Tudományos Diákköri Konferencia

Multi-ágens Megerősítéses Tanulás alkalmazása autópályás döntéshozatal esetére

Szerző:
Gujgiczer Dániel Tamás

Budapesti Műszaki és Gazdaságtudományi Egyetem
Közlekedés- és Járműirányítási Tanszék

Konzulens:
Dr. Szabó Ádám

TDK dolgozat,
2023. november 8.

A Kulturális és Innovációs Minisztérium **ÚNKP-23-1-I-BME-214** kódszámú Új Nemzeti Kiválóság Programjának a Nemzeti Kutatási, Fejlesztési és Innovációs Alapból finanszírozott szakmai támogatásával készült.

Kivonat

Az elmúlt évtized során a hagyományos, egyágenses Megerősítéses Tanulás először a társas-, és videójátékok, valamint a robotika területén került felhasználásra. A módszer széles körben történő alkalmazását a komplex döntéshozatali feladatok megoldására való képesség nagy mértékben segítette, mely által az autonóm járműirányításban is elterjedtté vált. Az ilyen algoritmusok ígéretes eredményeket mutatnak a saját tanítási környezetükben, azonban az egyedül tanított ágensek viselkedése kifogásolhatóvá válik más ágensekkel történő találkozáskor. Ezen probléma kiküszöbölésére használható a Multi-ágens Megerősítéses Tanulás, melynek számos fajtája közül az ún. Double DQN algoritmust alkalmaztam a kutatásom során, mely a népszerű DQN algoritmus túlbecslési problémáját küszöböli ki. A tanításokat a HighwayEnv szimulációs környezetben végeztem, mely a többsávos autópályás haladás során történő döntéshozatali szituációk egyszerű modellezésére szolgál.

A dolgozatom célja, hogy rávilágítson a Multi-ágens Megerősítéses Tanulás előnyeire az autonóm járműirányításban. Ehhez az egy- és többágenses algoritmusok teljesítménye egy vegyes, azaz ágenseket és előre meghatározott viselkedésű járműveket is tartalmazó környezetben kerül összevetésre. Ezen felül az ágensek számára visszacsatolt állapotreprezentációban szereplő tényezők hatását is tanulmányozom. Végül pedig a különböző jutalmazással végzett tanítások eredményei kerülnek összehasonlításra.

Tartalomjegyzék

1. fejezet

Bevezetés

1.1. Motiváció

A legutóbbi években a járműipari fejlődés egyik fő mozgatórugójává vált az autonóm járművek kérdésköre. A téma népszerűsége több szempontra vezethető vissza. Ezek közül az egyik legjelentősebb a közúti balesetek száma. Magyarországon a KSH adatai szerint [1], valamint az USA-ban az NHTSA tanulmánya alapján [2] a közúti közlekedésben történő balesetek 93-94%-áért a járművezetők tehetők felelőssé. A balesetek kiváltó okainak megoszlását mutatja be az 1.1 ábra.

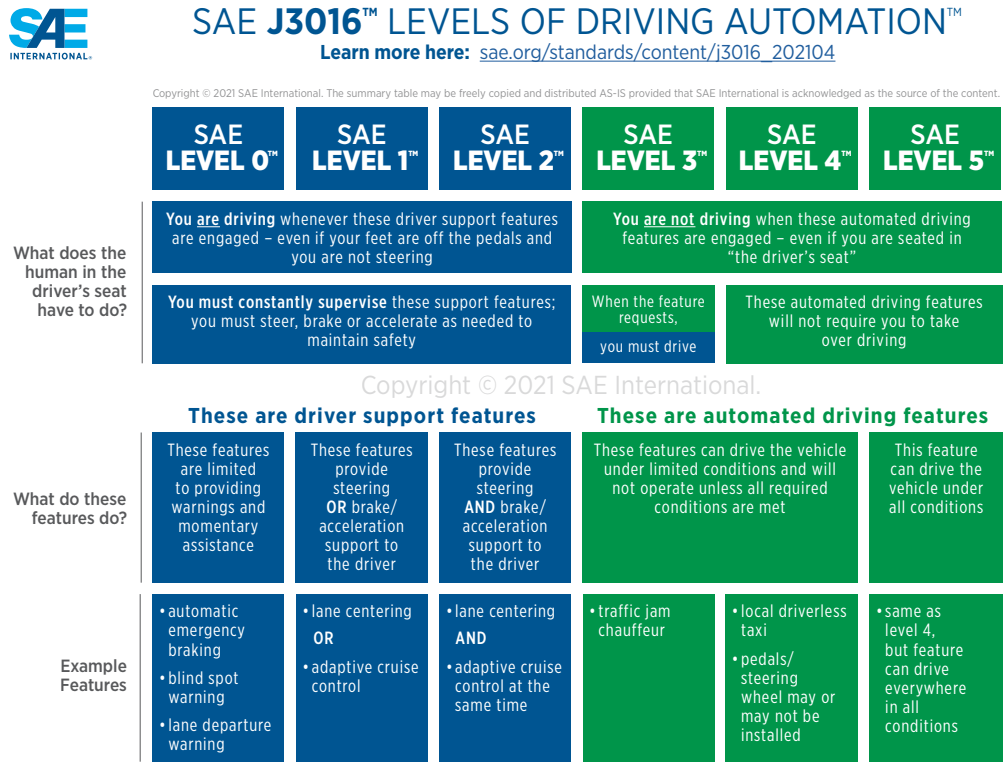
1.1. ábra. a) A közúti balesetek okai b) Az egyéb okok részletezése (forrás: [1])

Ezen események hátterében leggyakrabban a járművezetői figyelem hiánya áll (akár fáradtság, egészségügyi szempontok vagy ittas vezetés miatt). Mindezekre megoldást jelenthet az autonóm járművek alkalmazása, melyek által a balesetek száma jelentős mértékben csökkenthető [3]. [4] alapján az autonóm járművek számos további előnnyel rendelkeznek. Az előbb említett vezetőtől való függetlenség lehetővé tenné a vezetésre nem alkalmas embercsoportok (fogyatékkal élők, idősek vagy akár gyermekek) önálló közlekedését gépjárművel. Emellett a stressz csökkentéséhez is hozzájárul, ami a közlekedés biztonsága mellett egészségügyi szempontból is kiemelkedő fontosságú. További fontos tényezőként

kiemelhető a környezettudatosság. A rendelkezésre álló közlekedési infrastruktúra adta lehetőségek kihasználásával a forgalom telítettsége, valamint ezáltal az üzemanyag-felhasználás is csökkenthető lenne. Mindez a károsanyag-kibocsátás csökkentése mellett gazdasági szempontból is ideális megoldást nyújtana. A járművek közti kommunikáció által biztosíthatóvá válna az optimális haladás, akár idő, akár fogyasztás szempontjából is. A számtalan előny mellett azonban nem szabad szó nélkül hagyni az alkalmazásukkal járó hátrányokat sem. Az említett módszerek megvalósításának mind az anyagi, mind a számítási kapacitási igényei rendkívül magasak. Ezen felül természetesen az esetleges meghibásodásokkal is számolni kell. Jelenleg komoly problémát jelentenek a jogi és etikai kérdések, például, hogy a jármű az utasát vagy a gyalogost védje egy esetleges váratlan baleset során. Mindemellett a személyes adatok és a felhasznált technológiák megfelelő védelme is megoldandó feladat. Mindezen szempontokat figyelembe véve indokolható, hogy az autonóm járművek kérdésköre napjaink egyik leggyakrabban kutatott tudományterületévé vált [5].

Az autonóm járművek megvalósításának első gondolatai [6] alapján egészen 1918-ig nyúlnak vissza, míg az első tényleges koncepció a General Motors nevéhez köthető 1939-ből. Az ezt követő évtizedekben világszerte több kutatási program is indult az említett járművek létrehozására. A fejlődés első jelentősebb mérföldköveinek tekinthető az 1980-as évek végéről az Ernst Dickmanns által létrehozott autonóm jármű, valamint a PROMETHEUS nevű projekt keretein belül autópályás vezetésben elért sikerek [7]. Az Egyesült Államok Védelmi Minisztériumának kutatási egysége (Defense Advanced Research Projects Agency – DARPA) által a 2000-es években meghirdetett "Grand Challenge" és "Urban Challenge" versenyek nagy mértékben szolgálták az előrehaladást. Az elmúlt évtized során az informatika területén szintén jelentős fejlődés ment végbe. A mesterséges intelligencia, azon belül is a gépi tanulás segítségével számos sikert értek el, többek közt játékok [8] és a robotika [9] ágazatában, ami előrevetítette az említett módszerek járműiparban történő alkalmazását [10]. Ezek kezdetben vezetéstámogató rendszerekben kerültek felhasználásra. Egy adott jármű autonomitásának leírására a Society of Automotive Engineers (SAE) 2014-ben létrehozta a J3016-os szabványt, melyet a kutatások alakulásával párhuzamosan módosítottak, legutóbb 2021-ben [11]. Eszerint a jármű autonomitási szintje egy 0-5-ig tartó skálán határozható meg. Itt a 0. szint képviseli azon járműveket, melyek nem rendelkeznek vezetést segítő vagy autonóm rendszerekkel, így a járművezetőnek szükséges ellátnia az összes vezetési feladatot. Az 1. szintet a korábban említett vezetéstámogató rendszerekkel rendelkező járművek alkotják. Ezek közé tartozik például a blokkolásgátló fékrendszer (anti-lock brake system - ABS) vagy a menetstabilizátor (electronic stability program – ESP). Napjainkban már előírások szerint is számos hasonló berendezést kell tartalmaznia az új gépjárműveknek. Az ezt követő 2. szinten szerepelnek azon funkciók, melyek önálló beavatkozásra képesek, azonban ezek végrehajtásához a járművezető figyelmeztetésére és a tőle kapott megerősítésre is szükség van.

A közösségi felhasználásban jelenleg ezen a szinten lévő járművek érhetőek el, melyre az egyik legismertebb példa a Tesla Autopilot funkciója. A 3. szint újítása ehhez képest, hogy a járművezetőnek nem szükséges folytonosan fenntartani a figyelmét, azonban egy esetleges vészhelyzet esetén vissza kell tudnia vennie az irányítást a jármű felett. Emiatt a két szint közti átmenet megvalósítása fontos feladatként jelenik meg, melyek ugyan nem széles körben elterjedtek, de néhány példa található rájuk a közúton is. Az ilyen rendszerek közé sorolhatóak a "Highway Pilot" és a "Traffic Jam Pilot" jellegű autópályás haladás vagy közlekedési dugó során alkalmazott megvalósítások. Jelenleg a 4. szint jelenti a legmagasabb, már közúton is tesztelt automatizálási szintet. Ebben az esetben a járművek egy adott, számukra ismert környezetben teljesen önállóan képesek cselekedni. Ekkor csak az ismeretlen területeken, vagy nem megfelelő időjárási viszonyok esetén van szükség a járművezető irányítására. Ezen szint képviselői közé tartozik a Hyundai által Szöul városában tesztelt önvezető taxi [12]. A jelenlegi kutatások biztató eredményei alapján megállapítható, hogy a fejlődés egyértelműen az 5. szint, azaz a teljes autonomitás felé halad. A bemutatott szintek legfontosabb jellemzőit az 1.2 ábra összegzi. A még megoldandó feladatok közé tartozik a korábban felsorolt aggályok kiküszöbölése, mely által a társadalmi elfogadás is növelhető lenne. Ez mindenképp szükséges ahhoz, hogy az autonóm járművek használata által nyújtott lehetőségeket maximálisan kihasználhassuk.



