) használatától, mivel a szerzők olyan jutalmakat választanak a gyűjtögető ágensüknek, amellyel meg kell tanulnia a különböző típusú tárgyak különböző jutalmait. Az alapkörnyezetük azonban nem rendelkezik olyan végállapotokkal (termináló eseményekkel), mint a miénk. A mi megoldásunk tartalmaz tehát egy feature-t a terminál eseményekhez és kettőt az azonnali jutalmakhoz.

Megfogalmazásunkban az ágens fő feladata, hogy az autópályán haladjon a terminál események elkerülésével, és maximalizálja a megszerezhető jutalmakat. A kísérleteket a következő szakasz részletezi.

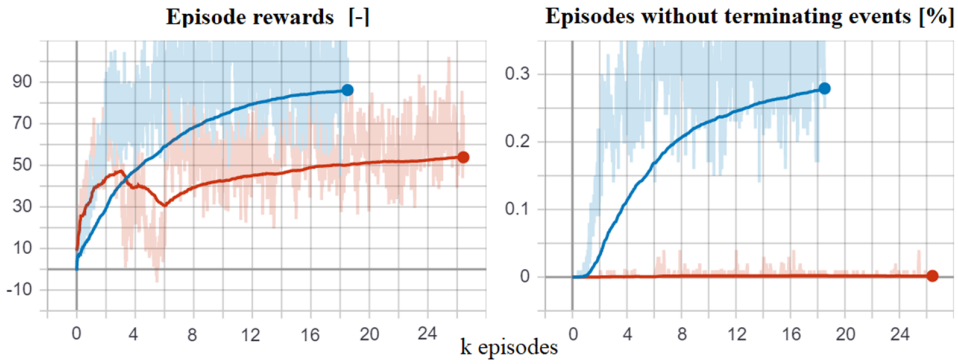
4.2 Elvégzett vizsgálatok

Az SF-ek használhatóságának vizsgálatához a következő kísérletet végeztük el:

Két Pytorch-ban kifejlesztett ágenst definiáltunk. Az *A ágenst* a 3.2 részben leírtak szerint, és a *B ágenst* egy egyszerű Q-Learning módszerrel hoztunk létre. Az *A ágens* esetében az említett jutalmak különböző, lineárisan független policy-k voltak, amelyeket a következő w preferencia vektor értékek indukálnak: $w = [[1,0,0], [0,1,0], [0,0,1]]^T$, a függvények szorzatának eredménye az r_1, r_2 *és* r_3. Az általunk megalkotott „feature”-ök ϕ_i és az általuk generált policy-k $\pi_i \in \Pi, i \in [1,3]$ a következők:

- π_1 a terminál események elkerülésére;
- π_2 felelős azért, hogy az elvárt sebességet a lehető legnagyobb mértékben tartsa, és
- π_3 a sikeres sávváltáshoz, amikor csak lehetséges.

Bár az állandó sávváltás nem egy kívánt viselkedés, de ugyanakkor könnyen észlelhető manővert eredményez, ezért ezt az eredmények fejezetben tovább vizsgáljuk.



2. ábra – Tanulási görbék mindkét ágensnek (A ágens: kék; B ágens: piros)

Az *A ágens* az SF-ek használatával módosított Q-learning metóduson keresztül tanul. Minden SF saját többrétegű megfigyelővel (MLP) rendelkezik, 2 lineáris réteggel. Az MLP-k teljesen előrecsatolt lineáris rétegek, amelyek nem lineáris aktivációs függvényeket tartalmaznak. A rejtett rétegek 128 és 64 neuront tartalmaznak, majd BatchNorm és ReLU aktivációs rétegek követik. Az állapotokat kiértékeljük az összes SF-en, majd a w segítségével előállítjuk a policy-ket, és kiválasztjuk a legjobb értékkel rendelkező akciót. Fontos megjegyezni, hogy egy epizód során véletlenszerűen választunk egy π policy-t a halmazból, és ennek megfelelően cselekszünk az epizód végéig. A visszajátszási memória (replay memory) 2048 elemet tartalmaz. Azt is meg kell jegyezni, hogy terminálás esetén az azonnali jutalmak 0-ra lettek állítva. Például π_3 nem kapott büntetést akkor sem, ha a sávváltás az autópálya elhagyását eredményezte.

Q-learning esetén a jutalmat az $r_q = r_1 + r_2 + r_3$ összetett jutalomfüggvény írja le. Ezzel egy olyan *B ágensre* számítunk a betanítás után, tartja az elvárt sebességet, elkerüli a terminálló eseményeket, miközben folyamatosan sávot vált. A (3) alapján ezt a policy-t a $w = [1,1,1]$ idézi elő az *A ágens* esetében, és várhatóan hasonló viselkedést tapasztalunk majd a betanult ágens ezen vektorral való következtetése során.

