

Multi-ágens Megerősítéses Tanulás alapú döntéshozás autópályás környezetben

Szerző:

Knáb István Gellért



M Ű E G Y E T E M 1 7 8 2

Közlekedésmérnöki és Járműmérnöki Kar

Konzulens:

Kővári Bálint

TDK dolgozat

Tartalomjegyzék

1. Bevezető	1
1.1. Probléma feltárása	1
1.2. Kapcsolódó kutatások	4
1.3. Megoldási kísérlet	7
2. Környezet	8
2.1. Forgalmi hálózat	8
2.2. Implementáció	9
3. Algoritmusok	11
3.1. Gépi tanulás	11
3.2. Megerősítéssel tanulás	13
3.3. Multi-ágens megerősítéssel tanulás	16
3.4. Mély tanulás	19
3.5. DRL problémáspecifikus implementációja	22
3.5.1. State	23
3.5.2. Action	24
3.5.3. Reward	24
4. Eredmények	26
4.1. Adatok a kiértékeléshez	26
4.2. Összehasonlító analízis	27
5. Konklúzió	31
5.1. Kiértékelés	31
5.2. Fejlesztési irányok	32

Kivonat

A növekvő járműforgalom hatására a telítődő utakon kialakuló tranziensek nagy mértékű kumulatív hibát eredményeznek, ezzel a járműcsoportok mozgásának folytonosságában visszaesést indukálva. Autópályás környezetben ez a jelenség halmozottan hátrányos helyzeteket szül a nagy sebességek miatt, ezért a feloldás érdekében lokális sebességkorlátozások bevezetése szükségeszerű, amivel az adott forgalmi szegmens be- és kilépő járműveinek különbségével lehet jobban terhelni vagy üriteni egy érintett zónát. A rendelkezésre álló infrastruktúra kihasználtságának növelése érdekében számos járműirányítási megközelítés létezik, ami dinamikus sebességkorlátozáson alapul, ezzel térben elosztva az áramló járműveket és csökkentve a torlódás mértékét. Ezeknek a beavatkozási formáknak nagy része elágazásokon alapuló algoritmus, melyek küszöbértékeket átlépve a kialakult mozgó forgalmi hullámok identifikációját követően aktiválnak egyes állapotokat, ezzel feloldva a már kialakult anomáliákat.

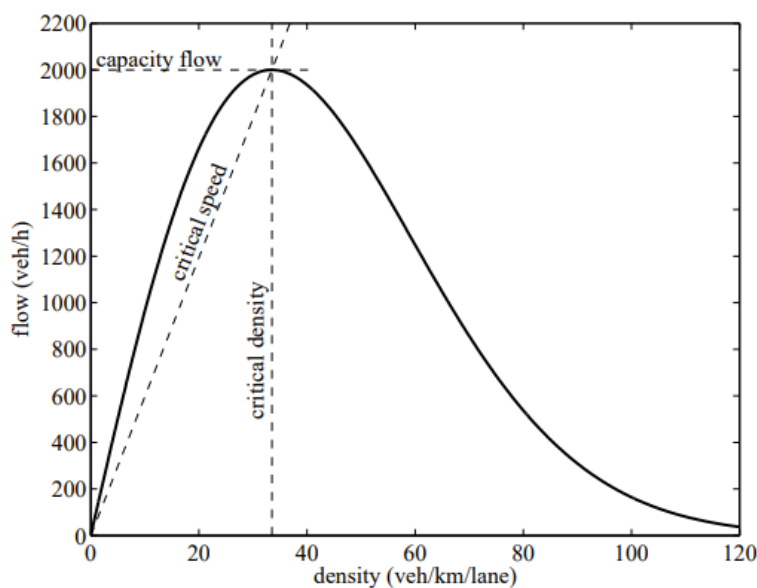
A dolgozatban ezekkel az állapotgépet megvalósító implementációkkal szemben egy Multi-Ágens Megerősítéses Tanuláson alapuló realizáció kerül bemutatásra, ami a Gépi Tanulás fő irányai közül a legkisebb ipari részesedéssel bír, ennek ellenére számos tulajdonsága mégis a korszerű és ígéretes kutatási területek közé emeli. Mély Tanulással ötvözve a Megerősítéses Tanulás célja, hogy a modell identifikációt és a fizikai egyenletek megfogalmazását elkerülve olyan prediktív irányítási struktúrát hozzunk létre, ami nem csupán feloldja, de meg is előzi a shock wave kialakulását.

1. fejezet

Bevezető

1.1. Probléma feltárása

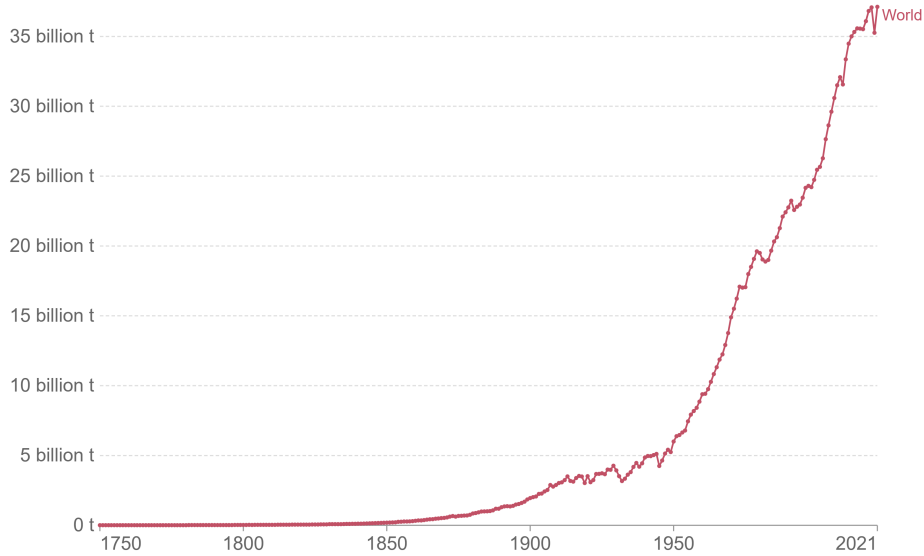
Az urbanizáció következtében a városok népességének növekedésével[1] együtt alakul a szándék a közlekedési hálózatok igénybevételére, ezzel a létrehozott utak maximális kihasználtságát megközelítve, adott esetben meg is haladva azt. A telítettség okán kisebb beavatkozások is olyan tranziensek létrejöttét eredményezhetik, amik az áteresztőképesség csökkenésével visszaesést mutatnak a forgalmi hálózat kapacitásában[2], ezzel a kumulatív jellegű hiba nem csak az adott járműnek, hanem a környezetének dinamikáját is negatívan befolyásolja. A cseppfolyós közegek mozgásának leírásához hasonlóan az áramlástan megmaradási törvényeivel fogalmazhatóak meg a fizikai formalizmusok[5]



1.1. ábra. Fundamentális diagram a kritikus kapacitás szemléltetésére[3]

Annual CO₂ emissions

Carbon dioxide (CO₂) emissions from fossil fuels and industry¹. Land use change is not included.



Source: Our World in Data based on the Global Carbon Project (2022)

OurWorldInData.org/co2-and-greenhouse-gas-emissions • CC BY

1. Fossil emissions: Fossil emissions measure the quantity of carbon dioxide (CO₂) emitted from the burning of fossil fuels, and directly from industrial processes such as cement and steel production. Fossil CO₂ includes emissions from coal, oil, gas, flaring, cement, steel, and other industrial processes. Fossil emissions do not include land use change, deforestation, soils, or vegetation.

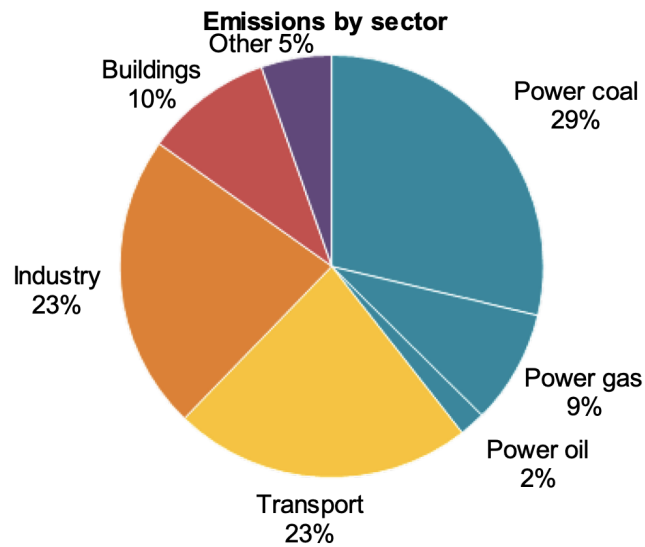
1.2. ábra. Az évenként kibocsájtott CO₂ mennyisége[4]

forgalomirányítás esetére is, kiemelve a kontinuitás tételét[6], ami a keresztmetszeti változások következtében befolyásolja a sebesség és sűrűség szorzatának értékét, mivel lineáris kapcsolatban áll velük. Ennek megfogalmazása egyenlet segítségével az alábbi módon tehető meg:

$$\rho_1 c_1 A_1 = \rho_2 c_2 A_2, \quad (1.1)$$

ahol a c_i értékek a sebességeket, a ρ_i a sűrűséget míg a A_i az áramlási keresztmetszetet jelölik. Az A_i drasztikus csökkenésének esetén, ami keletkezhet útépités okozta sávlezárásokból, balesetekből vagy autópályán felhajtók környezetében, adódik az egyenletből, hogy az áramlás folytonosságának érdekében a szűkülés után a megelőzőnél nagyobb sebesség realizálása szükségszerű, hogy a forgalom ne kerüljön olyan állapotba, ahol a sűrűség növekedése már kontraproduktívan hatna az áramlás mértékére (Sűrűség és áramlás nagysága közötti összefüggés ábrázolása az 1.1 ábrán).

Felmerülő megoldási irány az infrastruktúra térbeli kiterjesztése, ami a sűrűn beépített városok esetében igen korlátozott[7], vagy akár tejllességgel kivitelezhetetlen. Amennyiben mégis eszközölhető, akkor is hátránya az imént említett megközelítésnek, hogy csupán a frekvenciált időszakokban és rendhagyó esetek folyamán lépnek fel a forgalmi gátoltságok, amikor ténylegesen indokolt lenne ez a kiterjedés, így az idő jelentős részében nem lennének kihasználva a környezeti adottságok, ami monetáris oldalról sem

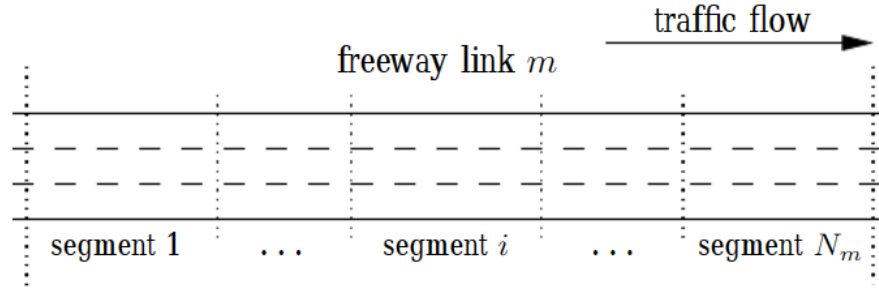


1.3. ábra. A kibocsátás szektoronkénti lebontása [8]

bizonyul jó döntésnek.

Az érintett időszakokban a sávfoglaltság növekedésével együtt változik az üvegházhatású gázok az és a zaj kibocsátása (Ezek közül a CO_2 emisszió tendenciája szemléltetve az 1.2 ábrán), valamint a forgalomban töltött idő is magasabb értékeket ér el, köszönhetően a meddő állapotként interpretálható várakozásnak. Az összképet tekintve (1.3 ábra) a közlekedés jelentős részt vállal az említett gázok emissziójában, ennek következtében alkalmas módszereket bevetve lényeges javulás érhető el ezen a téren, akár százalékos nagyságrendben.