1.2. ábra. Az SAE J3016-os szabványa szerinti autonomitási szintek (forrás: [11])

Habár léteznek end-to-end keretrendszerek az autonóm járműirányítás kihívásainak leküzdésére [13], a kutatók nagy része a problémák megoldását az egyes részfeladatok moduláris rendszerekkel való megvalósításában látja. A mozgástervezési feladat két részre bontható. Az alacsonyabb szintű szabályozók az adott jármű manővereit hajtják végre, mely a közlekedés többi résztvevőjétől függetlenül megy végbe. Ezzel szemben a magasabb szintű stratégiai döntéshozatal során figyelembe kell venni a többi járművet is. Ezek modellezése általában különböző járműkövetési modellekkel történik. Ezek közé tartozik például az "Intelligent Driver Model" (IDM) [14].

1.2. Kapcsolódó eredmények

A mozgástervezésben fellépő magasabb szintű szekvenciális döntéshozatali feladatok megoldására gyakran alkalmazott módszer a Megerősítéses Tanulás használata. [15] egy intelligens előzési döntéshozatali módszert mutat be Q-tanulás használatával autópályás autonóm járműirányítás esetére. Az ismertetett eredmények alapján megállapítható, hogy az algoritmus felülmúlja a hagyományos döntéshozatali metódusokat. Az ágensnek sűrű forgalom esetén tudnia kell kezelni a közeli járművek változó száma adta problémákat. A feladat megoldására [16] egy "attention-based" architektúrát mutat be, mely a neurális hálózatok számára lehetővé teszi a változó számú bemeneti adat közti összefüggések felfedezését. Ezáltal az ágens teljesítménye a teljesen összekötött-, valamint konvolúciós neurális hálóval megvalósított DQN-hez képest nagy mértékben növekszik, valamint a járművek interakciói könnyebben megjeleníthetővé válnak. Ahogy a Mély Megerősítéses Tanulási algoritmusok komplexitása növekszik, egyre inkább szükségessé válik, hogy megértsük a döntéshozatali folyamataikat a sikeres felhasználásuk érdekében. E célból [17] egy új keretrendszert javasol a vizsgálatukra.

Az egyágenses algoritmusok saját, determinisztikus tanítási környezetükben kiválóan teljesítenek, azonban ez a feltétel nem áll fenn valós forgalmi szituációk esetén. Ennek oka egyrészről, hogy a rendszer állapotai csak valamilyen bizonytalansággal mérhetők, melyeket a részlegesen megfigyelhető Markov döntési folyamatokkal (Partially Observable Markov Decision Process - POMDP) lehet leírni. Emellett a forgalom további résztvevőinek viselkedése is eltérő lehet, melyet különböző paraméterezéssel ellátott járművezetői modellekkel lehet figyelembe venni [18]. Ezen felül a bizonytalanság forrása lehet a többi, hasonló vagy különböző stratégiát követő ágenssel való interakció, melyre az ágens nincs felkészítve. Ennek megoldására használatos a Multi-ágens Megerősítéses Tanulás (Multi-Agent Reinforcement Learning - MARL), ahol több ágenst taníthatunk vegyes, azaz ágens és járművezetői modell által irányított járműveket is tartalmazó környezetben. [19] egy skálázható MARL keretrendszert mutat be, ahol az ágensek feladata a gyorsítósávról történő besorolásának segítése. [20] egy újszerű MARL architektúrát hoz létre, mely egyidejűleg több, különböző vezetési magatartást képes megtanulni. A szerzők az

ágensek homogenitását, azaz a számukra kitűzött feladat azonosságát figyelembe véve a "parameter sharing" módszerét alkalmazzák, mely az eredmények gyorsabb konvergenciájához vezet.

1.3. Feladat bemutatása

Az autonóm járműirányításban számos döntéshozatali szituáció jelenik meg. Ezek közül egy az autópályás haladás kérdésköre. Ekkor a minimálisan megfogalmazható követelmények közé tartozik a lehető legmagasabb megengedett sebességgel történő haladás. Természetesen emellett fontos tényező a kellő mértékű biztonság megléte, a balesetek elkerülése. Ezen felül a teljesség igénye nélkül feladat lehet akár a jobbra tartás és a KRESZ egyéb szabályainak betartása, az akadályok érzékelése, vagy a közlekedési táblák értelmezése is.

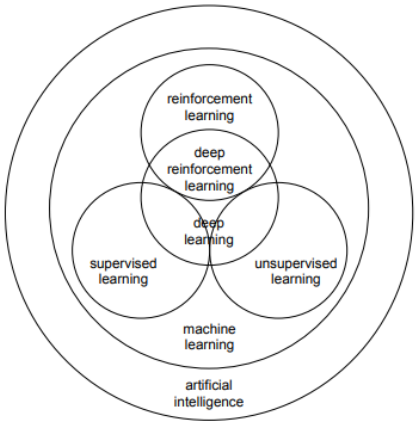
Kutatásom célja, hogy rávilágítson az egy-és többágenses algoritmusok különbségeire egy autópályás döntéshozatali szituációban, mely egy minimalista szimulációs környezetben kerül megvalósításra. A kutatás első eredményeit [21] ismerteti, melyet a TDK dolgozatom elkészítése során kibővítettem. Az egyes ágensek teljesítményei különböző szituációkban kerülnek kiértékelésre. Mindemellett a különböző összetételű állapotterek, valamint különböző jutalmazási stratégiák által a tanulás sikerességére gyakorolt hatás is vizsgálatra kerül.

A dolgozat a következőképp épül fel. A 2. fejezetben a kutatás során felhasznált módszerek elméleti háttere kerül részletesebb ismertetésre a vonatkozó szakirodalom alapján. A 3. fejezet a felhasznált környezetet mutatja be. A 4. fejezetben a tanítás tulajdonságai kerülnek ismertetésre. Az 5. fejezetben kerül sor a szimulációk eredményeinek és az algoritmusok teljesítményének kiértékelésére. Végül az utolsó, 6. fejezet az elért eredmények összegzése és a következtetések levonása mellett a jövőbeli lehetőségekre tartalmaz kitekintést.

2. fejezet

Metodológia

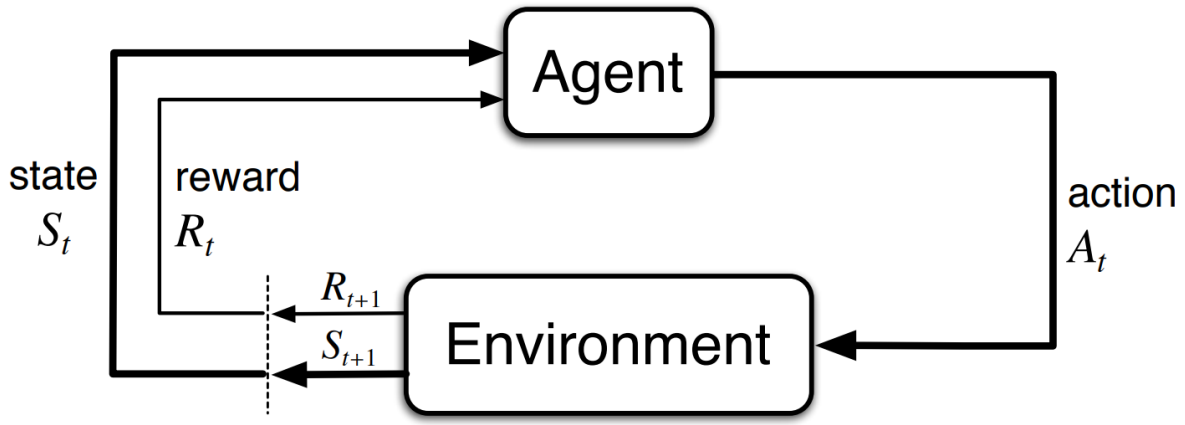
A gépi tanulás (Machine Learning - ML) jelenleg a mesterséges intelligencia (Artificial Intelligence - AI) egyik leggyorsabban fejlődő tudományterülete. Ez [22] alapján olyan algoritmusokat foglal magába, melyek a bemeneti adatok alapján végzett tanítások során szerzett tapasztalatok segítségével önállóan képesek döntések meghozatalára. Ezen belül 3 nagyobb csoportot szokás megkülönböztetni. Ezek közé tartozik a Felügyelt Tanulás (Supervised Learning), ahol címkézett adatokkal történik a tanulás, azaz egy adott bemenethez meghatározott kimenet tartozik, és ez alapján állítható fel köztük a függvénykapcsolat [23]. A második nagyobb csoport a Felügyelet nélküli Tanulás (Unsupervised Learning), ahol címkézetlen adatokból kerülnek meghatározásra a kimeneti összefüggések. Ezen módszerek hátránya, hogy egyaránt szükség van tanító adatszettekre, melyek ugyan egyre nagyobb számban állnak rendelkezésre, de sok esetben nem elegendőek. Erre nyújt megoldást a harmadik csoport, a Megerősítéses Tanulás (Reinforcement Learning - RL), melyet a kutatás során alkalmaztam, így a következő alfejezetekben részletesebben is bemutatásra kerül. A 3 klasszikus megközelítés mellett a mélytanulásról (Deep Learning - DL) is fontos szót ejteni, mely a korábban ismertetett módszerekkel közösen is alkalmazható [24]. Ezt a kapcsolatot mutatja be a 2.1 ábrán látható Venn-diagram.



2.1. ábra. A különböző gépi tanulási formák közti kapcsolat (forrás: [24])

2.1. Megerősítéses Tanulás

Megerősítéses Tanulás során a korábban bemutatott módszerekkel ellentétben nem a meglévő bemeneti adatszettekből, hanem a tanuló egység pillanatnyi megfigyelései alapján történik a tanulás. Ebben az esetben nincsenek előre definiált feladatok. A tanulás célja, hogy a döntéshozó olyan beavatkozásokat válasszon adott szituációkban, hogy egy numerikus jutalomértéket maximalizáljon. Ekkor tehát a konkrét "jó" megoldás megadása helyett az ágensnek a próbálkozásai során magának kell megismernie a legnagyobb jutalmat eredményező döntési sorozatot. Ezzel kapcsolatban kiemelendő a feladatot megfelelően körülíró jutalomfüggvény választásának fontossága. Erre a folyamatra egy ún. Markov döntési folyamatként (Markov Decision Process - MDP) tekinthetünk, mely absztrakt módon a következőképp írható le. Egy adott környezetben az ágens, vagyis a tanuló-döntéshozó egység folyamatos kölcsönhatásban van a környezetével. Ennek során megfigyeléseket végez a környezet adott pillanatbeli állapotáról, és ez alapján dönt az elvégzendő beavatkozásról. Ennek hatására a környezet állapota megváltozik, és az ágens számára egy numerikus jutalommal együtt visszacsatolódik, mely információt ad a beavatkozás minőségéről [25]. Ezt a folyamatos kölcsönhatási ciklust mutatja be a 2.2 ábra.



2.2. ábra. Az ágens és a környezet kölcsönhatása egy Markov döntési folyamatban (forrás: [25])

Az MDP leírása egy $\{S, A, P, R\}$ listaként történhet, ahol S a környezet pillanatnyi állapotát, A az ágens számára elérhető beavatkozásokat, és P az állapot-transzformációs valószínűségeket, azaz a választott beavatkozás alapján a környezetet új állapotába vivő függvényeket tartalmazza, valamint R a jutalomfüggvény. Az ágens tehát minden lépés során megkapja a környezet pillanatnyi s_t állapotát, majd az a_t beavatkozást választja. Ennek kivitelezése során a $P(s_t, a_t \mid s_{t+1})$ állapotátmenet megy végbe a környezetben. A beavatkozás minőségéről, azaz "jóságáról" a jutalom ad információt, mely minden időpillanatban kiosztásra kerül. A cél a folyamatos interakciók során a hosszú távon várható

jutalom maximalizálása, mely az alábbi egyenlet alapján határozható meg:

$$G_t = \sum_{t=1}^{T} \gamma^t r_t \tag{2.1}$$

ahol γ az ún. "discount factor", ami meghatározza a jövőbeli jutalmak fontosságát, valamint r_t az adott t időpillanatban kapott jutalom.