A látens tér felépítése hasonló az *A ágenséhez*, és a teljesítmény összehasonlíthatóbbá tétele érdekében szintén egy MLP-t használtunk, de kezdeti rétegén háromszor több neuront kap, mint az SF approximátorok. Így a *B ágens* kétszer annyi paramétert tartalmaz, mint az *A ágens*.

5. EREDMÉNYEK

Mindkét ágens 20 millió állapotváltást és akcióválasztást tapasztalt. Az *A ágens* esetében ez policy-nként körülbelül 6,67 millió átmenetet jelent. Ezt fontos szem előtt tartani a 2. ábrán látható eredmények összehasonlításakor. A hálózat paraméterei minden epizód után frissültek, a tanulási ráta 10^{-4}, a batch mérete 2048 és az ϵ a tanulás során egyről 0-ra csökkent.

Az A ágens tanulása kevesebb epizóddal zárult, ennek oka az, hogy minél több lépést tesz meg az ágens, annál kevesebb epizódra van szükségünk ahhoz, hogy azonos számú állapotváltozást lássunk.

Mivel az *A ágens* véletlenszerű policy-n tanul, csak az epizódok harmadában szeretné elkerülni a terminálló eseményeket (π_1 felel ezért), amit a 2. ábra jól szemléltet. Az *A ágens* az esetek 30%-ában volt képes sikeresen haladni az autópályán. A *B ágensnek* azonban nem sikerült megfelelő viselkedést találnia a tanítás során, amely a bonyolultabb jutalomfüggvénynek tudható be, és feltehetően nem sikerült a függvény komponensekre való bontása. Érdekes, hogy az epizódjutalmak tekintetében az *A ágens* szinte a kezdetektől felülmúlja a *B-t*.

Mindkét ágenst kiértékeltük további feladatokra, melyet a következő táblázat mutat.

1. Táblázat – Az ágensek teljesítménye a különböző feladatokon

		w_1	w_2	w_3	w_4
Ágens A	Jutalom	**-4.7**	**195.4**	**252.7**	**123.5**
	Lépés	242.3	211.9	258.2	73.6
Ágens B	Jutalom	-10	51.83	58	120.7
	Lépés	72.6	72.6	72.6	72.6

A preferencia vektorok a következők: $w_1 = [1,0,0]^T, w_2 = [1,1,0]^T, w_3 = [1,0,1]^T$ és $w_4 = [1,1,1]^T$.

A kiértékelésnél 100 epizód átlagos jutalmát vizsgáltuk meg 4 különböző feladatra, amelyeket mindkét ágens tapasztal. Nyilvánvaló, hogy a Q-learning minden feladaton ugyanazt csinálja, hiszen a tanulás során elsajátított jutalomfüggvényt próbálja maximalizálni, de az egyetlen különbség az elért jutalom, ami az adott feladathoz igazodik. A táblázatban kiemeltük a jobb teljesítményeket.

Megjegyzés: a különböző feladatok jutalmai nem hasonlíthatók össze egymással a különböző kumulánsok miatt, de minden lépés jutalmát megkaphatjuk $r = [r_1, r_2, r_3]$ vektor és az adott w szorzásával. Itt szeretnénk hangsúlyozni, hogy az *A ágens* milyen jól tud megoldani új feladatokat, annak

ellenére, hogy soha nem tanult rajtuk. Ez jól alátámasztja e munka fő célját és létjogosultságát.

A videós kiértékelés során látható volt, hogy a w_4 preferenciát használó *A ágens* jobb döntéseket hoz a vezetés közben, mint az ugyanezen a jutalmon tanult *B ágens*. Néhány helyzet azonban nem volt kezelhető az ágensek számára. Akadt olyan eset például, ahol mindhárom sáv el volt foglalva, és a vészfékezés túl későn kezdődött, ezért az ütközés nem volt elkerülhető. Ez betudható annak, hogy az ágensek úgy lettek tervezve, hogy az állapotok időbeli összefüggése elhanyagolásra került., így a vészfékezés kezdetének pontos meghatározása nem volt lehetséges. Ugyanakkor bátran feltételezhetjük, hogy a w_4 feladatra vonatkozó kiegészítő tanítás növelné az A ágens teljesítményét.