Városfejlesztési szempontból az említett okokból kifolyólag elengedhetetlen és releváns kutatási terület a közlekedésinfrastruktúra korszerűsítése, melynek újragondolása nem csak a fizikai megjelenésben nyilvánul meg, mint új forgalmi sávok és lámpák bevezetése, hanem a járművek emisszió szerinti behajtási szabályozásán valamint a forgalom menedzsmentjén keresztül is érhetőek el előrelépések. Itt nem csak a járművek egyéni mozgásának irányításáról és az önvezető funkciók megvalósításáról van szó, hanem a nagyobb folyamatként értelmezhető tranziensek elkerülése végett a járművek közötti kapcsolatról, ahol egyes döntéseknek már nem az egyénszintű trajektóriatervezés, hanem egy járműoszlopot megcélzó stratégiai jellegű döntéshozás a célja. Számos megoldás létezik hagyományos és korszerű eseményvezérelt algoritmusokra is, melyek egy része alkalmaz olyan adaptív járműirányítási rendszereket, mint városi környezetben a Traffic Signal Control (TSC)[10] igény szerint kapcsolható jelzőlámpákkal, vagy autópályák esetére a Variable Speed Limit Control(VSLC)[11][12], ami a cellákra felosztott útszakaszon beavatkozva lassítja a telített zónák beáramlását(1.4 ábra). Előnyük

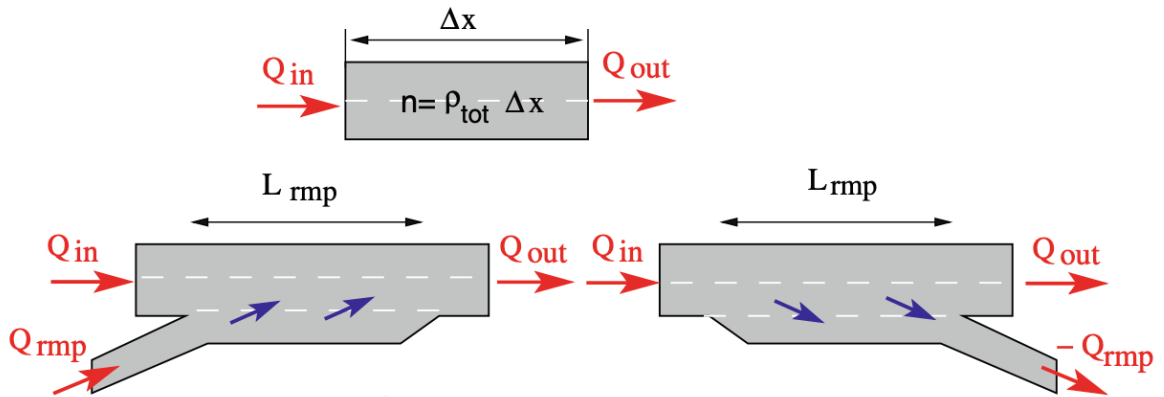


1.4. ábra. Variable Speed Limit Control számára felosztott útszakasz [9]

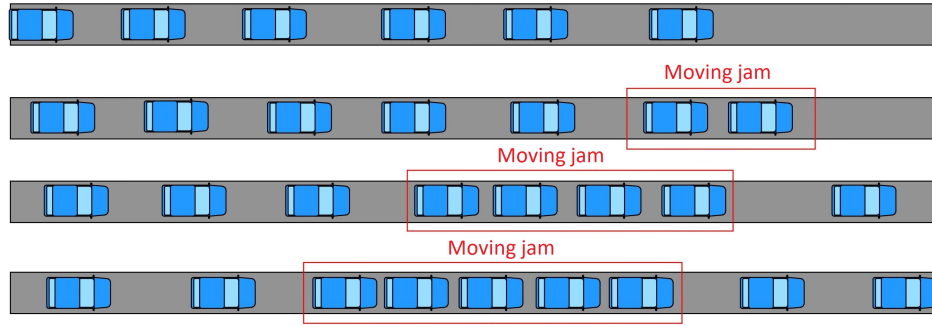
ahogy a neve is mutatja, adaptálódnak a forgalom aktuális állapotához, ezzel egyfajta intelligens közlekedési rendszert (intelligent transportation system - ITS)[13] kialakítva. Az ITS jelentősége együtt változik a technológia fejlődésének felgyorsulásával, és a növekvő számú rendelkezésre álló szenzoradat olyan algoritmusok használatát is lehetővé teszi, ami nagyobb adatigénnyel rendelkezik az egyes állapotokból valamilyen beavatkozás létrehozására.

1.2. Kapcsolódó kutatások

Korszerű autópályás hálózatokon, mint a holland és svéd[11] rendszerek megtalálhatóak a VSLC-n alapuló irányítási modellek, amik az áramlás térbeli elosztásának segítségével avatkoznak be az időbeli disztribúcióba, az egyes szegmensek (1.4 ábra) sebességhatárait keresztül. A működését tekintve hasonló a statikus sebességhatározó táblákhoz, annyi eltéréssel, hogy a környezet monitorozásának segítségével egy valós idejű eseményvezérelt módon alakulnak ki az egyes útszegmensek megengedett maximális sebességei állítható értékű táblákon közölve a járművezetővel. A 1.5 ábrán



1.5. ábra. Kontinuitás tétele a forgalom esetére [6]



1.6. ábra. Mozgó forgalmi hullám szemléltetése

szemléltetett módon az egyes szegmensekből kiáramló járművek száma megegyezik az őt követő szektor beáramlásának mértékével, így egy útszakaszt nézve a be- és kilépő járművek különbségének függvényében telítődik vagy ürül a zóna, ami a környező és aktív szektor sebességhatárát különbségein keresztül valósulhat meg. A módszer mögött számos algoritmus húzódik meg, melyek közül a Motorway Control System (MCS)[11] került alkalmazásba a fejlebb említett országokban, viszont a szakirodalom gyakran tárgyalja a SPEed ControllIng ALgorIthm using Shock wave Theory (SPECIALIST) [14][15] névre hallgató algoritmust is. Ezek a megoldások egyaránt a már detektáltan létrejött shock wave feloldására kerültek implementálásra, ami a forgalmi tranziensek kumulatív jellegéből adódóan alakul ki, lassítás esetén az összeadódó reakcióidők hatására megjelenő mozgó forgalmi hullámok formájában (1.6 ábra), az alábbi módon:

$$\tau = \sum_{i=1}^n \tau_i, \quad (1.2)$$

ahol az i a vizsgált járműoszlop hossza, míg a τ értékek az egyes járművek várakozási idejét jelölik. Abból adódóan, hogy a már kialakult anomália kezelése a cél, valamilyen mértékben megjelennek ezek a késleltetések a hálózaton, viszont lecsengési idejük csökkenthető ezeknek az algoritmusoknak a használatával. A következőkben röviden tárgyalva ezeket az algoritmusokat megmutatkozik alkalmazásuk előnye, de láthatóvá válik az is, hol szorulnak fejlesztésre.

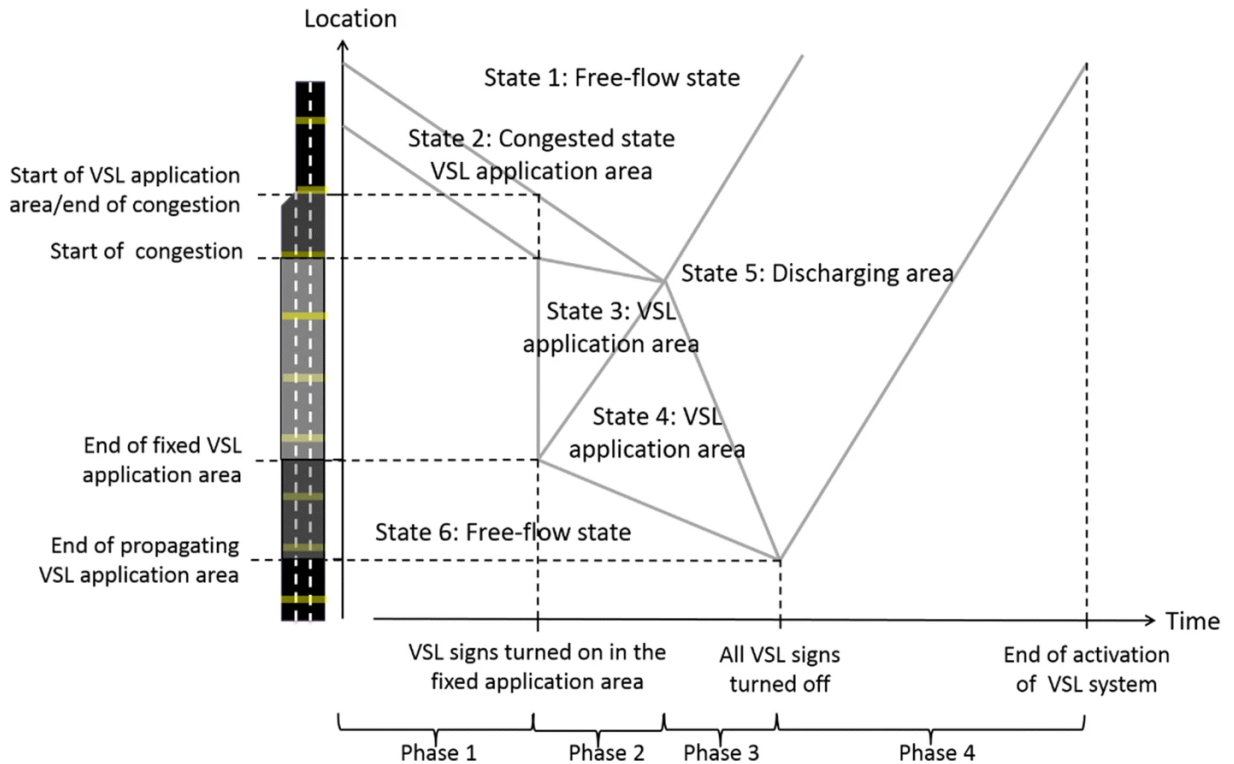
Az MCS egyszerű elágazáson alapuló algoritmus, ami az átlagsebességet megfigyelve avatkozik be az érintett forgalmi szegmensekben. Amennyiben ez egy adott küszöbérték alá csökken, a kialakult forgalmi hullám gyorsabb lecsengésének érdekében kerülnek bevezetésre a korlátozások a szenzoradatokat szolgáltató valamint az őt követő két célalban, ezzel csökkentve a beáramló járművek számát és elősegítve a kiürülést. Mivel

1. Algorithm MCS [11]

```
if  $v_{javg} \leq 45km/h$  then
  Speed limit at detector station  $v_{t,j} = 60km/h$ 
   $v_{t,j-1} = 80km/h$ 
   $v_{t,j-2} = 100km/h$ 
else
   $v_{t,j} = 120km/h$ 
   $v_{t,j-1} = 120km/h$ 
   $v_{t,j-2} = 120km/h$ 
end if
```

csak 2 állapotú a rendszer, nem képes finom különbségek detekciójára és megelőző beavatkozásra, így teljesítménye csupán csekély javulást eredményez. További hátránya ahogy a kiértékelés során szemléltetve lesz, hogy a megadott szegmenshosszak nagy mértékben befolyásolják a működését, amin a megválasztott sebességhatárok értékei tudnak javítani.

A SPECIALIST ezzel szemben egy szofisztikáltabb rendszer, ami 2 helyett 6 állapot (1.7 ábra) alapján határozza meg a forgalom aktuális helyzetét, ezzel egy pontosabb leírást biztosítva a köztes állapotok megkülönböztetésének köszönhetően. Teljesítményében is megmutatkozik alkalmassága az imént említett algoritmussal szemben, viszont figyelembe kell venni hogy nagyobb mennyiségű szenzoradat szükséges az imple-



1.7. ábra. A SPECIALIST fázisai és állapotai [11]

mentációja során, hiszen nem csak az átlagsebességek, hanem az áramlás nagysága is meghatározó szerepet játszik az egyes állapotok leírásában.

Abból kifolyólag, hogy az említett megoldások már identifikált forgalmi hullámok kezelésére készültek, azzal a céllal, hogy az érintett zónák minél előbb ki legyenek ürítve, a kumulatív természetű várakozási idők nem küszöbölhetőek ki, hiszen ezeknek a megelőzésére egy predikciót kéne biztosítani a döntéshozás számára, ami nem csak feloldja, hanem meg is előzi a kialakuló shock wave effektust. Ennek az előzetekintésnek a hiányában mindenképpen létrejönnek a mozgó forgalmi hullámok, a kérdés csupán az élettartamuk és a befolyásolt járművek száma.

1.3. Megoldási kísérlet

A megvalósult alkalmazások említett hiányosságának elkerülése érdekében olyan algoritmus implementálása szükséges, ami a végrehajtott predikció segítségével a következő állapotokat is szem előtt tartva lépi meg az egyes beavatkozásokat. Általános mérnöki szemlélet a fizikai egyenletek megfogalmazása, ezzel olyan modelleket építve, amik matematikailag alátámasztott módon vállalnak garanciát a működésre, a modell identifikáció pontosságának függvényében közelítve a valóságot. Ezzel szemben egyre népszerűbbek a műszaki alkalmazásban is az adat alapú döntéshozások, amik más jellegű szabályszerűségek során határoznak meg kimenetet adott jellemzők hatására.