A cél elérése érdekében az ágens ún. "policy" függvényét fejleszti, melyet π_t-vel szokás jelölni. Ez alapvetően az ágens stratégiáját, egy adott helyzetben történő döntéshozatalát írja le. Ennek változtatási módjai az egyes Megerősítéses Tanulási metódusok szerint különböznek. Annak meghatározására, hogy egy adott "policy" követése mellett mennyire "jó" az ágens számára egy adott helyzetben lenni, az ún. "value", azaz értékfüggvényeket használjuk, melyek a következőképp írhatóak le:

Az állapot-érték (state-value) függvény:

$$v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s] \tag{2.2}$$

ahol $\mathbb{E}_\pi[\cdot]$ a változó várható értékét adja meg, ha az ágens π "policy"-t követ, valamint t az adott időpillanat.

Ehhez hasonlóan a beavatkozás-érték (action-value) függvény:

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] \tag{2.3}$$

Megerősítéses Tanulás során a feladat egy olyan optimális "policy" megtalálása, mely a legnagyobb hosszútávú jutalmat eredményezi. Ezen optimális függvény meghatározására szolgál a (2.4) optimális állapot-érték függvény, valamint a (2.5) optimális beavatkozás-érték függvény, melyek a következőképp írhatók fel:

$$v_*(s) = \max_\pi v_\pi(s) \tag{2.4}$$

$$q_*(s, a) = \max_\pi q_\pi(s, a) \tag{2.5}$$

Ezen optimális értékek meghatározására a Bellman optimalitási egyenlet szolgál, mely q_* esetén a következőképp írható fel:

$$q_*(s, a) = \mathbb{E}[R_{t+1} + \gamma \max_{a'} q_*(s', a')] \tag{2.6}$$

ahol R_{t+1} az s állapotban történő a beavatkozás után kapott jutalom, az egyenlet második fele pedig a maximális csökkentett jutalom értéke az összes lehetséges jövőbeli (s', a') állapot-beavatkozás pár közül.

Mindezek tényezők mellett egy, a Megerősítéses Tanulásra vonatkozó fontos jellem-

ző az ún. "exploration or exploitation", azaz felfedezés vagy kihasználás kérdésköre. A tanulás folyamata során a korábban említettek szerint az ágens a lehető legnagyobb összjutalomra törekszik, melyhez nyilvánvalóan a korábban szerzett tapasztalatai alapján a legtöbb jutalmat eredményező beavatkozásokat választja. Ezzel szemben viszont szükség van arra, hogy az ágens megismerje többféle beavatkozás által is a kapható jutalmakat. Ezek alapján tehát egyaránt fontos a lehetőségek megismerése próbálkozások által, valamint a tanulás előrehaladtával ezek közül a legjobbak alkalmazása. A kétféle cselekvés közti egyensúly megtalálására alkalmas az ún. "Epsilon-Greedy" algoritmus. Ekkor meghatározunk egy ún. felfedezési rátát (exploration rate, jelölése: ϵ), mely a felfedezésnek, azaz a véletlen beavatkozás választásának valószínűségét adja meg. Természetesen ekkor a korábbit tapasztalatok alapján legjobbnak ítélt beavatkozás választási valószínűsége $1-\epsilon$. A tanítás során egy lehetséges megoldás az állandó értékű ϵ használata, azonban a tanulás felgyorsulását eredményezheti, ha először a tapasztalatszerzés céljából magasabb felfedezési rátát választunk, majd az értéket a folyamat előrehaladtával csökkentjük. Mindez történhet többek között egy bizonyos jutalom-határérték elérése után [26], valamint idő vagy lépésszám alapján. A kutatás során az ϵ értékét a lépésszám alapján exponenciális módon csökkentettem egy bizonyos határértékig, mely az alábbi módon írható le:

$$\epsilon_n = \begin{cases} \epsilon_0 \, d^n & \text{ha } \epsilon_{n-1} > \epsilon_{min} \\ \epsilon_{min} & \text{ha } \epsilon_{n-1} \leq \epsilon_{min} \end{cases} \tag{2.7}$$

ahol ϵ_n a felfedezési ráta az n-edik lépésben, ϵ_0 a felfedezési ráta kezdeti értéke, d a felfedezési ráta csökkenésének mértéke, ϵ_{min} pedig a felfedezési ráta minimális értéke.

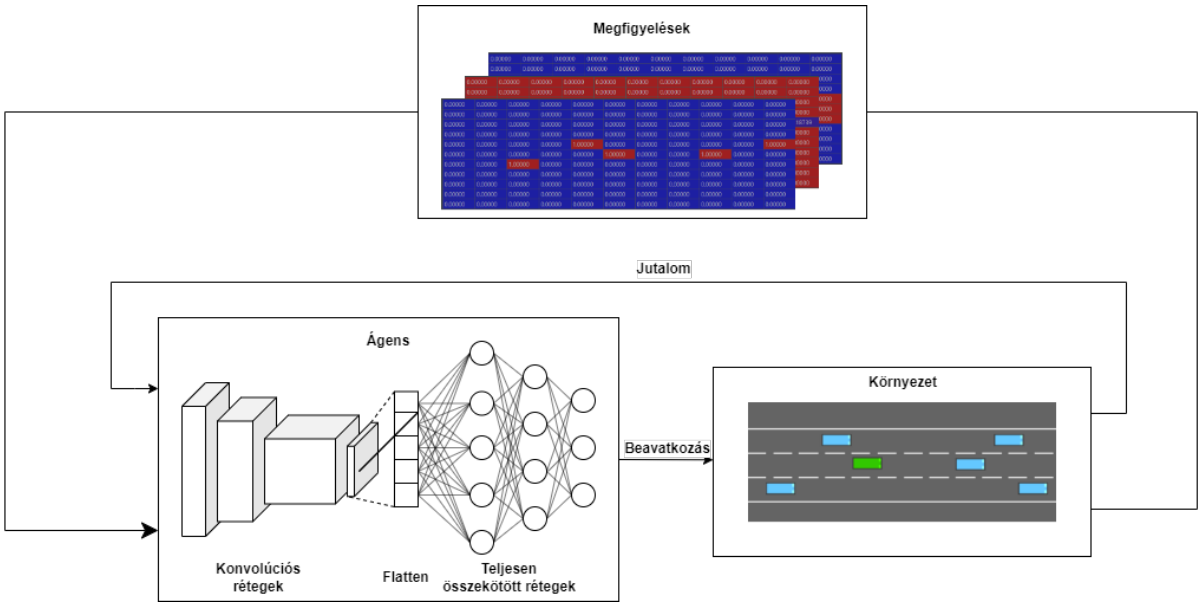
A Megerősítéses Tanulási algoritmusok többféle szempont alapján csoportosíthatóak. Attól függően, hogy az értékfüggvények vagy közvetlenül a "policy" optimalizálására törekszünk, megkülönböztetünk érték-alapú (value-based) és policy-alapú (policy-based) módszereket, valamint ezek kombinációját, az ún. "actor-critic" metódusokat. Az ágens környezetről alkotott képe alapján az algoritmusok lehetnek modell-alapúak (model-based), valamint modell nélküliek (model-free). Megerősítéses Tanulás esetén a leggyakrabban model-free algoritmusok kerülnek alkalmazásra.

2.2. Mély Megerősítéses Tanulás

Mélytanulás (Deep Learning) során mély neurális hálózatokat (Deep Neural Network) alkalmazunk számítási modellként, melyek a bemeneti adatok tetszőleges reprezentációját képesek megtanulni [27]. Egy ilyen hálózat szerkezetében a bemeneti és kimeneti rétegek mellett egy vagy több ún. rejtett réteg is található. A bemeneti után következő rétegek esetén az egyes neuronok bemeneti értékei az előző réteg neuronjai által adott kimenetek súlyozott összegeként határozhatóak meg [24]. Az egyes rétegek neuronjai közti

kapcsolatot ezek a súlyok adják meg, melyeket a θ paraméter tartalmaz. Az összegzés után általában valamilyen nemlineáris transzformációt vagy aktivációt alkalmaznak a neuron bemenetére, mely lehetőséget nyújt a modell számára komplexebb összefüggések megtanulására. Az ilyen függvényekre példa lehet a sigmoid, a tanh, valamint az egyre gyakrabban alkalmazott ReLU. A neurális hálózatokkal megvalósított tanulási modellek pontosságát a veszteségfüggvény fejezi ki. Ennek optimalizálása a "backpropagation", azaz hiba-visszaterjesztés módszerével történik, ahol a háló paramétereit a korábbi iteráció hibái alapján változtatjuk. Mindezt különböző optimalizáló algoritmusok, mint az SGD, Adam vagy RMSprop hajtják végre. Ezen módszer fontos paramétere az ún. "learning rate" vagy tanulási ráta (jelölése: α), mely a paraméterek változtatásának mértékét határozza meg iterációs lépésenként [28].

Jelenleg számos alkalmazási területe ismert a mélytanulásnak, például képek és videók feldolgozásában vagy beszédfelismerésben. Mindezen sikerek kapcsán felmerült a Megerősítéses Tanulással közös alkalmazása is. A két tanulási módszer kombinációját nevezzük Mély Megerősítéses Tanulásnak (Deep Reinforcement Learning). Ekkor a neurális hálózatokat függvényapproximátorként használjuk, például az értékfüggvények vagy a "policy" függvény modellezésére. A korábban alkalmazott approximátorokkal (pl. SVM, döntési fa) szemben nagy előny, hogy a neurális hálózatok képesek nagyobb mennyiségű adat feldolgozására [28]. A kutatás során is felhasznált Konvolúciós Neurális Hálózatok (Convolutional Neural Network - CNN) egyik fő felhasználási területe a kép jellegű adatok feldolgozása. Az ilyen hálózatok első rétegei a konvolúciós rétegek, ahol a bemeneti adatokra a "kernel" paraméter alapján, mely a feldolgozási szűrők számát adja meg, konvolúciós műveletet végzünk. Ez után a "pooling" réteg segítségével a konvolúció felbontását csökkentjük, majd a "flatten" réteggel a többdimenziós adatokból egydimenzióst csinálunk, hogy a hálózat végén szereplő teljesen összekötött rétegek bemenetét képezhessék. Az ilyen hálózattal tanított ágensek környezettel való interakcióját mutatja be a 2.3 ábra. Ez alapján látható, hogy a háló bemenetét a környezetről végzett megfigyelések képzik, a kimenet pedig a választott beavatkozás.