6. KONKLÚZIÓ

Munkánk során áttekintettük a „Successor Features” irodalmát és a „General Policy Update” elemeit és funkcióját. (GPE+GPI). Bemutattuk a szimulációs keret módosítását, amelyet kiegészítettük azzal a lehetőséggel, amely támogatja az RL ágensek epizodikus tanítását SF függvények használatával. A legújabb FastRL kutatási eredmények alapján kidolgoztunk egy járműirányítási problémákra megoldására alkalmas ágenst, és javasoltuk az autópálya-szcenáriók esetén alkalmazható SF-ek egy lehetséges definícióját.

Ezenkívül egy vizsgálat során rávilágítottunk az SF-ek használatának előnyeire, és arra, hogy ez hogyan használható az autonóm járművek irányítási feladataiban is. Az eredmények megegyeznek azon hipotézisünkkel, hogy az SF-ek ezen a területen is előnyökkel alkalmazhatók. Kiegészítő megjegyzésként elismerjük, hogy a Q-learning nem tudta megoldani a többcélú jutalmazás problémáját a vizsgált tanulási kísérleteinkben, de az SF-ek jobban teljesítenek, ezért a jövőbeni munkánk az SF-ek alkalmazási irányelveinek javítására fog összpontosítani.

Szintén ígéretes terület lehet a folytonos és összetettebb állapottérrel rendelkező SF -ek alkalmazása. A kérdés azonban továbbra is nyitva marad: "Hogyan kell megtervezni a jutalomfüggvényeket és az SF-eket a legnagyobb mobilitás és rugalmasság elérése érdekében az önvezető járműirányítási feladatokban?". Ez a kérdés rámutat további kutatási irányainkra, ahol az SF-ek alkalmazásában kívánunk mélyebbre ásni, és tökéletes alapfeladatokat (jutalmakat) határozunk meg. Ezek segítségével az ágensek azonnal képesek a már elsajátított „skill”-ek kombinálásával új feladatokat is jól megoldani. Jövőbeli tervünk az, hogy érzékenységi elemzést végezzünk a forgalmi torlódások különböző szintjeiről (pl. eltérő forgalomi terheltség), és az autópálya-geometria változásairól.

7. KÖSZÖNETNYILVÁNÍTÁS

EFOP-3.6.3-VEKOP-16-2017-00001: Tehetséggondozás és kutatói utánpótlás fejlesztése autonóm járműirányítási technológiák területén - A projekt a Magyar Állam és az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg.

8. HIVATKOZÁSOK

Aradi, S. (2020). *Survey of Deep Reinforcement Learning for Motion Planning of Autonomous Vehicles*. http://arxiv.org/abs/2001.11231

Barreto, A., Borsa, D., Hou, S., Comanici, G., Aygün, E., Hamel, P., Toyama, D., Hunt, J., Mourad, S., Silver, D., & Precup, D. (2019). The Option Keyboard Combining Skills in Reinforcement Learning. In *Advances in Neural Information Processing Systems* (Vol. 32).

Barreto, A., Borsa, D., Quan, J., Schaul, T., Silver, D., Hessel, M., Mankowitz, D., Zídek, A., & Munos, R. (2018). *Transfer in Deep Reinforcement Learning Using Successor Features and Generalised Policy Improvement*.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., Van Hasselt, H., & Silver, D. (2017). Successor Features for Transfer in Reinforcement Learning. In *Advances in Neural Information Processing Systems* (Vol. 30).

Barreto, A., Hou, S., Borsa, D., Silver, D., & Precup, D. (2020). Fast reinforcement learning with generalized policy updates. *Proceedings of the National Academy of Sciences*, *117*(48), 30079–30087. https://doi.org/10.1073/pnas.1907370117

Behrisch, M., Bieker, L., Erdmann, J., & Krajzewicz, D. (2011). *SUMO – Simulation of Urban MObility An Overview*. https://elib.dlr.de/71460/1/SUMO%7B%5C_%7Dsurvey%7B%5C_%7DSIMUL2011.pdf

Borsa, D., Barreto, A., Quan, J., Mankowitz, D., Munos, R., van Hasselt, H., Silver, D., & Schaul, T. (2018). UNIVERSAL SUCCESSOR FEATURES APPROXIMATORS. In *arXiv*.