Az első csoport egyik korszerű képviselője a Model Predictive Control (MPC), ami az állapotterez leírás segítségével létrehozott költségfüggvényt véges és végtelen horizontra bontva határozza meg a megfelelő beavatkozó jelet. A másik csoportba tartozó gépi tanulás alapú algoritmusok egy része, mint a projektben tárgyalt alkalmazás is neurális hálózat segítségével teremt kapcsolatot a kimenetek és bemenetek között, ezzel helyettesítve a bonyolult identifikációs feladatot és fizikai formalizmusokat. Különösen fontos azokban az esetben ez az irány, ahol olyan szintű véletlenszerűség fordul elő a járművek hálózaton való megjelenésében (egyes járművek más más dinamikai paraméterekkel, nem közelíthető egy kamion dinamikája a személygépjármű leírásához), hogy ez csupán átlagértékek segítségével lenne megadható, amik már jelentős hibával közelítik a valóságot.

A kutatás létrehozásának fő motivációja egy olyan ITS implementációja, ami a megjelölt minőségi jellemzők terén képes előrelépést felmutatni a jelenleg alkalmazásban lévő algoritmusokhoz képest, kihasználva a rendelkezésre álló szenzoradatokat, kibővíve olyan makroszkópikus jellemzők monitorozásával, amik elengedhetetlenek a forgalom állapotának leírásához, ezzel az infrastruktúrában rejlő adottságokat magasabb szinten kihasználva, eredményül egy élhetőbb várost és agglomerációt kialakítva.

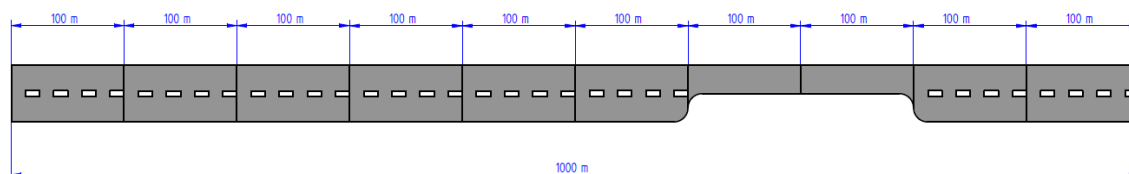
2. fejezet

Környezet

2.1. Forgalmi hálózat

A járműirányítás során számos cél mentén fogalmazhatóak meg feladatok, azt mérlegelve, hogy egyéni optimalizációs problémáról van szó, vagy egy járműcsoport dinamikáját hivatott befolyásolni a választott algoritmus. Ennek a gondolatnak a mentén a kialakuló csoportok a mikroszkópikus, mezoszkópikus és makroszkópikus modellek[16], melyek az egyén szintjéről ebben a sorrendben haladnak a forgalom stratégiai döntéshozására alkalmas modell felé. Mivel egy több szegmenst magába foglaló hálózat forgalmának irányítására készült implementáció a dolgozat témája, nem az egyes járműtrajektóriák megtervezése a cél, így az átlagos járműdinamikai paraméterek kerülnek felhasználásra a makroszkópikus modellen keresztül[17] [18], ezzel teljes mértékben eleget téve az algoritmus szenzoradat igényeinek.

A környezet kialakításában a fő irányelv a valós forgalmi szituációk és szabályszerűségek minél pontosabb leképezése. Ennek okán bizonyos megkötésekkel kell élni a vizsgált útszakasz és az áramló járművek mértékének létrehozásában és annak korlátozó tényezőiben. Az útszakasz terhelése kulcskérdés a problémában, hiszen ha nem egy frekvenciált időszak kerül modellezésre, az algoritmus használata teljes mértékben indokolatlan. Ebből kifolyólag a járművek óránkénti eloszlás szerint jelennek meg a



2.1. ábra. Vizsgált útszakasz

hálózaton olyan mértékben, hogy a szűkülés környezetében garantáltan kialakuljon a shock wave, ezzel az algoritmusok használatának létjogosultságát szemléltetve. A beavatkozási szakasz előtt az utat elő kellett készíteni valós helyzetekre, így a hálózat elejére generált járművek egy ideig irányítás nélkül áramolhattak, annak okán, hogy létrejöjjenek azok az anomáliák a telítődés folyamán, amiknek során már indokolt az algoritmusok használata.

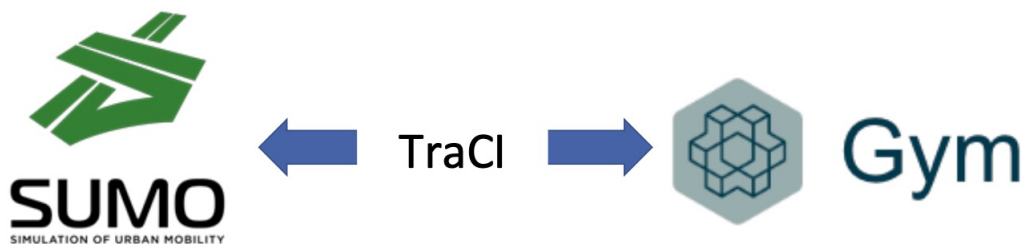
Szükségszerű a járművek egyéni megkötéseinek tisztázása, mint a sávváltások engedélyezésének bevezetése, ugyanis autópályás környezetben a keresztirányú dinamika nagy részét ez teszi ki, valamint a külső sáv lezárása miatt az ott érkező járműveknek szükséges besorolniuk. Emellett a megengedett maximális sebességkorlátozás is részét képezi azoknak a megkötéseknek, amik biztonsági szempontból nem haladhatóak meg, ennek a magyar autópályákon használatos 130 km/h lett megadva. Abból fakadóan, hogy a makroszkópikus modellből kinyert jellemzők segítségével sebességkorlátok lesznek meghatározva, szükséges a beavatkozások sűrűségét is megadni, hiszen az emberi reakcióidő és a jármű tehetetlenségének okán a sebesség megváltozása is olyan folyamat, ami nem közvetlenül az irányító jel közlése után valósul meg. Ennek kiküszöbölésére a TSC-ből ismert minimális fázisidő mintájára itt 10 másodpercenként van beavatkozási lehetősége az algoritmusnak, és azt is csak inkrementális jelleggel 10 km/h változással hozhatja létre, ezzel kiküszöbölve a hirtelen váltásokat. Ennek a lépésköznek a sebességkorlátokban egy további oka, hogy az implementáció során mindenképp olyan értékek adása szükséges, ami a használatban lévő forgalmi táblákon előfordul, ezzel a járművezetőknek könnyen betartható és megszokott feladatot adva.

Ezeket figyelembe véve az implementációt egy olyan szimulátorban kell elvégezni, ami a makroszkópikus jellemzőket mint szenzoradat szolgáltatja és a Pythonnal kompatibilis interfésszel rendelkezik, ezzel kapcsolatot teremtve az ágens és a szimulációs környezet között.

2.2. Implementáció

Az előző részben megfogalmazott igényt a szimulátorra a SUMO (Simulation of Urban Mobility) [19] nyílt forráskódú szimulációs szoftver fogja kielégíteni, ami alkalmas mind mikroszkópikus, mind mezoszkópikus és makroszkópikus forgalommodellezésre. Annak ellenére, hogy multimodális közlekedés is modellezhető benne, ez az implementáció szigorúan a személygépjocsi osztályra támaszkodva készül, és mivel az alapvető egységnek nem a járművek számítanak hanem az áramlás, egy makroszkópikus jellemzőkből összeálló modell kerül alkalmazásra.

A korábban felmerülő feladatot, ami a szimulátor és a python környezet között te-



2.2. ábra. A TraCI működése

remt kapcsolatot a TraCI (Traffic Control Interface)[20] végzi el. A pythonhoz készült csomagjában számos függvény segíti a forgalmi indikátorok lekérhetőségét, jármű és forgalomszinten is, melybe szintén ezeken a szinteken beavatkozást is biztosít.

A későbbiekben tárgyalt Megerősítéses Tanulásra előzetesen kitérve szükséges a neurális hálózat bemeneteit jelentő állapotokat és a rewardokat a környezetben előállítani, hogy az interakció során ezek az adatok elérhetőek legyenek az ágens számára. Bevetett struktúra ezen a téren az OpenAI Gym[21] típusú környezetek használata, aminek sajátossága hogy 3 függvény segítségével teremt kapcsolatot az ágenssel, amik a `step()`, ahol a beavatkozó jellel információt közlünk a környezettel ami erre a szükséges adatokkal visszatér, a `reset()` ahol az egyes epizódok előtt alapállapotba áll a környezet és `render()`, ami a vizuális megjelenítésért felel, jelen esetben szükségtelen, hiszen a SUMO ezt a feladatot is elvégzi. Sok esetben algoritmus összehasonlításra is alkalmazzák, így néhány előre lefejlesztett környezet is található benne mint a Cart Pole, a Lunar Lander vagy a Mountain Car. Visszatérve ennek a python környezetnek a szükségességére, itt valósul meg a szenzoradatokból az állapotrepresentáció összeállítása, a reward kiválasztása és kiszámítása valamint a teszt során felhasznált adatok előkészítése is.

A 2.2 képen látható módon SUMO és a Gym környezet egymással kommunikálva együttesen adja a felhasznált objektumot, ezzel minden eszközt megteremtve a tanításhoz környezet szempontból adatok és vizuális kiértékelés tekintetében.

3. fejezet

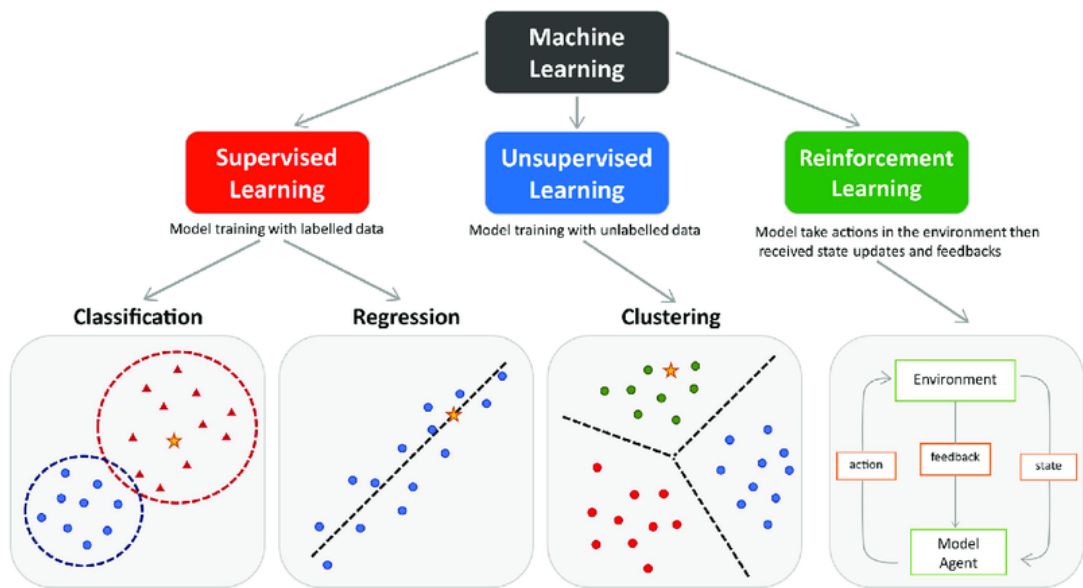
Algoritmusok

3.1. Gépi tanulás

A mesterséges intelligencia (Artificial Intelligence - AI) térnyerése napjaink egyik legrelevánsabb témája műszaki vonalon. Számos olyan feladat elvégezhető ilyen algoritmusokkal, amik normál esetben emberi intelligenciát igényelnének[22], gyakran nagyobb sikerrel, mint klasszikus modell alapú megközelítésekkel. Alkalmazása sokféle területen segíti az emberi munkát, elég itt a nyelvi modellekre, parkolóházakban a rendszámfelismerő rendszerekre vagy egyszerűen a gyakori keresések alapján hangolt böngészőkre gondolni. De komolyabb felelősséggel járó területeket vizsgálva is előkerülnek alkalmazások, mint az orvostudományban, a járművek szenzoradatainak feldolgozásában valamint a haditechnikában. Ennek az egyik alcsoportja a gépi tanulás(Machine Learning - ML)[23], melynek három fő ágát különböztetjük meg, ezek láthatóak a 3.1 ábrán.