2.3. ábra. A Mély Megerősítéses Tanulás megvalósítása Konvolúciós Neurális Hálózattal

2.3. DDQN

A Q-tanulás (Q-learning) fogalma először [29]-ben került először definiálásra, mely a Megerősítéses Tanulás fejlődésének egyik fontos mérföldköve volt. A módszer az érték-alapú, modell nélküli algoritmusok közé sorolható, mely során az optimális "policy"-t a beavatkozás-érték függvény módosításain keresztül érjük el. A Q-tanulás az ún. "Temporal Difference (TD)" tanulásra vezethető vissza [25]. A Q értékek frissítési szabályát a következő egyenlet írja le:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)] \tag{2.8}$$

A kezdetben alkalmazott tabuláris Q-tanulás esetén az egyes Q értékek táblázatszerűen tárolódtak, melyek az állapottér és a beavatkozási tér méretének növekedésével túl nagy számítási kapacitásigényhez vezettek, ezáltal indokolttá vált a beavatkozás-érték függvény neurális hálózattal történő közelítése. Ebben az esetben a (2.6) Bellman optimalitási egyenletben a Q értékek a háló θ_t paramétereitől is függővé válnak, így az egyenlet a következőképp módosul:

$$Q_t^* = R_{t+1} + \gamma\, Q(S_{t+1}, \underset{a}{argmax} Q(S_{t+1}, a; \theta_t); \theta_t) \tag{2.9}$$

A Q-függvény neurális hálóval történő, nemlineáris approximációja instabilitáshoz, vagy akár divergenciához is vezethet, melynek okai a megfigyelések sorozatában fellelhető korrelációk, a tényleges Q-értékek és a közelítendő célértékek közti kapcsolat, valamint az a tény, hogy a Q-függvényben történő kisebb módosítások is jelentős változásokat

okozhatnak a "policy" függvényben. Ennek kiküszöbölésére hozták létre [30] szerzői a "Deep Q-Network", azaz DQN algoritmust, mely a következő újításokat tartalmazza a hagyományos, neurális hálóval történő megvalósításhoz képest. A megfigyelési szekvencia korrelációinak feloldására az ún. "experience replay" módszerét alkalmazzák, mely során az ágens tapasztalatai adatszettekben tárolódnak el, amiből véletlenszerűen vesznek mintát a Q-tanulás paramétereinek frissítéséhez. Emellett a másik két említett probléma feloldását egy "target" hálózat alkalmazásával végzik, melynek paraméterei (θ_t^-) a folyamatosan frissített Q-hálózat értékeivel csak bizonyos lépésközönként íródnak felül. Ekkor az optimális Q érték egyenlete a következőképp írható fel:

$$Q_t^{*DQN} = R_{t+1} + \gamma\, Q(S_{t+1}, \underset{a}{argmax} Q(S_{t+1}, a; \theta_t^-); \theta_t^-) \tag{2.10}$$

A hagyományos Q-tanulás és DQN esetén is látható, hogy a beavatkozás kiválasztásához, valamint kiértékeléséhez is ugyanazon paramétereket alkalmazzuk. Emiatt az algoritmus hajlamosabbá válik túlbecsült értékeket választani, és az ebből eredő túlzott optimizmus a tanulási folyamat előrehaladtával jelentős eltéréseket okozhat. A túlbecslési probléma kiküszöbölését szem előtt tartva jött létre a Double Q-learning algoritmus. Ebben az esetben a beavatkozás kiválasztása elkülönítésre kerül a kiértékeléstől, mely célból két, különböző paraméterekkel (θ_t és $\theta_t^{'}$) rendelkező neurális hálózatot alkalmazunk. Az egyes hálókat váltakozóan, az epizódok során szerzett tapasztalatokkal véletlenszerűen frissítjük. Q^* egyenlete ekkor az alábbiak szerint alakul:

$$Q_t^{*DoubleQ} = R_{t+1} + \gamma\, Q(S_{t+1}, \underset{a}{argmax} Q(S_{t+1}, a; \theta_t); \theta_t^{'}) \tag{2.11}$$

Ez alapján látható, hogy a beavatkozás választását, valamint a kiértékelést különböző paraméterekkel rendelkező hálózatokkal végezzük.

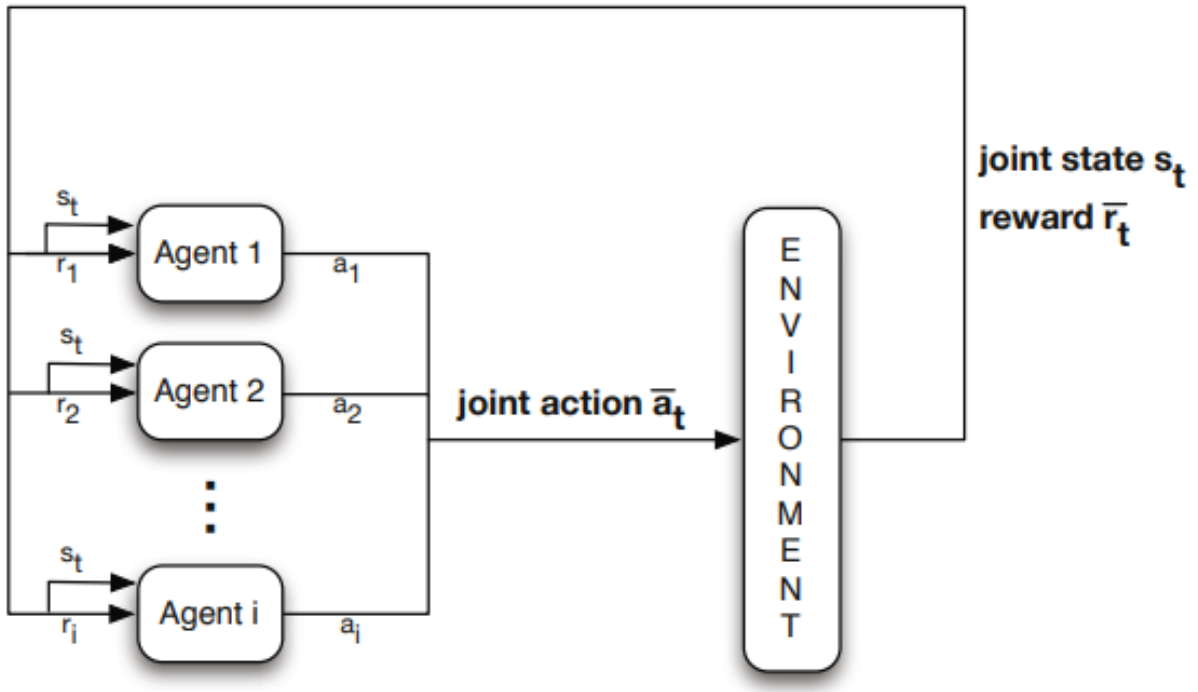
[31] szerzői a Double Q-learning és a DQN előnyeit ötvözve hozták létre a Double DQN (DDQN) algoritmust. Ekkor a Bellman optimális egyenlet a következő alakban írható fel:

$$Q_t^{*DDQN} = R_{t+1} + \gamma\, Q(S_{t+1}, \underset{a}{argmax} Q(S_{t+1}, a; \theta_t); \theta_t^-) \tag{2.12}$$

A (2.11) egyenlettel összevetve megfigyelhető, hogy a hagyományos Double Q-learning algoritmussal szemben mindössze annyi módosult, hogy az értékbecslés a DQN-ben bevezetett "target" háló értékei alapján történik. Emellett fontos kiemelni, hogy a "target" háló frissítése a DQN-ben megismert módon, bizonyos számú lépés után történik. Ezen kívül a DQN másik fontos összetevője, az "experience replay" szintén megtalálható a DDQN módszertanában is.

2.4. Multi-ágens Megerősítéses Tanulás

Multi-ágens Megerősítéses Tanulás (Multi-Agent Reinforcement Learning - MARL) esetén egy közös környezetben több szekvenciális döntéshozó ágens is szerepel, melyekkel egyidejűleg interakciót folytat [32]. Ekkor az egyágenses tanulásban megismert Markov döntési folyamat egy ún. "Markovi játékká" (Markov Game) alakul, mely először [33] által került leírásra. Ez az MDP kiterjesztésének fogható fel, mely által az ágensek képessé válnak a környezettel és az egymással való interakcióra. A korábban bemutatott lista az ágensek számának függvényében módosul, mely így $\{N, S, A_i, P, R_i\}$ formában írható fel, ahol N az ágensek száma, S az összes ágens által megfigyelt állapottér, A_i az egyes ágensek beavatkozásainak vektora, P az összes ágens beavatkozása által elért állapot-átmeneti valószínűségi függvény, R_i pedig az egyes ágensek számára külön visszacsatolt jutalomérték. Ezt az interakciós folyamatot mutatja be a 2.4 ábra.



2.4. ábra. Az ágensek és a környezet interakciója Multi-ágens Megerősítéses Tanulás esetén (forrás: [34])

A különböző MARL tanítási sémák csoportosítása többféle szempont szerint történhet. [35] alapján az ágensek feladata szerint megkülönböztethetünk (teljesen) kooperatív, (teljesen) kompetitív és vegyes algoritmusokat. Kooperatív esetben az ágensektől együttműködést várunk egy közös cél érdekében. Ebből kifolyólag az összes ágens egy közös jutalmat kap. Ezzel szemben kompetitív esetben az egyes ágensek csak a saját jutalmuk maximalizálására törekszenek, még abban az esetben is, ha ezzel másik ágenst hátráltatnak. Vegyes algoritmusok esetén pedig se nem teljesen kooperatív, se nem teljesen

kompetitív módszerekről beszélünk, tehát ebben az esetben nincsenek megkötések a jutalmazást tekintve. Emellett az egyes ágensek többiről való tudomása szerint is történhet a csoportosítás. Eszerint megkülönböztethetünk centralizált és decentralizált környezeteket. Centralizált esetben az összes ágens megfigyelései elérhetőek és ismertek, ezeket a beavatkozásokkal együtt egy közös neurális háló kezeli. Ezen módszerrel kis ágensszám esetén jó eredmények érhetőek el, azonban az ágensszám növelése során skálázhatósági problémák lépnek fel. A másik nagyobb csoportot a decentralizált környezetek adják, ahol minden ágens a saját "policy" függvénye alapján cselekszik, melyek párhuzamosan kerülnek tanításra. Ekkor a tanulási folyamat előrehaladtával, ahogy az ágensek egyre inkább alkalmazkodtak egymáshoz, folyamatosan nő a tanulási idő.

A több döntéshozó egységet is tartalmazó környezetek kezelésére való képesség mellett a Multi-ágens Megerősítéses Tanulás előnyei közé tartozik a tanulás sebességének növelése a párhuzamos számítások által. Ezen felül az ágensek közti kommunikáció (vagy tapasztalataik egymással való megosztása) is hasznos lehet azonos feladatot ellátó ágensek között, valamint az ágensek egymást is taníthatják ezáltal. Mindemellett lehetőség nyílik akár arra is, hogyha valamelyik ágens nem tudja feladatát ellátni, akkor egy másik ezt helyette elvégezheti. Több ágens jelenléte egy környezetben azonban többféle problémához is vezethet. Ezek közé tartozik a nem stacioner környezet, melyet az egyszerre történő több beavatkozás okoz. A részleges megfigyelhetőség problémája a Részlegesen Megfigyelhető Markov döntési folyamat (Partially Observable Markov Decision Process - POMDP) használatával küszöbölhető ki, ahol a korábban bemutatott listában lévő állapottér szétválik az egyes ágensek megfigyeléseinek vektorára, valamint a rendszer nem megfigyelt állapotaira. Kooperatív környezetekben a közösen osztott jutalom alapján az egyes ágensek nem tudnak következtetni arra, hogy ehhez mennyi volt a saját hozzájárulásuk. Ez okozza az ún. "Credit Assignment" jutalmazási problémát. Erre a jelenségre például szolgálhat akár a közúti közlekedés is, ahol ugyan egyértelműnek tűnhet a jutalmazási stratégia megválasztása, de a jutalmak eloszlása összetett kérdéseket vet fel. [36] alapján a kooperatív jutalmazás eredményezhet egy önzetlen, kooperatív stratégiát, azonban felmerülhet az ún. "lazy agent" probléma is. Mindezek mellett [32] az ún. "shadowed equilibrium" problémát emeli ki.

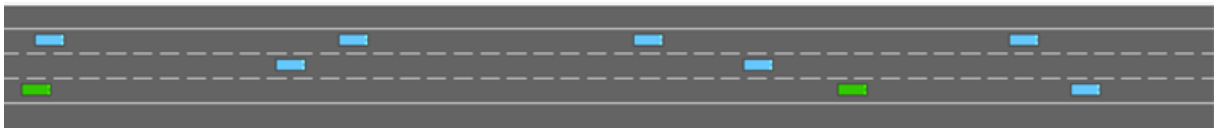
A centralizált tanulás legszélsőségesebb fajtája az ún. "parameter sharing" [37], mely a Megerősítéses Tanulásban először [38]-ben került definiálásra, majd később a Mély Multi-ágens Megerősítéses Tanulásban is sikeresen alkalmazták [39]. Ez az algoritmus egyetlen "policy" függvényt tartalmaz, melyet az ágensek közösen fejlesztenek, ezáltal kézenfekvő megoldásként szolgál abban az esetben, amikor az egyes ágensek számára hasonló, vagy akár ugyanaz a megvalósítandó feladat. Mindez az erőforrás-igény csökkenését, valamint a konvergencia gyorsulását eredményezi.