Chen, J., Yuan, B., & Tomizuka, M. (2019). Model-free Deep Reinforcement Learning for Urban Autonomous Driving. *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, 2765–2771. http://arxiv.org/abs/1904.09503

Daniel Krajzewicz, Georg Hertkorn, Peter Wagner, C. R. (2002). *SUMO (Simulation of Urban MObility)*. https://elib.dlr.de/6661/2/dkrajzew_MESM2002.pdf

Dayan, P. (1993). Improving Generalization for Temporal Difference Learning: The Successor Representation. *Neural Computation*, *5*(4), 613–624. https://doi.org/10.1162/neco.1993.5.4.613

Favarò, F., Eurich, S., & Nader, N. (2018). Autonomous vehicles' disengagements: Trends, triggers, and regulatory limitations. *Accident Analysis \& Prevention*, *110*, 136–148. https://doi.org/https://doi.org/10.1016/j.aap.2017.11.001

Gu, S. (Shane), Lillicrap, T., Turner, R. E., Ghahramani, Z., Schölkopf, B., & Levine, S. (2017). Interpolated Policy Gradient: Merging On-Policy and Off-Policy Gradient Estimation for Deep Reinforcement Learning. In I. Guyon, U. V Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 30, pp. 3846–3855). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/a1d7311f2a312426d710e1c617fcbc8c-Paper.pdf

Hansen, S., Dabney, W., Barreto, A., Van De Wiele, T., Warde-Farley, D., & Mnih, V. (2019). Fast task inference with variational intrinsic successor features. In *arXiv*.

Kővári, B., Hegedüs, F., & Bécsi, T. (2020). Design of a Reinforcement Learning-Based Lane Keeping Planning Agent for Automated Vehicles. *Applied Sciences*, *10*(20), 7171. https://doi.org/10.3390/app10207171

Kulkarni, T. D., Saeedi, A., Gautam, S., & Gershman, S. J. (2016). *Deep Successor Reinforcement Learning*. http://arxiv.org/abs/1606.02396

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., & Wießner, E. (2018). Microscopic Traffic Simulation using SUMO. *IEEE Intelligent Transportation Systems Conference (ITSC)*. https://elib.dlr.de/124092/

Navarro, P. J., Fernández, C., Borraz, R., & Alonso, D. (2017). A Machine Learning Approach to Pedestrian Detection for Autonomous Vehicles Using High-Definition 3D Range Data. *Sensors*, *17*(1). https://doi.org/10.3390/s17010018

Puterman, M. L. (1990). Chapter 8 Markov decision processes. In *Stochastic Models* (Vol. 2, pp. 331–434). Elsevier. https://doi.org/https://doi.org/10.1016/S0927-0507(05)80172-0

Rajasekhar, M. V, & Jaswal, A. K. (2015). Autonomous vehicles: The future of automobiles. *2015 IEEE International Transportation Electrification Conference (ITEC)*, 1–6. https://doi.org/10.1109/ITEC-India.2015.7386874

Sallab, A. El, Abdou, M., Perot, E., & Yogamani, S. (2017). *Deep Reinforcement Learning framework for Autonomous Driving*. https://doi.org/10.2352/ISSN.2470-1173.2017.19.AVM-023

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning, second edition: An Introduction*. MIT Press. https://books.google.de/books?id=uWV0DwAAQBAJ

Szoke, L., Aradi, S., Becsi, T., & Gaspar, P. (2020). Vehicle Control in Highway Traffic by Using Reinforcement Learning and Microscopic Traffic Simulation. *2020 IEEE 18th International Symposium on Intelligent Systems and Informatics (SISY)*, 21–26. https://doi.org/10.1109/SISY50555.2020.9217076

Szőke, L., Aradi, S., Bécsi, T., & Gáspár, P. (2020). Driving on Highway by Using Reinforcement Learning with CNN and LSTM Networks. *2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES)*, 121–126. https://doi.org/10.1109/INES49302.2020.9147185

Toromanoff, M., Wirbel, E., & Moutarde, F. (2019). End-to-End Model-Free Reinforcement Learning for Urban Driving using Implicit Affordances. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 7151–7160. http://arxiv.org/abs/1911.10868

Tuncali, C. E., Fainekos, G., Ito, H., & Kapinski, J. (2018). Simulation-based Adversarial Test Generation for Autonomous Vehicles with Machine Learning Components. *2018 IEEE Intelligent Vehicles Symposium (IV)*, 1555–1562. https://doi.org/10.1109/IVS.2018.8500421

Van Niekerk, B., James, S., Earle, A., & Rosman, B. (2019). *Composing Value Functions in Reinforcement Learning*. PMLR. http://proceedings.mlr.press/v97/van-niekerk19a.html

Wang, P., Li, H., & Chan, C.-Y. (2019). Continuous Control for Automated Lane Change Behavior Based on Deep Deterministic Policy Gradient Algorithm. *IEEE Intelligent Vehicles Symposium, Proceedings*, *2019-June*, 1454–1460. http://arxiv.org/abs/1906.02275

Wang, Y., Wang, C., Zhao, W., & Xu, C. (2021). Decision-Making and Planning Method for Autonomous Vehicles Based on Motivation and Risk Assessment. *IEEE Transactions on Vehicular Technology*, *70*(1), 107–120. https://doi.org/10.1109/TVT.2021.3049794