Közös tulajdonságuk, hogy mindegyiknek a terméke egy modell, ami a környezetből érkező számszerűen reprezentálható információ segítségével állít elő valamilyen kimenetet, nem a hagyományos fizikai összefüggéseket felhasználva (white-box megközelítés)[24], hanem black-box módon tabuláris megoldásokat, fakeső algoritmusokat vagy neurális hálózatokat bevetve empirikus úton. Ennek az adat alapú döntéshozásnak nagy előnye, hogy fizikailag nehezen megfogalmazható problémáknak csupán néhány tulajdonságát monitorozva az előálló beavatkozó jelet nem terheli identifikációból származó hiba, hiszen a szenzoradatok és beavatkozások közötti modell a tanítás során dinamikusan alkalmazkodik a környezetéhez, nem egyenletekkel közelítve, hanem mindig visszajelzés alapján az aktuális teljesítményt kiértékelve.

Az egyes ágak tárgyalását a felügyelt tanulással(Supervised Learning - SL)[25] célszerű kezdeni, hiszen ipari részesedését tekintve igen jelentős irányzatról van szó. Ahogy a 3.1 ábrán látható, két feladattípus tartozik ennek az ágazatnak az eszköztárába, a klasszifikáció és a regresszió. A klasszifikáció lényege, hogy az adatok előkészítve címké-



3.1. ábra. A gépi tanulás csoportjai [26]

zetten érkeznek és a modell feladata egy bejövő adatsor hatására megállapítani milyen osztályba tartozik[25], amivel a későbbiekben címkézetlen adatok klasszifikálására is képes. Ennek alkalmazása a járműiparban főként a gépi látás területén terjedt el, ezzel lehetővé téve objektumdetekciót és szenzoradatok alapján például forgalomfigyelő rendszerek és biztonsági funkciók használatát. A regresszió szintén címkézett adatokkal dolgozik, csupán annyiban eltérő viselkedést mutatva, hogy nem diszkrét kimeneti halmazsal rendelkezve[27] itt az egyes címkék nem osztályok hanem értékek, aminek segítségével valamilyen görbe illesztése a megvalósítandó feladat. A gyakran alkalmazott SL algoritmusok közül a teljesség igénye nélkül néhány a support vector machine (SVM), a döntési fák, neurális hálózatok[28], bár ezek közül van átjárás a másik ágakba is, hiszen a dolgozat témáját képző irány is használ neurális hálózatokat, de akár fakeső algoritmusok is fellelhetők Megerősítéses Tanulás esetére is bizonyos alkalmazásokban, mint például a Monte Carlo Tree Search[29].

A felügyelet nélküli tanulás (Unsupervised Learning - USL) célja egészen más, olyan esetekben használható, ha címkézetlen adatok érkeznek és ezek között kell valamilyen kapcsolatot teremteni. A bemeneteken hasonlóan ugyanúgy megjelenhetnek képek pixelei vagy hanghullámok[30], viszont abból fakadóan hogy nincs megadva számára kimenet, olyan feladatok ellátására alkalmazható mint dimenziószűkítés vagy klaszterezés [30], ezeken belül mintafelismerés, társadalmi hálózatelemzés vagy ajánló rendszerek[31]. Klaszterezésre bevett eljárás a K-Means Clustering[32], ami a megadott klaszterszám szerint rendezi az egyes pontokat, de az USL ágába tartozik a második világháború előtt kifejlesztett[33] Principal Component Analysis is[32], ami egy orto-

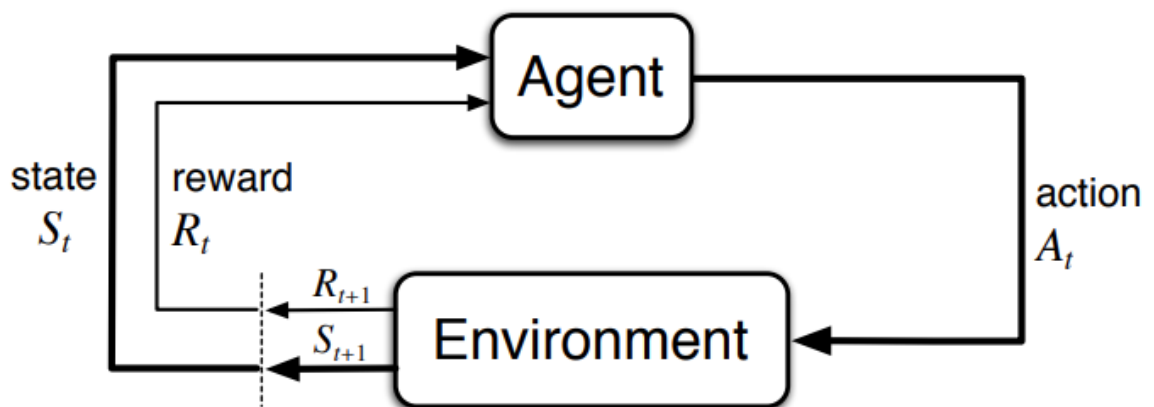
gonális transzformáció segítségével konvertálja az esetlegesen összefüggő adatokat egy lineárisan nem összefüggő adathalmazzá ezeket fő komponensnek nevezve, ezzel a dimenziószűkítéssel a későbbi számításokat megkönnyítve és felgyorsítva[32].

A Megerősítéses Tanulás (Reinforcement Learning - RL) egészen jól elkülöníthető az előző két ágazattól tanítóminták és megvalósítandó feladat tekintetében is, és mivel ez képezi a dolgozat témáját, külön fejezetben tárgyalva bővebben kerül leírásra.

3.2. Megerősítéses tanulás

Az ML legkisebb ipari részesedéssel bíró csoportja az RL, ami trial-and-error módon törekszik egy ideális döntési forma elsajátítására [34]. Alapvető működési elve két objektum kommunikációja, ami egy környezet és egy ágens, ahol az ágens minden egyes iteráció során meglép egy beavatkozást, amire válaszul egy új állapotot valamint a döntés minőségét számszerűen reprezentáló rewardot kapja vissza, amint az a 3.2 ábrán látható. Az ágens az optimális döntésekre való törekvését a visszacsatolt reward maximalizálásával teszi, a korábbi tapasztalataira hagyatkozva. Az RL matematikai keretrendszerét a Markov-döntési folyamat adja, aminek alapvető elemei [35]:

- S : az állapotok halmaza
- A : az ágens által megléphető beavatkozások halmaza
- R : az ágens beavatkozásai által halmozott jutalom
- P : az állapotátmenet valószínűség mátrix
- γ : a discount factor,



3.2. ábra. Megerősítéses tanulás sematikus ábrája[36]

melyek közül az állapot, a reward és beavatkozás a tárgyalt objektumok kommunikációjából meghatározásra került (3.2 ábra), a P az állapotátmenet valószínűségét adja meg, hogy s_t állapotból a_t beavatkozásra s_{t+1} állapotba kerül az ágens, míg a discount factor azért felelős, hogy a kapott rewardokat egyfajta előretekintéssel meghozva létrejövő láncolatban az időben hamarabb bekövetkezők nagyobb súllyal számítsanak (3.4 ábra), a későbbiekben bővebben tárgyalva.

Menetét tekintve az RL elkülönül az ML másik két ismertetett csoportjától, hiszen itt nincs szükség a tanítóminták gondos előkészítésére, mivel az algoritmus tanulás közben generálja saját magának a tapasztalatokat, ami mentén a későbbiekben ideális döntések születnek. Fontos aspektus, hogy a minták nem kerülnek azonnal felhasználásra, hanem egy bufferben lesznek eltárolva egy tuple formájában, ami az alábbi módon implementálható:

$$Experience = (state_i, action_i, reward_i, state_{i+1}), \quad (3.1)$$

ezzel az összes kommunikációs jelet a két objektum között felhasználva, aminek segítségével a későbbiekben ismertetett módon megvalósul az optimalizáció. Ezekből az adatokból aztán véletlenszerűen mintavételezve meghatározódik egy csomag, ami függvényapproximátort fogja majd információval ellátni.

A tapasztalatainak eltárolását követően szükséges ismertetni, miként valósul meg a minták felhasználása az optimalizáció során. Tapasztalat alapú döntéshozásról a folyamat elején nem lehet szó, hiszen nincsenek még tanítóminták, ennek okán véletlenszerűen hozza meg az ágens a döntéseket. Amint az idő előrehaladtával kezdi kiismerni a környezetet, a rendszerben megforduló tanítómintákkal szerzett gyakorlata alapján tud jobb döntéseket hozni. Így praktikusán az RL ágensek kezdeti fázisban a környezet felfedezésére fordítják lépéseiket, ezzel a későbbi optimális viselkedésre felkészítve önmagukat. Ennek mentén a kétfajta meghozható döntés a felfedező random beavatkozás, valamint a rendszert kizsákmányoló maximális rewardot jelentő döntés. A növekvő tudással a hálózat hangolásához egyre gyakrabban meghozott ideális döntés implementációja az ϵ -greedy policy segítségével valósul meg, ahol az exponenciálisan csökkenő ϵ -t összehasonlítva egy véletlenszerűen generált számmal dől el, milyen jellegű beavatkozás kerül meghozatalra. Amennyiben a generált szám kisebb, felfedező lépést tesz az ágens, ha nagyobb, akkor egy ideális döntés lesz meghozva. Az idő előrehaladtával csökkenő jelleg ekvivalens a magasabb számú jó döntéssel, ezzel olyan tanítómintákat forgatva a bufferben, amik a sikeres lépésekhez vezetnek.

A tanítás végén, amikor már elegendő minta megfordult a rendszerben, a hálózatnak elég információja van az egyes állapotok beavatkozásainak minőségéről, így a modell illesztése megvalósulhat nagyobb számú optimális döntéssel is, melyek a sikeres tanítás

esetén már nem csak a lokális, hanem a globális optimumot is jelentik. Természetesen ez azt is magával vonja, hogy az ϵ értéke alapvetően fontos a tanítás sikerességét tekintve, hiszen ha lassan csökken nem lesz elég információ a szándékos jó döntésekről, ha meggyorsabb a megfelelő mértéknél, akkor a lerövidített felfedező szakasz következtében nem ismeri eléggé a környezetét, így könnyen lokális szélsőértékbe futhat, ezzel egy szuboptimális viselkedést megvalósítva.

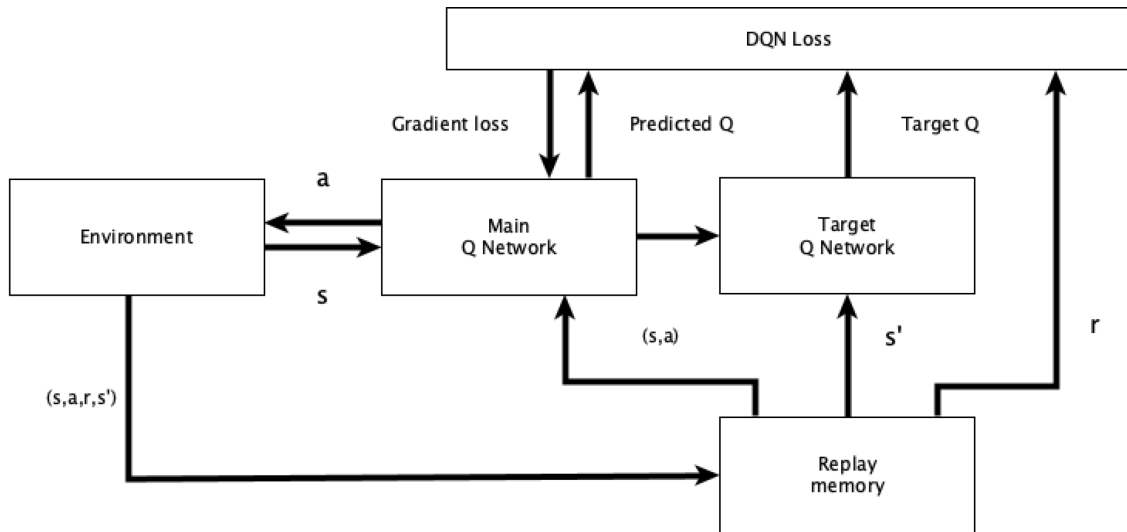
A tanítóminták alapján a döntések minőségének elkülönítésére több megoldás is található. RL-en belül is találkozhatunk érték és policy alapú algoritmusokkal, melyekből az első csoportba tartozó Q-learning egy olyan megközelítés, ahol az egyes állapot és beavatkozás párokhoz rendelve Q-értékek alakulnak ki, ezzel meghatározva az állapotátmenet minőségét egy skalárszámmal jellemezve. Akár táblázat, akár mély tanulás alapú megközelítés kerül alkalmazásra, a Q-értékek a Bellman-egyenlet segítségével számíthatóak (3.2 egyenlet), ahol a két objektum egymással közölt adatai közül a state-action mentén kialakított állapotátmenetekhez tartozó reward direkt módon befolyásolja a kialakuló skalárszámot.

$$Q(s, a) = Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s, a') - Q(s, a)) \quad (3.2)$$

A későbbiekben ez fontos lesz a neurális hálózat hangolása esetén, hiszen az alábbi egyenlet magába foglalja a legnagyobb jövőbeni Q-értékkel rendelkező esetet is, aminek az elérése az ágens által megfogalmazott cél. A formalizmusban szereplő α és γ két skalárszámot jelölő konstans, a learning rate és a discount factor, amik súlyozó konstans szerepet töltenek be az egyenletben. Ezek közül a korábban már tárgyalt γ felelős azért, hogy mivel az ágens egyfajta előretekintő jelleggel hozza meg a döntéseit, az aktuális állapotot jobban befolyásoló rewardok értéke nagyobb súlyozással számítson az összképet tekintve. Ennek kifejezése a következőképpen tehető meg:

$$R_t = \sum_{t'=t}^T \gamma^{t'-t} r_{t'} \quad (3.3)$$

ahol az 1-nél kisebb értékű γ egyre magasabb hatványértékeken egyre kisebb súlyt ad az állapottól aktuális távolságban elhelyezkedő rewardnak, ezzel a jelenlegi helyzethez közelebb állók magasabb értékét prioritizálva a sokkal később következőkkel szemben. Az α értéke a tanulás sebességét nagy mértékben befolyásoló paraméter, hiszen ez súlyozza az állapothoz tartozó Q-értéket azzal szemben, hogy az adott reward és elérhető maximális Q-értéket jelentő lépésből milyen mértékben tanul az algoritmus, ezért is kapta a learning rate nevet.