3. fejezet

Felhasznált környezet

3.1. Autópálya környezet

A szimulációk során a bevezetésben ismertetett probléma modellezésére a Farama Foundation projektjének részét képző Highway-env [40] környezetet használtam fel. Ez többféle közlekedési szituáció (pl. autópálya, kereszteződés, körforgalom) minimalista reprezentációjának gyűjteményét biztosítja az autonóm járműirányításban fellépő döntéshozatali feladatok modellezésére. Mindezek közül a kutatás során a highway, azaz autópálya környezetet alkalmaztam, mely egy többsávos autópályás forgalmi szituációt reprezentál. A 3.1 ábra a környezet grafikus megjelenítését mutatja be, melyen zöld színnel az ágens által irányított, míg kékkel az egyéb járművek láthatóak.



3.1. ábra. A környezet vizualizációja

A következőkben a környezet egyes jellemzői a környezet dokumentációja alapján kerülnek bemutatásra. A haladás során az ágens feladata több részre bontható. Ezek közül a legfontosabb az előre meghatározott tartománybeli referenciasebesség tartása, a többi járművel való ütközés elkerülése mellett. Ezen felül a jobbra tartás is figyelembe vett szempont. A felsorolt tényezők határozzák meg a tanítási folyamat során az ágens jutalmazását, tehát a jutalomfüggvény ezekből az összetevőkből áll össze. A sebesség alapú jutalom egy meghatározott sebességtartományon belül kerül kiosztásra, nulla és a maximális jutalom közt lineárisan leképezve. Ez a következőképp írható le:

$$R_s = \begin{cases} \frac{v - v_{min}}{v_{max} - v_{min}} R_{s_{max}}, & ha\ v_{min} \leq v \leq v_{max} \\ 0, & egyébként \end{cases} \tag{3.1}$$

ahol R_s a sebesség alapú jutalom, v az irányított jármű adott pillanatbeli sebessége, v_{min} és v_{max} a jutalmazási sebességhatárok, valamint $R_{s_{max}}$ a maximális sebességnél elérhető jutalom értéke.

Más járművekkel való ütközés esetén az ágens büntetést kap, valamint az epizód megszakításra kerül:

$$R_c = \begin{cases} R_{crashed} & ha\ ütközött \\ 0 & egyébként \end{cases} \tag{3.2}$$

ahol R_c az ütközési jutalomrész és $R_{crashed}$ a büntetés értéke.

A jobbra tartás jutalma R_l szintén lineáris leképezéssel áll elő az alapján, hogy melyik sávban halad a jármű. Ebben az esetben tehát a leginkább jobboldali sáv jelenti a maximális, míg a legbelső sáv a 0 jutalmat. A jutalomfüggvényben ezek a részek kerülnek összegzésre, majd 0 és 1 között normalizálásra:

$$r = R_s + R_c + R_l \tag{3.3}$$

$$R = \frac{r - R_{crashed}}{R_{s_{max}} + R_{l_{max}} - R_{crashed}} \tag{3.4}$$

A járművek kinematikáját a Kinematikai Kerékpármodell [41] adja meg, mely az alábbi egyenletekkel írható le:

$$\dot{x} = v\cos(\psi + \beta) \tag{3.5}$$

$$\dot{y} = v\sin(\psi + \beta) \tag{3.6}$$

$$\dot{v} = a \tag{3.7}$$

$$\dot{\psi} = \frac{v}{l}\sin\beta \tag{3.8}$$

$$\beta = \tan^{-1}(\frac{1}{2}\tan\delta) \tag{3.9}$$

ahol (x, y) a jármű pozíciója, v a sebessége, ψ az irányszöge, a a gyorsulása, β a szlip szöge a súlypontnál, δ pedig az első kerekek kormányszöge.

A nem ágens által irányított járművek hosszirányú viselkedését az IDM modell [14], míg keresztirányú viselkedését a "Minimizing Overall Braking Induced by Lane change" (MOBIL) modell határozza meg [42].

Az IDM modell a következő egyenletekkel adja meg a jármű gyorsulását:

$$\dot{v} = a[1 - (\frac{v}{v_0})^{\delta} - (\frac{d^*}{d})^2] \tag{3.10}$$

$$d^* = d_0 + Tv + \frac{v\Delta v}{2\sqrt{ab}} \tag{3.11}$$

ahol v a jármű sebessége, d az előtte haladó járműtől való távolsága, a dinamikáját pedig a következő paraméterek adják meg, melyeket a környezet inicializálásakor definiálunk: v_0 a kívánt sebesség, T a kívánt időrés, d_0 a kívánt távolság, a és b a legnagyobb gyorsulás és lassulás értékek, valamint δ a sebességi kitevő.

A MOBIL modell szerint a jármű akkor vált sávot, amikor biztonságos (vagyis nem "bevág" a másik jármű elé) (3.12), valamint ha arra ösztönzi a forgalom (3.13). Ez a két feltétel a következő egyenletekkel írható le:

$$\widetilde{a}_n \geq -b_{safe} \tag{3.12}$$

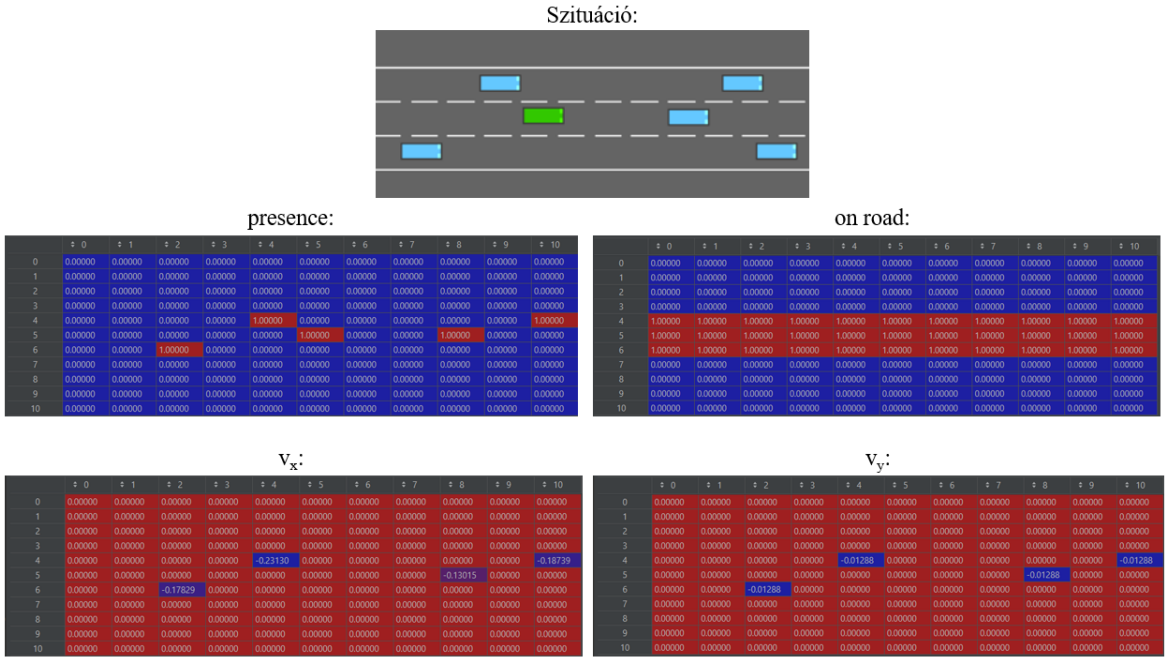
$$\widetilde{a}_c - a_c + p(\widetilde{a}_n - a_n + \widetilde{a}_o - a_o) \geq \Delta a_{th} \tag{3.13}$$

ahol c jelöli az irányított járművet, o a sávváltás előtti, míg n a sávváltás utáni követő járművet, a és $\widetilde{a}$ a járművek sávváltás előtti és utáni gyorsulása, valamint a környezet inicializálásakor meghatározott paraméterek közé tartozik p "előzékenységi" kitevő, Δa_{th} gyorsulásváltozás, mely a sávváltás előidézéséhez szükséges, valamint b_{safe} a maximálisan kiszabható lassulás "bevágás" esetén.

3.1.1. Állapotreprezentáció

A tanítás szemszögéből az egyik legfontosabb tényező az, hogy a jármű hogyan érzékeli és írja le a környezetét, mivel a neurális háló bemenetét ezen megfigyelések képzik. Ezt a környezet „Observation” paramétere adja meg, melyből a projektben alapvetően több változat is megtalálható, többek közt a következőek: "Kinematics", "Grayscale Image", "Occupancy grid" és "Time to collision". A feladat megoldására ezek közül választottam állapotreprezentációt. A kiválasztás során fontos szempont volt, hogy az eleve magas számításigényű Multi-ágens Megerősítéses Tanulási folyamat tanulási idejét ne növelje túlságosan a választott állapottér, valamint tetszőleges számú jármű és megfigyelt tényező kezelésére legyen képes. A felsorolt reprezentációk közül a "Grayscale Image" egy szürkeárnyalatos képét képzi az adott időpontbeli állapotnak, melynek feldolgozása túlságosan erőforrásigényes. A "Kinematics" állapotterének nagyságát a környezetben szereplő járművek száma határozza meg, mely paraméter így nem módosítható tetszőlegesen. A "Time to collision" reprezentáció esetén pedig csak a közelben lévő járművekkel való ütközés idejére történik becslés a járművek sebessége alapján, más paraméter megfigyelésére nincs lehetőség. Ezen tényezők figyelembevételével esett a választásom az "Occupancy Grid" állapotreprezentációra, mely egy 3D mátrixként fogható fel. Ez a jármű körüli kétdimenziós teret előre meghatározott méretű cellákra osztja úgy, hogy az irányított jármű mindig középen található. A mátrix harmadik dimenzióját a megfigyelt jellemzők száma határozza meg, melyek így külön csatornákon kerülnek eltárolásra. Az állapotreprezentáció egyes cellái az ágenshez képesti relatív értékeket tartalmazzák, valamint 0 és 1 közé

vannak normalizálva. Erre mutat be egy példát a 3.2 ábra, ahol egy adott szituációban láthatóak az egyes tényezők szerint elkülönített csatornák.



3.2. ábra. Az "Occupancy Grid" állapotreprezentáció

A választott reprezentáció előnye, hogy annak mérete független a környezetben található járművek számától. Ennek köszönhetően ezen paraméter értéke véletlenszerűen felvehető, mellyel a valóságos közúti forgalmi szituációk közelíthetőek. A cellákra osztással kellően részletes, de nem túl komplex információt kaphatunk a járművek környezetéről. Mindemellett pedig tetszőleges számú tényező figyelhető meg vele. Ezzel szemben a hátrányai közé tartozik, hogy sok "üres", azaz 0 értéket tartalmazó eleme van, mivel az út környezetét is érzékeli. Ezzel kapcsolatban a megfigyelt tér méretének igazítása, azaz "keskenyítése" is felmerülhet. Azonban mindezen jellemzőket szem előtt tartva, mivel az összes korábban megfogalmazott elvárást teljesíti, így indokolható a reprezentáció választása.