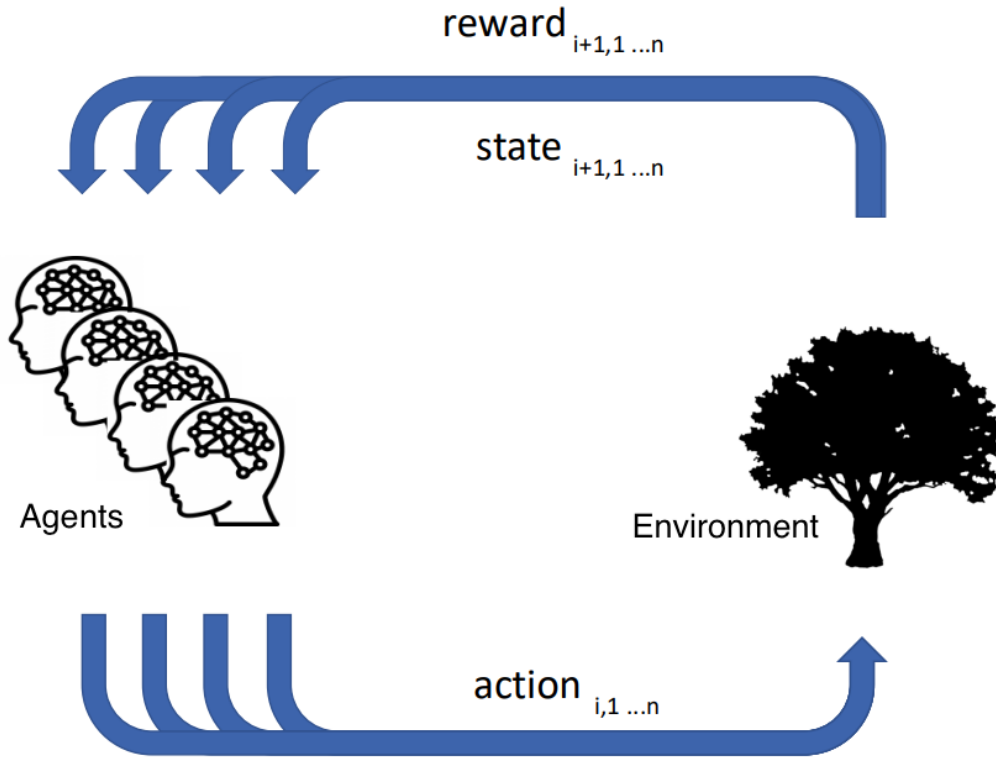
3.3. ábra. Tanítóminta kiválasztás és neurális hálózat hangolása[37]

Az RL alapvető koncepcióját kitárgyalva valamint a forgalmi hálózatra visszatérve látható, hogy ez az eszköztár olyan hálózaton ahol több ponton is szeretnénk beavatkozni nem kielégítő vagy óriási neurális hálózatot eredményezne, így olyan irányba kell további megoldásokat keresni, ahol nem csak egy ágens kap helyet, hanem a hálózat hozzáférhetőségét megnövelve önálló entitásokra bontva valósul meg az irányítás.

3.3. Multi-ágens megerősítéssel tanulás

Az RL általános leírásánál megállapításra került, hogy az algoritmus célja, hogy egy ágens adott környezetben elsajátítson egy ideális viselkedési formát. Adódnak azonban olyan rendszerek, ahol az összes beavatkozás nem reprezentálható egyetlen ágenssel, a több beavatkozási pont és elszeparált hálózati csomópontok miatt. Erre az esetre célszerű Multi-Ágens rendszerekben gondolkodni, ahol az ágensek önálló entitásokként képesek egy megosztott környezetben döntéseket hozni.

Mivel az RL alapvető működésével teljesen megegyező jellegű, annak ellenére hogy több beavatkozó osztozik a tárgyalt környezeten, a kommunikált jelek is megegyeznek, csupán annyi eltéréssel, hogy minden ágens fog döntést hozni[38], és az állapotreprezentáció és reward kialakításának függvényében információt visszkapni (3.4 ábra). Ez célszerűen egy hozzáadott ciklus segítségével valósul meg, ami időlépésenként minden ágenszt megszólítva beavatkozási lehetőséget ad. A döntéshozás több módon is megvalósulhat, annak függvényében, hogy mi az egyes ágensek célja.



3.4. ábra. MARL

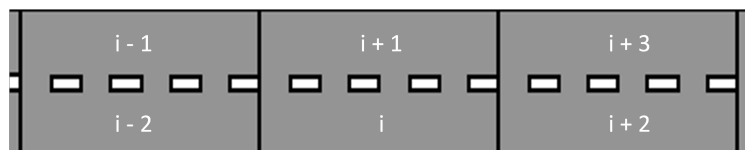
Amennyiben az ágensok egy közös célt szolgálnak, a implementációt célszerű úgy elvégezni, hogy a döntéshozás ne az egyénszintű optimumot tűzze ki megvalósítandó kötelességének, hanem kooperatívan egy magasabb szinten keressen ideális beavatkozásokot. Gyakran alkalmazott megközelítés forgalomirányítás esetére, hiszen a járművek együttes mozgása olyan kooperatív feladat, ami esetenként egyéni optimum feláldozásával jár a járműoszlop dinamikájának javára. Emellett léteznek kompetitív ágensok, melyek célja a többi beavatkozó kárára a saját helyzetből kihozható legnagyobb reward realizálása. Végül szükséges megemlíteni a vegyes működésű ágenseket, melyek egyik csoportba sem sorolhatóak egyértelműen[39], vagy akár az idő előrehaladtával változtatják jellegüket.

A MARL feladatok során egyértelmű előnyként mutatkozik, hogy egyes ágensok tudnak tanulni a többiek tapasztalataiból, ezt megoszthatják egymással ha megosztó ágensekként vannak létrehozva[40], hiszen azonos környezetben végzik feladatukat, de igény esetén tudásuk rejtve is maradhat a többiek előtt, ugyanis kompetitív viselkedés esetén nem célszerű az ellenféllel megosztani tapasztalataikat. Architektúrájukat tekintve lehetnek homogén és heterogén ágensok, ami a belső felépítésre vonatkozik[40], hasonlóság esetén akár egy neurális hálózattal reprezentálhatóak.

Az alábbi besorolásokból adódna, hogy triviális a működés módjának megfogalmazása és az ágensek jellegének meghatározása, ami az állapotrepresentáció és a reward kialakítása során megfontolandó részfeladat, viszont kitérve a szerzett tapasztalatok eloszlására nem ennyire egyszerű a helyzet, ugyanis egy kompetitív játék felfedező szakaszában is előállhat az eset, hogy a véletlen kezdeti beavatkozásokból összeálló inger-szegény környezetben szerzett tapasztalatok redundanciája miatt egyes ágensek nem olyan mértékben tanulnak mint a többiek, így mérlegelendő a későbbiek elősegítése végett ideiglenes kooperáció.

A viselkedési formák elkülönítése mellett az alapvető objektumok közti kapcsolatok leírásának tárgyalása is szükséges, hiszen az ezek közti korrelációk meghatározását hivatott elvégezni a neurális hálózat. Állapotrepresentációnak célszerű nem teljes mértékben kihagyni a többi ágens állapotát, hanem őket is a környezet részének tekinteni, hiszen az ágensek önálló entitások, így a többiek mozgása folyamatosan változtatja a saját környezetüket. A reward meghatározása nem csak abban jelent nehézséget, hogy a környezet melyik jellemzőit vegye figyelembe az ágens, hanem ezeknek térbeli elhelyezkedése is számottevő, ugyanis a kooperatív feladatok esetére érdemes az egész hálózatot egy közös reward függvénnyel jutalmazni. Ennek neve identical payoff[39], ami az adott feladat kooperatív jellegét tekintve megfelelő döntésnek mutatkozik, így ez kerül felhasználásra az érintett problémában. Emellett létezik megosztott értékű reward, ahol az egyes ágenseknek adott reward összege nulla, így csupán eloszlásuk kerül meghatározásra.

A teljesség igénye nélkül tárgyalt Multi-ágens rendszer két neurális hálózat segítségével fogja meghozni döntéseit, melynek ágensei a 2.1 részben tárgyalt út szakaszai, melynek indexelése a 3.5 ábrán láthatóak. Ezt bevezetve a teljes úthálózatot lefedve mindenhol képes a rendszer beavatkozni, ezzel megvalósítva a kívánt jobb hozzáférhetőséget.



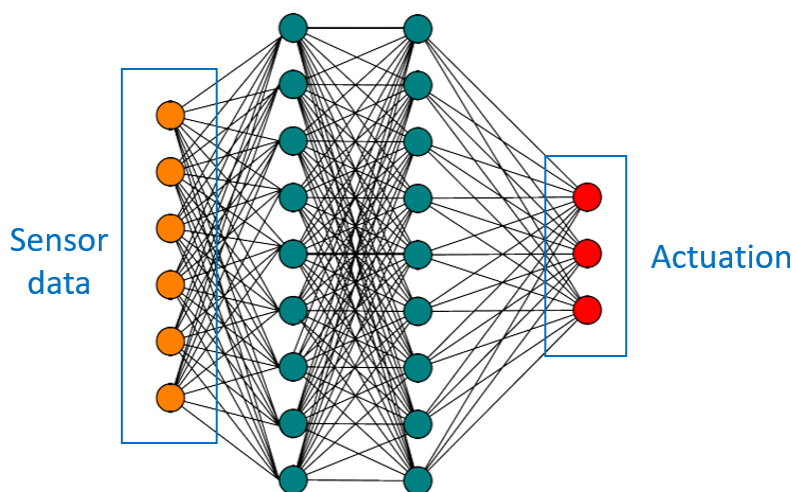
3.5. ábra. Ágensek indexelése

3.4. Mély tanulás

Az AI Winter az RL-t is hasonlóan érintette mint a tárgyalt egyéb ágazatokat. A kifejlesztett algoritmusok számításigényességét nem kiszolgálni képes számítási kapacitás problémája ellehetetlenítette az algoritmusok használatát, irreálisan sok ideig tartott egy-egy modell megalkotása. A technika és a hardveres fejlődés viszont egészen új irányokat nyitott meg az ML-ben, hiszen számítógépek és algoritmusok fejlesztése olyan mértékben szinkronba került a korszerű GPU-k és TPU-knak köszönhetően, hogy ismét releváns kutatási területté lépett elő, így teret adva a Mély Tanulásnak (Deep Learning - DL) is, akár bérelhető szervereken futtatva a tanításokat.

Az ML egy részhalmaza a DL, amely korszerű megoldást biztosít a probléma méretétől függően neurális hálózat (Neural Network - NN) illesztésével, hogy fizikai egyenletek helyett létrejöjjön egy kapcsolat szenzoradatok és kimenetek között. Alkalmazásának fő motivációja, hogy összehasonlítva a táblázat és fakereső algoritmusokkal, itt meghatározott számú matematikai művelet végrehajtása valósul meg minden iteráció során, így egyrészt determinisztikus számítási időket megvalósítva, valamint a feladat mértékével és a megszerzett tapasztalatok számával szinkronban nem növekszik exponenciálisan a számítás igénye.

Az NN lényegében tenzorműveleteket magába foglaló módszer, ahol az egyes rétegek neuronjai között elhelyezkedő súlyok és aktivációs függvények teremtenek kapcsolatot. Ezek közül a feladat a bemeneten elhelyezkedő szenzoradatok valamint a kimeneten reprezentált beavatkozások között olyan összefüggést találni, ami az RL által kitűzött reward maximalizáló jelleget képes megvalósítani. Olyan NN esetén, amik legalább kettő takart réteggel rendelkeznek beszélhetünk DL-ről.



3.6. ábra. Neurális hálózat kapcsolatteremtése szenzoradatok és beavatkozások között

A tevékenység természetétől függően sokféle NN architektúra fellelhető, ezek a Python nyelv használata esetén meghívhatóak a PyTorch és TensorFlow könyvtárakból. Legegyszerűbb fajtája a Multilayer Perceptron, ami súlyok és neuronok lineáris kapcsolatának összegéből valamint az adott csomópont bias-éből áll össze, az alábbi módon:

$$y = f(b + \sum_{i=1}^n x_i w_i) \quad (3.4)$$

áthaladva egy aktivációs függvényen, ami a probléma jellegétől függően lehet Sigmoid, ReLu, Leaky ReLu vagy SoftMax [41], csak hogy néhány említésre kerüljön a teljesség igénye nélkül. Ezek lényegében azért felelnek, hogy egy neuronnak meghatározzák hogy adott állapotban tüzel vagy sem. Az implementáció során ilyen típusú háló lesz alkalmazva, viszont a kiválasztás során megfontolásra került más kialakítás is, mint a gépi látás terén gyakran használt architektúra a konvolúciós NN, ami lényegében karakterisztika vesztes nélküli alacsonyabb dimenzióra konvertálást jelent. Itt például képek terén a képcsatornáknak megfeleltethetően 3 darab 2D vektor a hálózat bemenete, ahol az egyes bemeneti neuronok pixel intenzitást jelentenek[42], amiket konvolúciós és pooling rétegek követnek. Ebből a konvolúciós rétegek célja a lokális együttállások detektálása, míg pooling lényege a szemantikailag hasonló jellemzők egybeolvasztása [42]. További hálózati architektúrák a konvolúciós hálózatok mellett az RNN (Recurrent Neural Network) és az ebből származó LSTM (Long short-term memory)[43][44], de ezek alkalmazása máshol, az idősorok esetén célszerű.

A hálózatok felépítését követően még lényegi aspektus a bemenő érték és az kívánt előállítandó kimenet közti összefüggés, hiszen a modell célja, hogy előállítsa azt a kimenetet, ami számunkra a legkedvezőbb, klasszifikáció esetén a megfelelő címkét, míg egy RL ágens esetén a megfelelő beavatkozást. Erre a veszteségfüggvények alkalmasak, hiszen ezek a különbség mértékétől függően adnak meg egy értéket, aminek minimalizálására törekszik a hálózat. Jelen esetben a Mean Squared Error lett alkalmazva, mivel nem csak az eltérések irányára érzéketlen, hanem a Mean Absolute Errorral ellentétben a nagyobb eltéréseket nagyobb mértékkel bünteti. Egyenlet formájában a következő módon néz ki:

$$L_{MSE} = \frac{1}{n} \sum_{i=1}^n (Q_{pred} - Q(s, a))^2 \quad (3.5)$$

A rövid kitérőt követően visszatérve az aktuális probléma megoldásához a leggyakrabban alkalmazott könyvtárak közül a PyTorch került használatra, könnyű személyre szabhatósága miatt. A mély tanulás felvezetését követően a következő feladat a neurális hálózat integrációja az RL-be. Erre a Deep Q Network (DQN) algoritmus kerül be-

Neural Network					
Layers	10	128	128	128	3
Activation Function	ReLu				
Loss	Mean Squared Error				
Learning rate	0.001				
Discount factor	0.97				
Batch size	512				
Experience memory size	100000				
Freq. of weight sharing	40				
Optimizer	Adam				

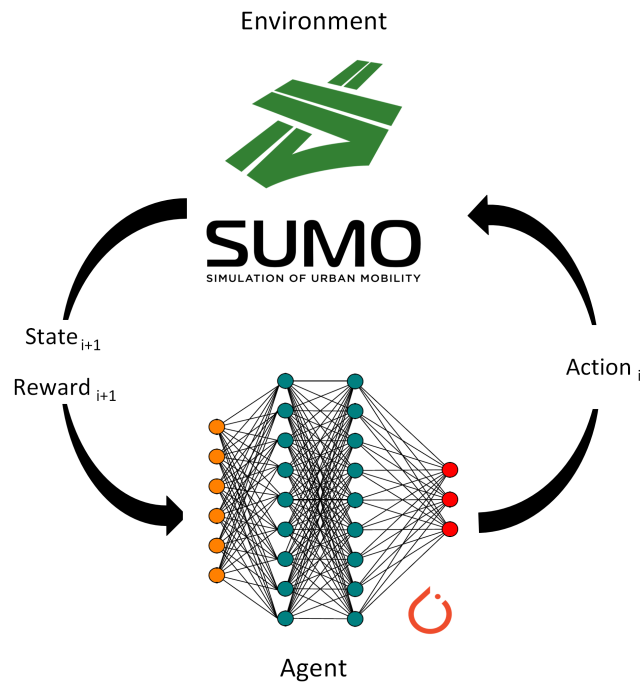
3.1. táblázat. Neurális hálózat paraméterei

vezetésre [45], aminek megjelenése egészen új perspektívát nyitott az RL-t tekintve. Lényege, hogy két létrehozott neurális hálózat segítségével történik meg a hangolás, amikből az egyik minden lépés során illesztve folyamatosan változik, míg a másik bizonyos időközönként lemásolja. A két hálózat azért indokolt, hogy a folyamatosan változó súlyok miatt kis eltérésre is változhat az ideálisnak vélt kimenet, így a hangolás során érdemes az elérhető maximális Q-értékeket egy másodlagos Target hálózatból kinyerni, ami a végén megegyezik a Model hálózattal, csupán alacsonyabb frissítési rátával.

3.5. DRL problémaspecifikus implementációja

Az eddigiekben tárgyalt témákat felhasználva egy olyan feladat megvalósítása szükséges, ami ötvözi a kooperatív MARL és a DRL előnyeit, ezzel elvégezhető mennyiségű számításra támaszkodva az emissziós ráták egész hálózaton értelmezett javításával. A tanító ciklus két fő objektuma a 3.7 ábrán látható módon alakul ki, ahol a SUMO és Gym implementáció adják a környezetet, míg bár Multi-ágens problémáról van szó, az ágensek homogén természeténél fogva egy kimenő neurális hálózat tanításával áll elő a modell, ami bármelyik útszakaszra felhasználható hálózatként fog működni.

A MARL implementációja egy további belső ciklus, ami minden időlépésben végiglépteti az összes ágensen a hálózatot, ezzel megadva a beavatkozás lehetőségét minden útszakaszban, de ezt csak 10 másodpercenként engedélyezve, ahogyan az a fizikai környezet kényszerei között meg volt fogalmazva mint szempont. A másik alkalmazott kényszer a beavatkozásoknál lesz tárgyalva, hiszen nem magához a környezethez, hanem az ágensekhez tartozó megkötés, ami a beavatkozási tartományt illeti. Az algoritmus inicializálásánál véletlenszerűen megválasztott súlyok segítségével jön létre a két hálózat, melyek majd a tanítás során folyamatosan változnak. Az egyes kommunikált értékek kifejtéséből kirajzolódik, milyen fizikai paraméterek terén elvárt fejlődés a hálózat által kitűzött cél.



3.7. ábra. Megerősítéses tanulás problémaspecifikus implementációja

2. Algorithm MARL DQN[45]

```
Initialize replay memory  $D$  to capacity  $N$ 
Initialize action-value function  $Q$  with random weights  $\theta$ 
Initialize target action-value function  $\hat{Q}$  with weights  $\theta^- = \theta$ 
for episode = 1,  $M$  do
     $state_t \leftarrow \text{reset}()$ 
     $done \leftarrow \text{FALSE}$ 
     $reward_t \leftarrow 0$ 
    for time step = 1,  $T$  do
        for agents = 1,  $A$  do
            With probability  $\epsilon$  select a random action
            otherwise select  $a_t = \text{argmax}_a Q(s, t)$ 
             $state_{t+1}, reward_t, done \leftarrow \text{step}(a_t)$ 
            Store transition  $(s_t, a_t, s_{t+1}, r_t, d_t)$  in  $D$ 
            Sample random minibatch of transitions  $(s_t, a_t, s_{t+1}, r_t, d_t)$  from  $D$ 
             $s_t = s_{t+1}$ 
            if  $d' = TRUE$  then
                 $y_j = r'_j$ 
            else
                 $y_j = r'_j + \gamma \cdot \max_a \hat{Q}(s'_{t+1}, \hat{\theta})$ 
            end if
            Perform a gradient descent step on  $(y_j - Q(s_t, a_j, \theta))^2$  with respect to the
            network parameters  $\theta$ 
            Every  $C$  steps reset  $\hat{Q} = Q$ 
        end for
    end for
end for
```

3.5.1. State

Amint az a DL részben tárgyalva volt, az ágensként használt neurális hálózat bemeneti rétegén elhelyezkedő információ az állapotreprezentáció. Itt szenzoradatok kapnak helyet, amik segítségével a környezet állapota írható le. Mivel stratégiai jellegű döntéshozásról van szó, olyan makroszkópikus jellemzők használata indokolt, melyek segítségével a cellahatárok állapota hozzáférhető. A keresztmetszet adottság, így a sűrűségek és sebességek monitorozásán keresztül előáll a forgalom aktuális állapota. A korábbi MARL részben tárgyalt indirekt kooperáció megvalósításához azonban szükség van nem csak a saját jellemzőkre, hanem a környezetiekre is, hiszen az állapot függvényében kialakított differencia a ki és beáramló járművek számából tevődik össze.

Ahogy az a 3.3 részben a 3.5 ábrán látható, a számozás indokolja hogy az aktuális cella elé és mögé tekintése miatt mindkét irányból kettő szomszédos is az állapotreprezentáció részét képezze, ezzel minden feltételt megteremtve a differencia előállításához.

Ennek fényében ezek az értékek a 10 bemeneti neuron megfeleltetései.

Szükséges feltételként az algoritmus gyorsítása végett a szűkület utáni cellák nem avatkoznak be, hiszen ott telítődés nem történik. Amit még ki kell kötni, hogy a hálózat elején sem alkalmas az állapotrepresentáció hogy ágens helyezkedjen el, hiszen előtte nincsenek cellák, de ez nem is probléma, hiszen a szimuláció úgy lett elvégezve, hogy az irányított zóna előtt mindenhol a megengedett maximális sebességgel lépjenek be a járművek, ezzel egyenlő feltételeket teremtve a teszt során.

$$state_i = \begin{bmatrix} velocity_{i-2} \\ density_{i-2} \\ \cdot \\ \cdot \\ \cdot \\ velocity_{i+2} \\ density_{i+2} \end{bmatrix} \quad (3.6)$$

3.5.2. Action

A beavatkozásokat érintő korlátozások a környezet részben definiálva voltak, így ezek mentén inkrementális jelleggel a 3.7 vektorban látható értékeket tudja meglépni az aktív ágens. Természetesen itt minden esetben figyelembe veszi a maximális 130 km/h végsebességet valamint egy hozzáadott kényszerként azt is, hogy 30 km/h sebességhatárnál egyik szegmensre sem szab kisebb korlátozást.

Hatalmas előny ez a jellegű döntéshozás, hiszen a teljes tartományban tud lefedni úgy beavatkozásokat, hogy csupán 3 kimeneti neuronnal rendelkezik a neurális hálózat, valamint az inkrementális jellegben rejlő korábban említett előny, hogy nem okoz hirtelen változásokat is emellett szól.

$$action = \begin{bmatrix} +10 & km/h \\ 0 \\ -10 & km/h \end{bmatrix} \quad (3.7)$$

3.5.3. Reward

Amint az az RL tárgyalásánál definiálva volt, az algoritmus célja olyan viselkedési forma elsajátítása, ahol ő a legnagyobb rewardot realizálja. Mivel a Bellman egyenletben látható módon a reward az egyetlen, ami a Q-értékek kialakításában számszerűen részt vesz, alapvető feladat ennek az ideális megfogalmazása. Célszerű olyan fizikai jellemzők használata a feladat során, amik a lehető legtöbb információval rendelkeznek a jó és

rossz állapotok elkülönítéséről.