3.1.2. Beavatkozási tér

A tanulás egy másik alapvető tényezője a neurális háló kimenetét képző választott beavatkozás. Ezzel kapcsolatban fontos kiemelni, hogy a döntéshozatal folyamata két szintre bontható. Az alacsonyabb szintű döntéshozatal során közvetlenül a jármű gyorsulását és kormányzási szögét szabályozzuk. Ezzel szemben a magasabb szinten a sáv- vagy a sebesség változtatásáról születik döntés, ami egy utasítás formájában kerül továbbításra az alacsonyabb szintű szabályozóknak a kivitelezéshez. A környezetben mindkét szintű beavatkozáshoz találhatóak implementációk. Az alacsonyabb szinten folytonos és diszkrét

beavatkozási típusok érhetőek el, azonban a feladat során az ún. "Discrete Meta-Actions" típust használtam fel. Ez a magasabb szintű döntéseket reprezentálja, azaz a beavatkozási tér a következő diszkrét beavatkozásokból áll: balra vagy jobbra történő sávváltás, gyorsítás vagy lassítás, valamint az ún. „idle”, azaz sáv- és sebességtartás. Az ágens által kiválasztott beavatkozást ebben az esetben egy arányos (P) hosszirányú, valamint egy arányos-derivatív (PD) keresztirányú szabályozó hajtja végre. A reprezentáció előnyei közé tartozik még, hogy olyan szituációkban, amikor a választott beavatkozás nem lenne végrehajtható (pl. bal szélső sávban haladás esetén balra sávváltás, maximális sebességgel haladáskor gyorsítás), automatikusan az "idle" beavatkozás lép életbe.

3.2. Módosítások

A környezet alapvetően támogatja a MARL algoritmusok használatát, azonban ez nem minden részkörnyezet esetén érhető el. Ebből kifolyólag a „highway” módosítására volt szükség, mely során az állapotreprezentáció, a választott beavatkozások, valamint a jutalmak számítása úgy változott, hogy egyszerre több ágens kezelésére is képes legyen. Mindezt az egyes járművek megfigyeléseinek és beavatkozásainak vektorba rendezésével valósítottam meg. Ezen vektorok hosszát az ágensek száma határozza meg, mely által a módszer egyágenses tanításra is alkalmazható. A tanítási folyamat felgyorsítása érdekében a környezet egyik gyorsabb változatát, a „highway-fast”-et használtam fel. Ez a szimuláció frekvenciáját és a környezetben szereplő járművek számát csökkenti, mely által az epizódok hossza is lerövidül. Emellett a nem ágens által irányított járművek közti ütközések sem kerülnek figyelembevételre.

4. fejezet

Tanítási folyamat

A felkonfigurálás után egy DDQN ágenst implementáltam a környezetbe. A feladat megoldására a cella jellegű információk feldolgozására gyakran alkalmazott konvolúciós neurális hálózatot (Convolutional Neural Network – CNN) használtam, figyelembe véve a korábban megadott állapotreprezentációt. Az ágens és a hálózat hiperparamétereit a 4.1 táblázat foglalja össze.

4.1. táblázat. A tanításhoz használt hiperparaméterek

Paraméter	**Érték**
Learning rate	0,0001
Discount factor	0,99
Epsilon decay	0,999995
Epsilon min	0,01
Batch size	128
Burn-in	256
Memória mérete	100000
Sync every	20
Rejtett konvolúciós rétegek száma	3
Szűrők száma (rétegenként)	32, 128, 2
Rejtett teljesen összekötött rétegek száma	3
Neuronok száma (rétegenként)	4096, 256, 128
Aktivációs függvény	ReLU
Optimalizáló algoritmus	Adam
Veszteségfüggvény	MSE
Sebesség jutalmazási tartomány	[20, 30]
Sebességi jutalom	0,4
Ütközési büntetés	-1
Jobbra tartási jutalom	0,1

A tanításokat egy 3 sávos, 20 járművet tartalmazó környezetben végeztem, melyek közül 1-et vagy 4-et irányított ágens. Az egy- és többágenses tanításokat első körben

egyaránt az eredeti állapotreprezentációval végeztem, ahol a megfigyelt tényezők a következőek voltak:

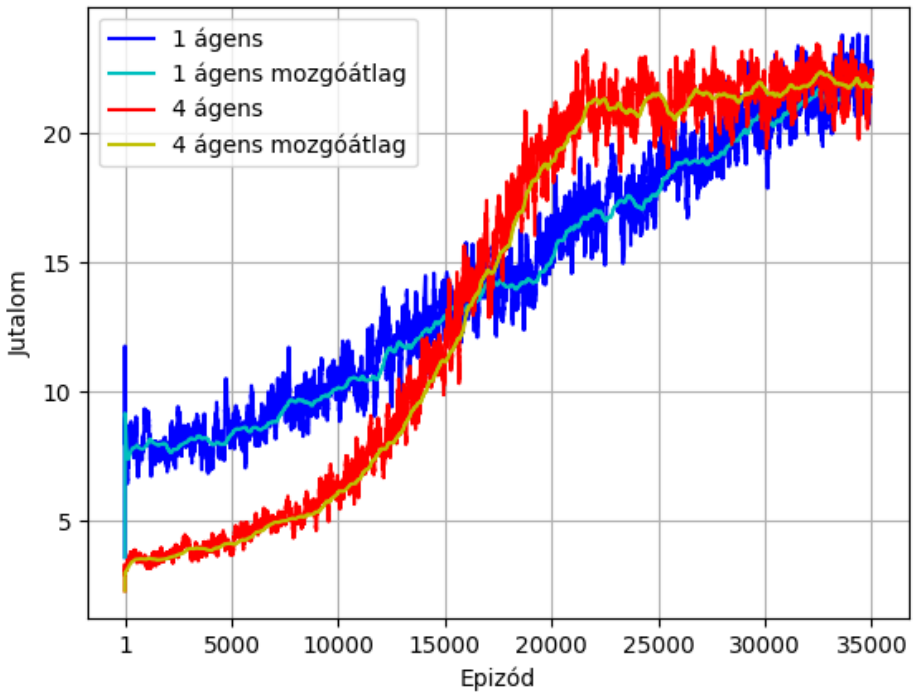
- A jármű adott cellában való jelenlétét jelző bináris indikátor (*presence*)
- Az utat és a környezetét elkülönítő bináris indikátor (*on road*)
- Az adott cellában található jármű hosszirányú sebessége (v_x)
- Az adott cellában található jármű keresztirányú sebessége (v_y)

Az első iterációkat követően az állapottér az alábbi tényezőkkel került bővítésre:

- A jármű következő lépésben kívánt helyzete irányszögének szinusza ($\sin d$)
- A jármű következő lépésben kívánt helyzete irányszögének koszinusza ($\cos d$)

Mindezen bővítések hatására a megfigyeléseket végző jármű a többi ágens pillanatnyi állapota mellett a jövőbeli szándékaikról is információhoz jutott.

A Multi-ágens Megerősítéses Tanulás során minden esetben a korábban bemutatott „parameter-sharing" módszert alkalmaztam. Az ágensek jutalmazása teljesen kompetitíven történt, azaz minden ágens a saját cselekedetei alapján külön jutalmat kapott. A tanítási folyamat, ahogy az a 4.1 ábra alapján is látható, 35000 epizódon keresztül tartott.



4.1. ábra. A tanítási folyamat átlagos jutalomértékei

5. fejezet

Eredmények

A kutatás során a tanítást először egy ágenssel végeztem mindkét korábban bemutatott állapottér esetére, majd Multi-ágens megerősítéses tanítási algoritmussal ismételtem meg, mely 4 ágenst irányított egyidejűleg.

Az egyes betanított hálózatok teljesítményét egy 100 epizódból álló teszt segítségével értékeltem ki. A tesztet a tanításhoz hasonlóan egy 3 sávos és 20 járművet tartalmazó környezetben végeztem el. Mindegyik hálózat esetén egy, négy, nyolc és tizenkettő ágens által irányított járművel is megismételtem a tesztet. A teljesítmények összehasonlítására a következő metrikákat vizsgáltam:

- Ütközések száma a 100 epizód során
- Átlagos jutalom
- Az ágensek által irányított járművek átlagsebességei

5.1. Az ágensek számának hatása

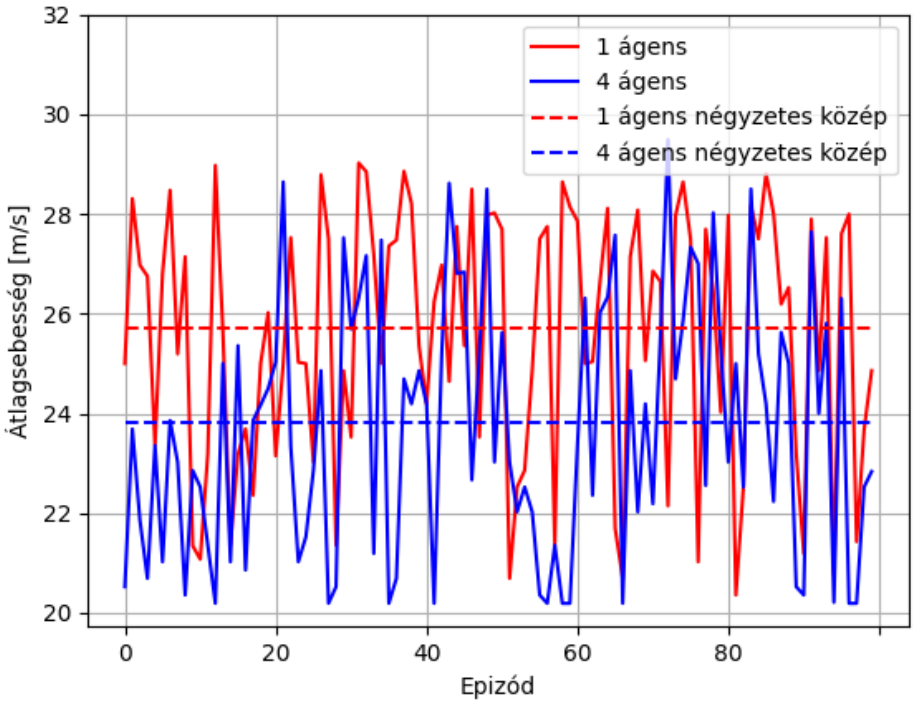
Elsőként a tanítási ágensszám okozta különbségeket vizsgáltam az eredeti állapottér esetére. Mindezen tesztek eredményeit az 5.1 táblázat és az 5.1 ábra foglalja össze.

Az 5.1 táblázat alapján megállapítható, hogy a két módszer közül a multi-ágens "policy" használatával jelentősen kevesebb ütközés történt az ágensszám növelése esetén is. Ezen okból kifolyólag az átlagos jutalmak is magasabb értékeket vettek fel, mely szintén a robusztusabb viselkedést tükrözi. Az egyágenses algoritmus abban az esetben ért el magasabb átlagjutalmat, amikor a környezet nem tartalmazott másik ágenst. Ekkor azonban a sikertelen epizódok száma is növekedett. Mindez arra vezethető vissza, hogy az ágens a sebesség alapú jutalom maximalizálásának érdekében magasabb sebességet hajlamos választani, ami azonban gyakran eredményez ütközést. Ezzel szemben a multi-ágens algoritmus alacsonyabb átlagsebességet választ, mely által az ütközések száma is csökken,

5.1. táblázat. Teszteredmények az eredeti állapottér használatakor

Ágensek száma (kiértékelés)	Ágensek száma (tanítás)	Ütközések	Átlagos jutalom
1	1	5	24,56
	4	2	23,94
4	1	30	21,19
	4	14	23,39
8	1	55	18,82
	4	32	21,07
12	1	79	14,26
	4	46	19,92

és így a jutalmak eloszlása is egyenletesebbé válik az epizódok során. Az ágensszám növelésével észrevehető, hogy a két módszer teljesítménye közti különbség növekszik, azaz az ütközések számának növekedése és így az átlagjutalmak csökkenése egyágenses "policy" esetén nagyobb mértékű, mint multi-ágens esetben.