Bár környezetvédelmi aspektusa is fontos vonulata az algoritmusnak, mégis az áramlás folytonosságának irányából megközelítve kerül megfogalmazásra a reward függvény. Annak okán, hogy az állások meddő állapotként interpretálhatóak, ezek büntetése csökkentheti ennek a jelenségnek az előfordulását. Minél nagyobb a realizált átlagsebesség, annál több jármű képes áthaladni a hálózaton, így ennek növekedését jutalmazni szükséges, hogy az algoritmusba egy olyan visszacsatolásként érkezzen, hogy ehhez mértén tudja illeszteni a hálózatot. A metrikákból adódóan a hálózaton értelmezett késleltetések (w) valamint az átlagsebességek (v_{avg}) segítségével a reward jutalomból és büntetésből egyszerre áll, melyek egymáshoz képesti súlyait jól meghatározva sikeres tanítás víziója áll az algoritmus előtt.

Két fajta reward koncepció került alkalmazásra, melyek közül az első mutatott érdekes viselkedése súlyozási problémákat vetett fel, ennek a viselkedésnek a következményeként került sor a második reward implementálására. A külső sávot feláldozva engedett minden járművet maximális sebességgel a belső sávban, hiszen a várakozások mértékével nem vesztett eleget az ágens, hogy indokolt legyen a belső sávokban beavatkozni.

$$R = \frac{v_{avg}}{1 + w}, \quad (3.8)$$

Ennek okán a nevezőt négyzetre emelve már olyan súly került a rendszerbe, ami nagyobb mértékben bünteti az előforduló várakozásokat, és vizuális ellenőrzés után jól láthatóan csillapítja a forgalmat a kritikus keresztmetszet környezetében, viszont ennek mértéke már a következő részben lesz ismertetve.

$$R = \frac{v_{avg}}{(1 + w)^2}, \quad (3.9)$$

A megvalósult tanulást követve kiértékelés szükséges a rewardok teljesítményének meghatározására, amiből kiderül, sikeres-e a kapcsolatteremtés a neurális hálózat és a forgalmi rendszer között.

4. fejezet

Eredmények

4.1. Adatok a kiértékeléshez

A városok és agglomerációjuk élhetőségének fejlesztése érdekében megfogalmazott javítandó mutatók mentén kerül sor a kiértékelésre, kitérve minden motivációként megfogalmazott feladatra. A kötelességként kitűzött tranziensek elkerüléséhez felhasznált adatok mind a SUMO-ból érkeznek, ezek közül van mért, valamint van empirikus modell alapján számított mutató is. A hálózaton értelmezett várakozási idők mért adatok, az egyes járművekhez és sávokhoz van hozzárendelve, viszont a kibocsájtások már más megközelítést igényelnek.

Ahogy az majd a számszerű kiértékelés során is látható lesz, az emissziók trendje hasonló mértékű javulást eredményez, aminek magyarázata az kibocsájtást számító modellben rejlik. Erre a SUMO több alternatívát is kínál, melyek közül a HBEFA v2.1 [46] alapértelmezett került alkalmazásra. Ez az alábbi módon számítja az egyes kibocsájtásokat:

$$E = c_0 + c_1va + c_2va^2 + c_3v + c_4v^2 + c_5v^3, \quad (4.1)$$

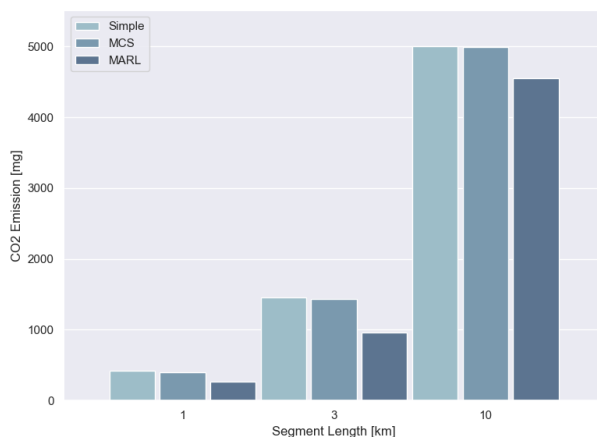
a kívánt emissziók mértékét $\frac{mg}{s}$ -ban megadva. A kontinuitás szempontjából lényeges megjeleníteni a mozgó forgalmi hullámokban részt vevő járművek számát, ezt több mutató segítségével meghatározva, ugyanis ezeknek a meddő állapotoknak az elkerülése fő cél. Első nem szándékolt viselkedési forma a megállásra kényszerített járművekben jelenik meg, aminek időbeli alakulása érdekes, ezzel jól szemléltetve hol nem tud megbirkózni az irányítási forma a forgalom nagyságával. Ennek időtartama is számottevő mutató a forgalom leírására, hiszen ha már bekövetkezett a probléma indokolt a mihamarabbi feloldás. Utolsó mutatóként pedig az átlagos sebességek kerülnek használatra, hiszen ez jól reprezentálja adott algoritmusok esetén az út terhelhetőségét, legalábbis

amennyiben a hálózat mérete alkalmas megfelelő finomhangolásra, és a rajta létrejövő hullám nem elenyésző mértékű az egész hálózatot tekintve. Itt szükséges kikötni, hogy elvártan ez a mutató a kisebb hálózatokon fog javulni, hiszen a telített VSL zónák esetén is a feláramló maximális sebességgel érkező járművek nagy hosszokon javíthatják az átlagot és a jobb haladás illúzióját kelthetik.

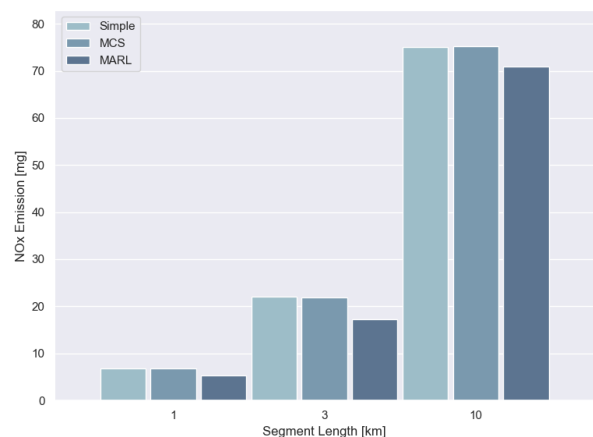
4.2. Összehasonlító analízis

Amint az a motiváció megfogalmazása mentén kirajzolódott, a fő problémát okozó tranziensek elkerülését kitűzve célul, kiértékelés során az elvárt viselkedés az ingadozások csillapítása és a magasabb realizált átlagsebesség, valamint a csökkenő mennyiségű álló kibocsájtás. Az összehasonlítás elvégzése három esetre lett vizsgálva, melyek közül a dolgozat témáját képző algoritmus bemutatása magától értetődő, viszont létjogosultságának igazolása okán összehasonlítandó az algoritmusmentes környezetekkel, melyek az útrendszerek nagy részét képezik, valamint egy alkalmazásban lévő algoritmussal, ami jelen esetben az MCS. Minden eshetőség három különböző hosszon volt vizsgálva, abból az okból kifolyólag, hogy az algoritmus alkalmazkodóképessége összehasonlítható legyen az MCS-sel, valamint az esetleges valós alkalmazás esetén az ideális szektorhosszok meghatározásra kerüljenek. Az alkalmazkodóképesség demonstrálásának érdekében azonos modell került használatra minden hálózaton.

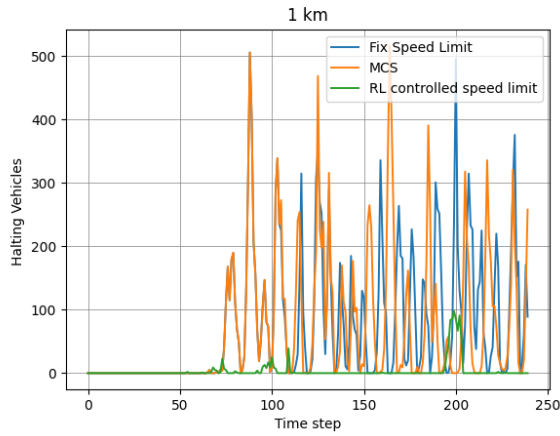
A teszt során első megfigyelt viselkedés, hogy bár az MCS minden hálózaton alkalmas volt csekély mértékű javulást felmutatni az irányítás nélküli realizációval szemben, a bevezetésben leírt értékek valamint az itt használt VSL zóna pozíciója átgondolást igényelt, bizonyos esetekben hangolás nélkül nem volt képes a beáramló sebesség-



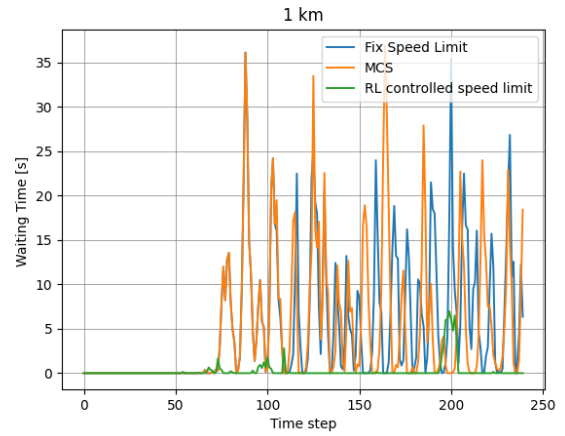
4.1. ábra. CO_2 kibocsájtás összehasonlítása a három vizsgált hosszra



4.2. ábra. NO_x kibocsájtás összehasonlítása a három vizsgált hosszra



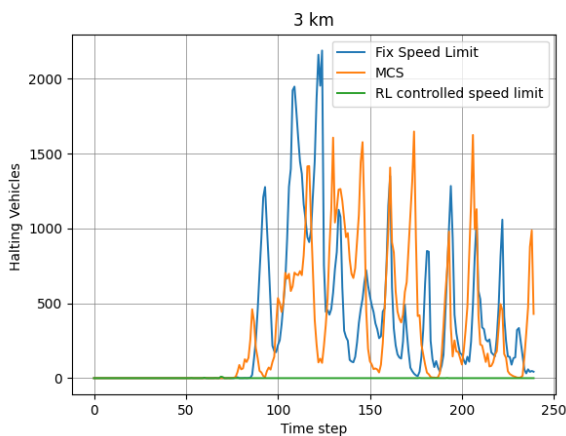
4.3. ábra. Megállásra kényszerített járművek száma 1km hosszú útszakaszon



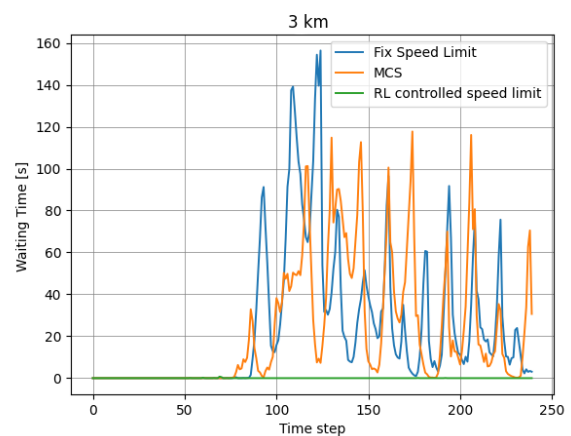
4.4. ábra. Várakozási idő alakulása az időben 1 km hosszú útszakaszon

korlátokat megközelíteni sem a járművek sebessége a kialakult mozgó forgalmi hullámok miatt.

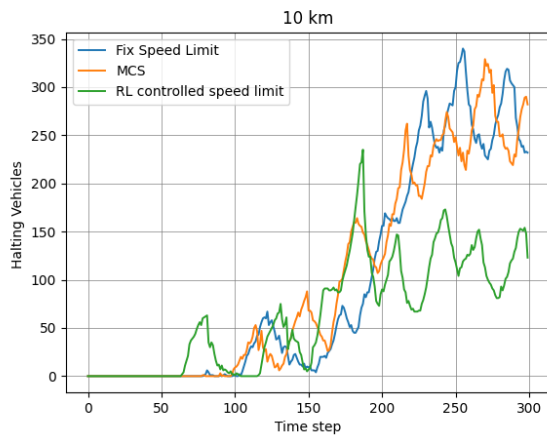
A MARL valódi előnye viszont tisztán kirajzolódik a tesztek során. Itt minden változtatás nélkül egyszerűen betöltve a neurális modellt számottevően nagyobb javulások tapasztalhatóak, mint a másik két módszer esetén. Az sem zavarta meg a hálózatot, hogy teljesen más fizikai kiterjedéseken került vizsgálatra, hiszen a jó állapotreprezentáció és reward következtében a sebesség-sűrűség párokból továbbra is működőképesen tudott kimenetet előállítani a hálózat. A megoldásnak köszönhetően így a részletesebben felbontott hálózat bár megnövelt szenzoradattal dolgozva, de egy működőképes realizációval szolgál.