5.1. ábra. Az első ágens átlagos sebességértékei az eredeti állapottér esetén

5.2. A megfigyelt jellemzők hatása

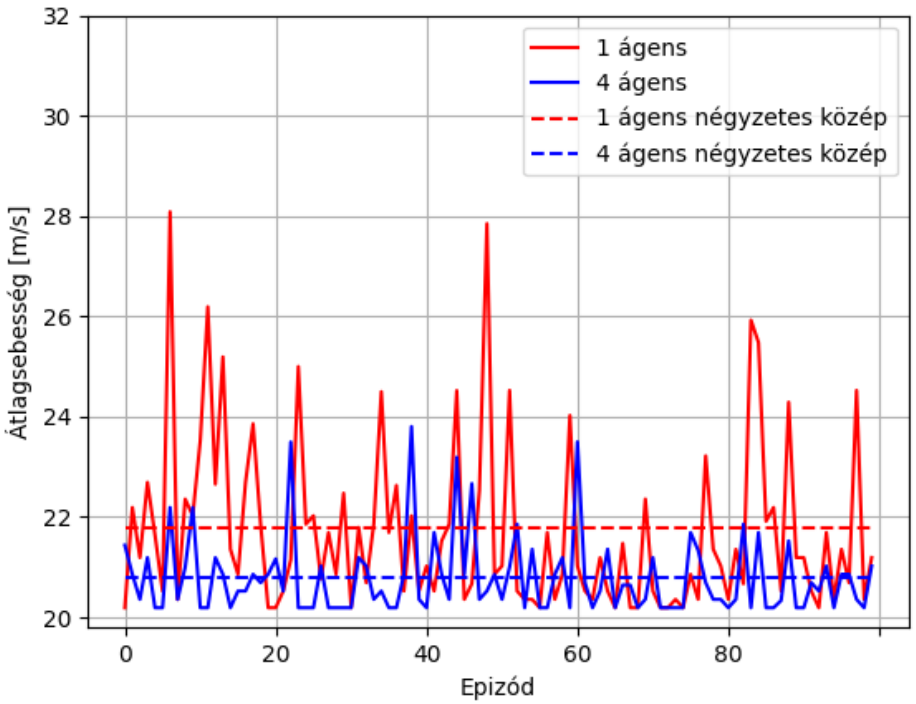
Ezt követően a kibővített állapottérrel tanított ágensek teljesítményét vizsgáltam, melyet szintén a korábban bemutatott teszttel végeztem el. A kiértékelés eredményeit az 5.2 táblázat és az 5.2 ábra mutatja be.

5.2. táblázat. Teszteredmények a kibővített állapottér használatakor

Ágensek száma (kiértékelés)	Ágensek száma (tanítás)	Ütközések	Átlagos jutalom
1	1	7	20,91
	4	0	22,33
4	1	15	20,32
	4	2	22,33
8	1	45	17,01
	4	16	21,02
12	1	79	12,42
	4	51	17,49

Az állapottér bővítésével mindkét algoritmus jobb eredményeket ért el a második és a harmadik tesztelési esetben. Az átlagos jutalmak ugyan kis mértékben csökkentek, de az ütközések számát tekintve figyelemre méltó javulás mutatkozik. Azonban észrevehető, hogy a kiegészítő információk csak egy bizonyos ágensszámig biztosítanak előnyt. Ezt az értéket egy bizonyos határérték felé emelve már nem jelentkezik javulás a teljesítményben. Emellett az első (egy ágenst tartalmazó) tesztelési esetben érdekes módon az ütközések száma nem változott jelentősen a korábbi eredményekhez képest, azonban az átlagos jutalom csökkent, mely az óvatosabb stratégiának tudható be. Mindkét algoritmus megtanulta felhasználni a kiegészítő információkat, melyre azonban csak a többi ágens viselkedésének becslésére, valamint az ehhez történő alkalmazkodáshoz van szükségük. Az utolsó tesztelési esetben pedig látható, hogy a kritikus ágensszám túllépésekor már nem érhető el növekedés az átlagos jutalmakban vagy csökkenés az ütközések számában.

Az ütközések számát a két tanítási halmaz során vizsgálva elmondható, hogy a kívánt irányszögek szinuszának ($\sin d$) és koszinuszának ($\cos d$) a megfigyelési térhez történő hozzáadása a multi-ágens "policy" esetén jelentősebb hatást fejtett ki. Számszerűsítve, a második tesztelési esetben az egyágenses "policy" ütközéseinek száma 30-ról 15-re, azaz 50%-kal csökkent, míg multi-ágens esetben ez az érték 14-ről 2-re változott, azaz 85,71%-kal csökkent. Mindez a multi-ágens tanítás előnyeit mutatja, mivel az ágens a tanulási folyamat során megismerhette a többi ágens által irányított jármű viselkedését, szándékait, így ehhez alkalmazkodni tudott. Ezen felül az említett információátadás az ágensek szándékairól a kommunikáció előnyeit is megmutatja, habár ez ennek csak egy kezdetleges formájaként fogható fel.



5.2. ábra. Az első ágens átlagos sebességei kibővített állapottér esetén

Az 5.2 ábrán látható átlagsebességek az előző teszthalmazhoz hasonló tendenciát mutatnak. Az egyágenses "policy" hajlamos bátrabban cselekedni, azaz magasabb sebességeket választani, mely azonban néha ütközéshez vezet, így az átlagos jutalom csökken, valamint az epizódok jutalmai ingadozóvá válnak. Az eredeti állapottér esetén tapasztalt jutalomingadozás mértéke mindkét esetben csökkent, mely a multi-ágens "policy" esetén jelentősebb. Ekkor az átlagos jutalmak szinte kivétel nélkül 22 és 24 közötti értéket vesznek fel.

Ahogy az korábban is említésre került, a multi-ágens "policy" teljesen kompetitív jutalmazással lett tanítva, azaz az egyes ágensek külön kapták a jutalmat, saját cselekedeteik alapján. Ezt amiatt fontos kiemelni, mivel ez alapján önzőbb viselkedés lenne elvárható az ágensektől, azonban az ütközések büntetésének, valamint a tanítás közbeni változatos viselkedések megfigyelésének köszönhetően a multi-ágens algoritmus egy robusztusabb stratégia megtanulására volt képes, mely a vizsgált szempontok alapján összességében jobb eredményt ért el.

5.3. A jutalom módosítása

Az ütközések számából kiindulva az esetleges további csökkentés érdekében az ütközés esetén adott büntetés értékét -1-ről -2-re változtattam, így annak a jutalomfüggvényben való hatását növeltem. Emellett a tanulás sebességét befolyásoló "learning rate", valamint a

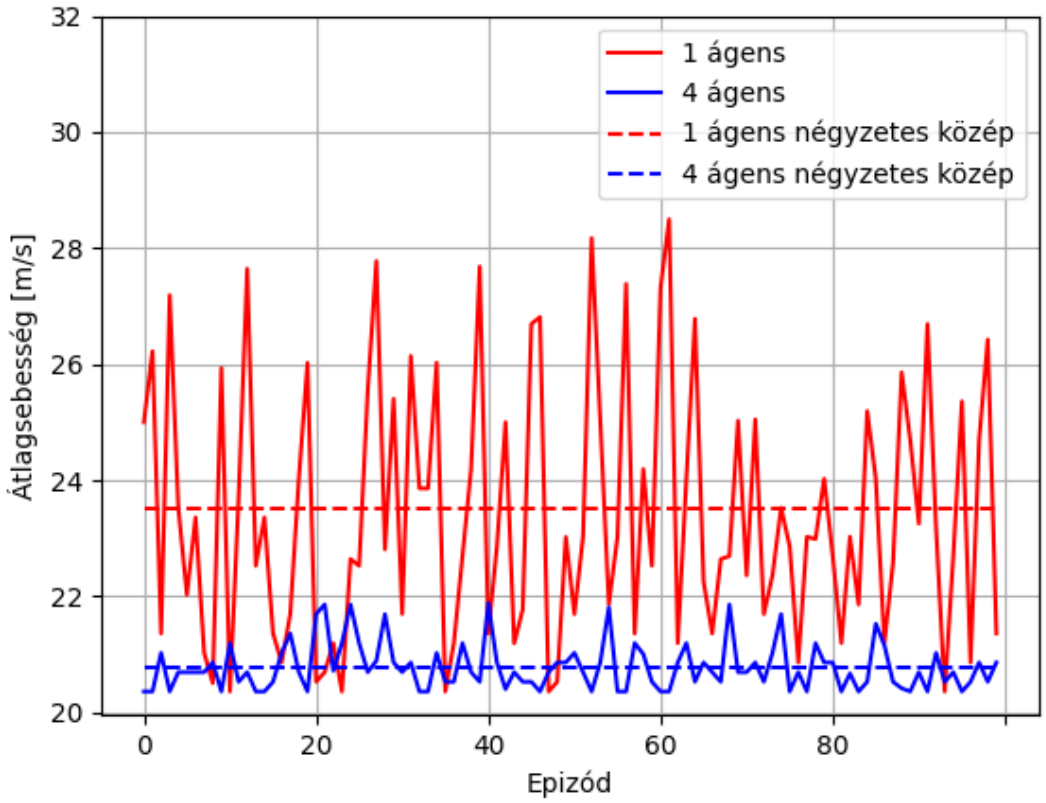
"target" Q-hálózat frissítési gyakoriságát megadó "sync every" paraméter módosításának hatásait is vizsgáltam. Mindegyik tanítás során a korábban bemutatottak szerint előnyös kibővített állapotteret alkalmaztam. A különböző paraméterekkel végzett tanítások teszteredményeit az 5.3 táblázat összegzi.

5.3. táblázat. Teszteredmények a jutalmazás változtatásakor

Learning rate	Sync every	Ágensek száma (tanítás)	Ágensek száma (kiértékelés)	Ütközések	Átlagos jutalom
0,0001	20	1	1	1	25,77
			4	20	23,76
			8	46	20,03
			12	66	16,62
		4	1	1	25,09
			4	7	24,77
			8	13	23,82
			12	52	18,55
0,001	20	1	1	16	25,91
			4	54	19,80
			8	77	15,23
			12	65	16,66
		4	1	2	25,09
			4	12	23,39
			8	32	20,99
			12	63	16,19
0,001	10	1	1	8	24,44
			4	20	22,88
			8	52	18,85
			12	79	13,88
		4	1	6	24,59
			4	11	24,07
			8	28	21,62
			12	66	16,25

A táblázat első sora azt az esetet képviseli, amikor csak az ütközési büntetés értéke változott. Ekkor az ütközések számában bizonyos esetekben javulás lép fel, a többi esetben viszont rosszabbodik a teljesítmény. A jutalmak értéke ugyan növekszik, azonban ez a normalizálásban történő értékváltozás miatt nem eredményez jobb teljesítményt. Erről árulkodik az 5.3 ábra is, ami összevetve az 5.2 ábrán látható diagrammal hasonló átlagos sebességeket eredményez. Ez alapján kijelenthető, hogy önmagában az ütközések nagyobb mértékű büntetése nem hozott jelentős változást az ágensek teljesítményében.

A tanulási ráta növelése által a tanulás során elért jutalmak valamivel hamarabb konvergáltak egy bizonyos értékhez, ami azonban így a teljesítmény csökkenését eredményezte. A romló tendencia a táblázat második sora alapján is észrevehető. Az ütközések



5.3. ábra. Az első ágens átlagos sebességei módosított jutalmazás esetén

száma ekkor jelentősen nő, valamint ebből kifolyólag az átlagjutalmak is csökkennek. A gyakoribb frissítés az "online" és a "target" Q-hálózat között a táblázat harmadik sora alapján bizonyos mértékben javította az ágens teljesítményét, azonban ezen módosítással sem volt képes felülmúlni az ágens a korábban bemutatott algoritmusok eredményét.

6. fejezet

Konklúzió

6.1. Elért eredmények

Ebben a dolgozatban az egy- és többágenses Megerősítéses Tanulás közti különbségek kerültek bemutatásra egy autópályás döntéshozatali helyzetben. A tanított ágensek teljesítménye különböző számú ágens által irányított járművet tartalmazó környezetekben került kiértékelésre. A multi-ágens módszer jelentősen kevesebb ütközést eredményezett az egyágenses algoritmushoz képest. Továbbá a jövőbeli szándékokkal kiegészített megfigyelési tér hatásai is vizsgálatra kerültek. Ez az ütközések számának további csökkenését, valamint a jutalmak és sebességek bizonyos átlagérték körüli stabilizálódását eredményezte. Az ütközések számának további csökkentése érdekében vizsgálatra került az ütközések nagyobb mértékű büntetésének, valamint a különböző tanulási paraméterek használatának hatása, melyek azonban nem vezettek a teljesítmény javulásához.

Az eredményekkel kapcsolatban fontos hangsúlyozni, hogy az ismertetett algoritmusok teljesítménye jelen állapotban még nem alkalmas a való életben történő alkalmazásra, hiszen a valós forgalmi szituációkban 100 esetből még két ütközés is elfogadhatatlanul magas érték. Azonban az algoritmusokat összehasonlítva egyértelműen megállapítható a MARL előnye a döntéshozatali szituációk kezelésében.

6.2. Jövőbeli lehetőségek

A tanított ágensek teljesítményének javítására több lehetőség is fennáll. Ezek közül a legalapvetőbb a hálózat méretének növelése, valamint a fejlettebb algoritmusok használata. Mindez azonban a számítási kapacitásigényt, így ezzel a jelenleg is hosszú tanulási időt növelné. Emiatt elsősorban más fejlesztési módszereket kívánok alkalmazni. A való életben fellépő forgalmi szituációk lehető legpontosabb modellezése érdekében célszerű a környezet paramétereinek (pl. járművek száma, sávok száma) véletlenszerű értékre történő felvétele. Mindemellett a terveim közt szerepel a "Curriculum Learning" tanítási módszer alkalma-

zása is, mely a [43] által megfogalmazott "start small" elven alapul. Eszerint a könnyű feladatokkal való kezdés, majd azok nehezítése az emberek és az állatok tanításában elért sikereken kívül a gépi tanulásban is hasznos lehet. Ebből kiindulva [44] definiálta a "Curriculum Learning" fogalmát. Ekkor a tanítás kezdetén a tanítómintákból nagyobb valószínűséggel választunk a könnyebb feladatokból, majd a tanítás előrehaladtával fokozatosan növeljük a nehezebb feladatok választási valószínűségét, míg végül a tanítás végén azonos súlyozással kerül kiválasztásra a feladat. Mindez a tanítás felgyorsítását, valamint az ágensek teljesítményének javulását eredményezheti. A módszer [45] alapján széles körökben került alkalmazásra, például gépi látás (computer vision), Természetes Nyelvek Feldolgozása (Natural Language Processing), vagy Megerősítéses Tanulás esetén. A fokozatosan nehezedő feladatok meghatározására többféle módszer is létezik. Jelen esetben a nehezítést a környezetben szereplő ágensek számának növelésével szeretném megvalósítani.

Irodalom

[1] *Személysérüléses Közúti Közlekedési balesetek, 2022. I. Negyedév*. letöltve: 2023.09.08. URL: https://www.ksh.hu/docs/hun/xftp/stattukor/bal/20221/index.html#tovbbiadatokinformcik.

[2] Ekim Yurtsever és tsai. „A survey of autonomous driving: Common practices and emerging technologies". *IEEE access* 8 (2020), 58443–58469. old.

[3] Mrinal Bachute és Javed Subhedar. „Autonomous Driving Architectures: Insights of Machine Learning and Deep Learning Algorithms". *Machine Learning with Applications* 6 (2021. dec.), 100164. old. DOI: 10.1016/j.mlwa.2021.100164.

[4] Firoz Khan és tsai. „Autonomous vehicles: A study of implementation and security." *International Journal of Electrical & Computer Engineering (2088-8708)* 11.4 (2021).

[5] Max Peter Ronecker és Yuan Zhu. „Deep Q-network based decision making for autonomous driving". *2019 3rd international conference on robotics and automation sciences (ICRAS)*. IEEE. 2019, 154–160. old.

[6] Asif Faisal és tsai. „Understanding autonomous vehicles". *Journal of transport and land use* 12.1 (2019), 45–72. old.

[7] Sorin Grigorescu és tsai. „A survey of deep learning techniques for autonomous driving". *Journal of Field Robotics* 37.3 (2020), 362–386. old.

[8] Volodymyr Mnih és tsai. „Playing atari with deep reinforcement learning". *arXiv preprint arXiv:1312.5602* (2013).

[9] Timothy P. Lillicrap és tsai. *Continuous control with deep reinforcement learning*. 2019. arXiv: 1509.02971 [cs.LG].

[10] Nahid Parvez Farazi és tsai. „Deep reinforcement learning and transportation research: A comprehensive review". *arXiv preprint arXiv:2010.06187* (2020).

[11] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. SAE J3016. SAE International, 2021. DOI: https://doi.org/10.4271/J3016_202104.

[12] Jinwoo Ha és Dongsoo Kim. „Exploring acceptance of autonomous vehicle policies using KeyBERT and SNA: Targeting engineering students". *arXiv preprint arXiv:2307.09014* (2023).

[13] Abdoulaye O Ly és Moulay Akhloufi. „Learning to drive by imitation: An overview of deep behavior cloning methods". *IEEE Transactions on Intelligent Vehicles* 6.2 (2020), 195–209. old.

[14] Martin Treiber, Ansgar Hennecke és Dirk Helbing. „Congested traffic states in empirical observations and microscopic simulations". *Phys. Rev. E* 62 (2 2000. aug.), 1805–1824. old. DOI: 10.1103/PhysRevE.62.1805. URL: https://link.aps.org/doi/10.1103/PhysRevE.62.1805.

[15] Xin Li, Xin Xu és Lei Zuo. „Reinforcement learning based overtaking decision-making for highway autonomous driving". *2015 Sixth International Conference on Intelligent Control and Information Processing (ICICIP)*. 2015, 336–342. old. DOI: 10.1109/ICICIP.2015.7388193.

[16] Edouard Leurent és Jean Mercat. *Social Attention for Autonomous Decision-Making in Dense Traffic*. 2019. arXiv: 1911.12250 [cs.LG].

[17] Irwan Bello és tsai. *Neural Combinatorial Optimization with Reinforcement Learning*. 2017. arXiv: 1611.09940 [cs.AI].

[18] Angelos Mavrogiannis, Rohan Chandra és Dinesh Manocha. „B-gap: Behavior-rich simulation and navigation for autonomous driving". *IEEE Robotics and Automation Letters* 7.2 (2022), 4718–4725. old.

[19] Dong Chen és tsai. *Deep Multi-agent Reinforcement Learning for Highway On-Ramp Merging in Mixed Traffic*. 2022. arXiv: 2105.05701 [eess.SY].

[20] Meha Kaushik, Phaniteja S és K. Madhava Krishna. *Parameter Sharing Reinforcement Learning Architecture for Multi Agent Driving Behaviors*. 2018. arXiv: 1811.07214 [cs.LG].

[21] Dániel Tamás Gujgiczer, Ádám Szabó és Tamás Bécsi. „Egy- és multiágenses Megerősítéses Tanulás összehasonlítása HighwayEnv környezetben". *IFFK 2023: XVII. Innováció és fenntartható felszíni közlekedés*. 2023.

[22] M. I. Jordan és T. M. Mitchell. „Machine learning: Trends, perspectives, and prospects". *Science* 349.6245 (2015), 255–260. old. DOI: 10.1126/science.aaa8415. eprint: https://www.science.org/doi/pdf/10.1126/science.aaa8415. URL: https://www.science.org/doi/abs/10.1126/science.aaa8415.

[23] Sotiris B Kotsiantis, Ioannis Zaharakis, P Pintelas és tsai. „Supervised machine learning: A review of classification techniques". *Emerging artificial intelligence applications in computer engineering* 160.1 (2007), 3–24. old.

[24] Yuxi Li. *Deep Reinforcement Learning*. 2018. arXiv: 1810.06339 [cs.LG].

[25] Richard S Sutton és Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[26] Aakash Maroti. *RBED: Reward Based Epsilon Decay*. 2019. arXiv: 1910.13701 [cs.AI].

[27] Yann LeCun, Yoshua Bengio és Geoffrey Hinton. „Deep learning". *nature* 521.7553 (2015), 436–444. old.

[28] Vincent François-Lavet és tsai. „An introduction to deep reinforcement learning". *Foundations and Trends® in Machine Learning* 11.3-4 (2018), 219–354. old.

[29] Christopher John Cornish Hellaby Watkins. „Learning from delayed rewards". (1989).

[30] Volodymyr Mnih és tsai. „Human-level control through deep reinforcement learning". *nature* 518.7540 (2015), 529–533. old.

[31] Hado Van Hasselt, Arthur Guez és David Silver. „Deep Reinforcement Learning with Double Q-Learning". *Proceedings of the AAAI conference on artificial intelligence*. 30. köt. 1. 2016.

[32] Sven Gronauer és Klaus Diepold. „Multi-agent deep reinforcement learning: a survey". *Artificial Intelligence Review* (2022), 1–49. old.

[33] Michael L. Littman. „Markov Games as a Framework for Multi-Agent Reinforcement Learning". *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*. ICML'94. New Brunswick, NJ, USA: Morgan Kaufmann Publishers Inc., 1994, 157–163. old. ISBN: 1558603352.

[34] Marco A Wiering és Martijn Van Otterlo. „Reinforcement learning". *Adaptation, learning, and optimization* 12.3 (2012), 729. old.

[35] Lucian Busoniu, Robert Babuska és Bart De Schutter. „A comprehensive survey of multiagent reinforcement learning". *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38.2 (2008), 156–172. old.

[36] Peter Sunehag és tsai. „Value-decomposition networks for cooperative multi-agent learning". *arXiv preprint arXiv:1706.05296* (2017).

[37] J. K. Terry és tsai. *Parameter Sharing For Heterogeneous Agents in Multi-Agent Reinforcement Learning*. 2022. arXiv: 2005.13625 [cs.LG].

[38] Ming Tan. „Multi-agent reinforcement learning: Independent vs. cooperative agents". *Proceedings of the tenth international conference on machine learning*. 1993, 330–337. old.

[39] Jayesh K. Gupta, Maxim Egorov és Mykel Kochenderfer. „Cooperative Multi-agent Control Using Deep Reinforcement Learning”. *Autonomous Agents and Multiagent Systems*. Szerk. Gita Sukthankar és Juan A. Rodriguez-Aguilar. Cham: Springer International Publishing, 2017, 66–83. old. ISBN: 978-3-319-71682-4.

[40] Edouard Leurent. *An Environment for Autonomous Driving Decision-Making*. `https://github.com/eleurent/highway-env`. 2018.

[41] Philip Polack és tsai. „The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?”: *2017 IEEE intelligent vehicles symposium (IV)*. IEEE. 2017, 812–818. old.

[42] Arne Kesting, Martin Treiber és Dirk Helbing. „MOBIL: General lane-changing model for car-following models”. *Proceedings of the Transportation Research Board Annual Meeting*. 2006.

[43] Jeffrey L Elman. „Learning and development in neural networks: The importance of starting small”. *Cognition* 48.1 (1993), 71–99. old.

[44] Yoshua Bengio és tsai. „Curriculum learning”. *Proceedings of the 26th annual international conference on machine learning*. 2009, 41–48. old.

[45] Xin Wang, Yudong Chen és Wenwu Zhu. „A Survey on Curriculum Learning”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (2022), 4555–4576. old. DOI: `10.1109/TPAMI.2021.3069908`.

Ábrák jegyzéke

Táblázatok jegyzéke