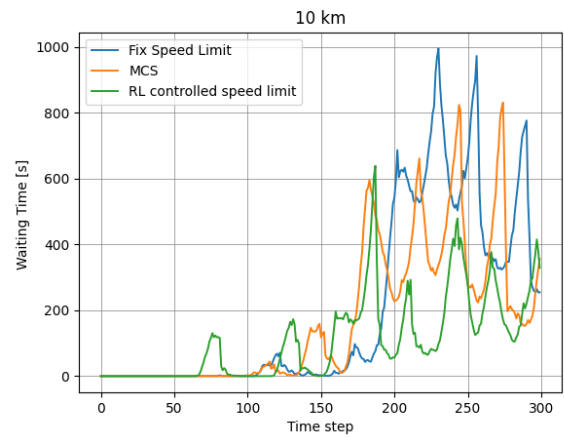
4.5. ábra. Megállásra kényszerített járművek száma 3km hosszú útszakaszon



4.6. ábra. Várakozási idő alakulása az időben 3 km hosszú útszakaszon



4.7. ábra. Megállásra kényszerített járművek száma 10km hosszú útszakaszon



4.8. ábra. Várakozási idő alakulása az időben 10 km hosszú útszakaszon

Az egyértelmű javulás mellett látható, hogy egy olyan dilemmát indít meg a szenzoradatok térbeli sűrűsége, ami a pontosabban leírt rendszer növekvő költségeit állítja szembe azzal, hol növelhető még a hatásfok, és az alkalmazásban az anyagi ráfordításokat ez alapján kell mérlegelni. Egyes hálózatokat megfigyelve megmutatkozik, hogy a legnagyobb esetre a ritkás szenzoradatok miatt kialakulnak a hullámok, viszont a MARL továbbra is nagy mértékű javulásra teszi alkalmassá a vizsgált környezetet. A két kisebb hálózat hasonló viselkedést mutat a saját realizáció tekintetében, viszont az MCS esetére tekintve a diagramokat az alacsonyabb csúcértékek bekövetkezése egyértelműen mutatja, hogy felgyorsul a hullámok kezelése.

A három hosszon ami még mindenképpen tanulságos, tekintve a diagramokat látható, hogy a kisebb hálózatokon sokkal gyakoribb, de gyorsabb lecsengéssel jelennek meg a mozgó forgalmi hullámok, míg a nagyobb esetben ezek egy idő után folyamatosan részét képezik a forgalomnak. Ebből kifolyólag bármilyen alkalmazás kerül használatra, mindenképpen érdemes a sűrűbb szenzoradatokra törekedni egy bizonyos mértékig, hiszen amint a 10 kilométeres autópályaszegmens vizsgálatánál látható, már a MARL sem képes teljes mértékben kiküszöbölni a káros jelenséget. Viszont ami a kutatás részfeladatát képző ágens alkalmazkodóképességet illeti, szépen megmutatkozik, hogy bár eltérő mértékben, de alkalmas metódus ilyen jellegű problémák esetére.

Results				
Length	Metric	Simple	MCS	MARL
1 km	CO2 [mg]	420.38321	402.28047	260.66972
	NOx [mg]	6.87358	6.77200	5.30981
	Mean Speed [m/s]	16.66204	16.86369	17.85730
	Estimated Travel Time [s]	9933.68547	7912.26722	1206.78500
	Waiting Time [s]	5.42857	5.09791	0.25446
3 km	CO2 [mg]	1458.40422	1431.63130	955.66162
	NOx [mg]	21.99125	21.8748	17.25388
	Mean Speed [m/s]	16.78304	16.92804	17.60460
	Estimated Travel Time [s]	13326.52431	10471.01248	41.89551
	Waiting Time [s]	24.54404	22.85565	0.007142
10 km	CO2 [mg]	5003.25073	4987.89459	4556.32901
	NOx [mg]	75.02422	75.24406	70.85701
	Mean Speed [m/s]	22.63837	22.59483	22.49749
	Estimated Travel Time [s]	1971.82494	1230.71240	772.22486
	Waiting Time [s]	205.43928	173.36738	109.90261

4.1. táblázat. Eredmények szemléltetése különböző hosszokon

5. fejezet

Konklúzió

5.1. Kiértékelés

A dolgozat létrehozásának motivációja az volt, hogy az infrastruktúra térbeli kiterjesztésének elkerülése mellett nagyobb forgalmi teljesítmény realizálódjon a hálózaton. Mint ez a számszerű kiértékelést követően láthatóvá vált, az eseményvezérelt jellegnek köszönhetően az egyszerűbb algoritmus, kisebb szenzorigénnyel is képes előállítani a szükséges differenciát, viszont a kívánt előretékelő jelleg egészen más módon, nem csupán felodotta, hanem kisebb hálózatok esetén meg is előzte a mozgó forgalmi hullámok kialakulását.

Természetesen az elért eredmények közútra való kikerülése előtt megfontolandó, hogy a pontos leképezésekre való törekvések ellenére a szimulátorban hozott eredmények közvetlenül reprodukálhatóak vagy sem valós forgalomban, hiszen az emberi tényező, a nem betartott közlekedési szabályok és azok következményei nem modellezhetőek egy ilyen szimulációs környezetben. Ennek ellenére az ilyen jellegű ITS jelentősége az idő előrehaladtával és az egyre magasabban automatizált járművek mellett egyre nagyobb szerepet fog kapni, és a teljesen autonóm módon működő járművek esetén egy jó víziónak ígérkezik, hiszen olyan tartományokat vizsgál, ami az emberi érzékelés határait bőven meghaladja mind feldolgozhatóság, mind beavatkozási sebesség szempontjából, hiszen egy kilométerre az érintett sofőr előtt haladó járművek állapotát képtelenség emberi léptékkal felmérni.

Utolsó fontos aspektusa az alkalmazásnak, hogy számos olyan problémával kell autópályákon szembenézni, aminek irányításmentes környezetben egy meddő állapotokkal terhelt rendszer az eredménye. A túl sűrűn elhelyezett karbantartási és építési munkálatok előzetes tesztelése során az algoritmus teljesítményét kiértékelve adódhat, hogy akár kisebb térbeli mozgatással nagy mértékű javulás érhető el. Ugyanígy érintettek az időszakosan nagy terhelést elszenvedő útszakaszok, mint egy autópálya felhajtó

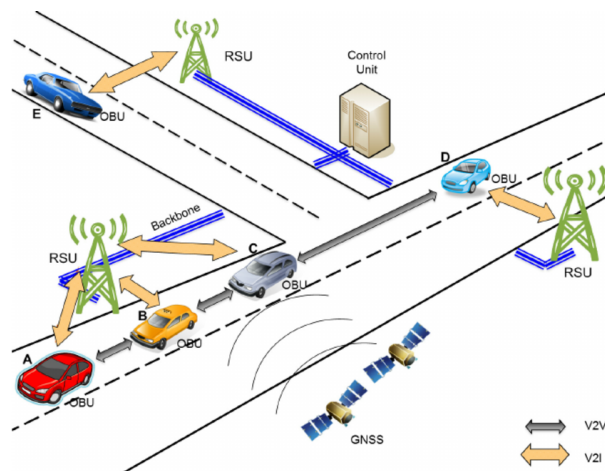
környezete ami a város felé irányul, a munkanap elején.

Következtetésként levonható, hogy az algoritmus teljesítményigényéhez képesti viszonylag alacsony szenzorigény korszerű és releváns megoldássá emeli a MARL alapú VSLC-vel ellátott autópályát, és a szükséges megfontolások mellett alkalmazása valódi előrelépést jelenthet.

5.2. Fejlesztési irányok

A megvalósult eredményeken túlmutatóan még mindig elképzelhető előrelépés a rendszerben, hiszen ahogy a legnagyobb hálózaton látható volt, ott alakultak ki mozgó hullámok a ritkás szenzoradatok okán, bár igaz, még így is nagy javulást elérve. További tesztek elvégzése az ideális hossz megválasztása után azonban még mindig akadnak teendők. Az emberi tényező amíg részét képezi a rendszernek, a reakcióidők továbbra is hátráltató tényezőként fognak megjelenni, viszont a döntéshozás átcsoportosítása gépi megvalósításra az irányítójel és megvalósított közti átmeneti időszakokat csupán a jármű tehetetlenségéből adódóakra csökkentené. Erre természetesen további V2I és V2V kommunikáció létrehozása szükséges.

A V2X-ből (5.1 ábra) származó további információ segítségével, a további szenzoradatok feldolgozása révén lehetőség nyílik több fedélzeti számítógépen olyan számítások elvégzésére, amik emberi léptékben messze nem elvégezhetőek, ezzel mind biztonsági mutatókat, mind pedig a forgalom stabilitását növelve. Kombinálva a teljesen autonóm egyénszintű irányítással az embert csupán felügyelő szerepbe helyezve ezek a számítási kapacitások valóban működésbe lépve nyújthatnak olyan megoldást, ami akár újratervezett trajektóriákkal további emisszió és várakozási idő csökkenést hozhat a rendszerbe.



5.1. ábra. V2X kommunikáció[47]

Ábrák jegyzéke

1.1.	Fundamentális diagram a kritikus kapacitás szemléltetésére[3]	1
1.2.	Az évenként kibocsájtott CO_2 mennyisége[4]	2
1.3.	A kibocsájtás szektoronkénti lebontása [8]	3
1.4.	Variable Speed Limit Control számára felosztott útszakasz [9]	4
1.5.	Kontinuitás tétele a forgalom esetére [6]	4
1.6.	Mozgó forgalmi hullám szemléltetése	5
1.7.	A SPECIALIST fázisai és állapotai [11]	6
2.1.	Vizsgált útszakasz	8
2.2.	A TraCI működése	10
3.1.	A gépi tanulás csoportjai [26]	12
3.2.	Megerősítéses tanulás sematikus ábrája[36]	13
3.3.	Tanítóminta kiválasztás és neurális hálózat hangolása[37]	16
3.4.	MARL	17
3.5.	Ágensek indexelése	18
3.6.	Neurális hálózat kapcsolatteremtése szenzoradarok és beavatkozások között	19
3.7.	Megerősítéses tanulás problémaszpecifikus implementációja	22
4.1.	CO_2 kibocsájtás összehasonlítása a három vizsgált hosszra	27
4.2.	NO_x kibocsájtás összehasonlítása a három vizsgált hosszra	27
4.3.	Megállásra kényszerített járművek száma 1km hosszú útszakaszon	28
4.4.	Várakozási idő alakulása az időben 1 km hosszú útszakaszon	28
4.5.	Megállásra kényszerített járművek száma 3km hosszú útszakaszon	28
4.6.	Várakozási idő alakulása az időben 3 km hosszú útszakaszon	28
4.7.	Megállásra kényszerített járművek száma 10km hosszú útszakaszon	29
4.8.	Várakozási idő alakulása az időben 10 km hosszú útszakaszon	29
5.1.	V2X kommunikáció[47]	32

Táblázatok jegyzéke

3.1. Neurális hálózat paraméterei	21
4.1. Eredmények szemléltetése különböző hosszokon	30

Rövidítések

VSLC	-	Variable Speed Limit Control
ITS	-	intelligens közlekedési rendszert
MCS	-	Motorway Control System
SPECIALIST	-	SPeEd ControllIng ALgorIthm using Shock wave Theory
MPC	-	Model Predictive Control
SUMO	-	Simulation of Urban MObility
AI	-	Mesterséges intelligencia
TraCI	-	Traffic Control Interface
ML	-	Gépi tanulás
SL	-	Felügyelt tanulás
SVM	-	Support Vector Machine
USL	-	Felügyelet nélküli tanulás
RL	-	Megerősítéses tanulás
MARL	-	Multi-ágens Megerősítéses tanulás
DQN	-	Deep Q Network
V2X	-	Vehicle to Everything
V2I	-	Vehicle-to-infrastructure
V2V	-	Vehicle-to-vehicle intelligens közlekedési rendszert

Irodalomjegyzék

- [1] Abriha Dávid. Gépi és mély tanulási eljárások alkalmazása a városi környezet vizsgálatában, nagyfelbontású és különböző dimenziójú távérzékelt adatok alapján.
- [2] Gábor Orosz, Bernd Krauskopf, and R Eddie Wilson. Bifurcations and multiple traffic jams in a car-following model with reaction-time delay. *Physica D: Nonlinear Phenomena*, 211(3-4):277–293, 2005.
- [3] András Hegyi. *Model predictive control for integrating traffic control measures*. Netherlands TRAIL Research School, 2004.
- [4] Hannah Ritchie, Max Roser, and Pablo Rosado. Co and greenhouse gas emissions. *Our World in Data*, 2020. <https://ourworldindata.org/co2-and-greenhouse-gas-emissions>.
- [5] Gabriella Bretti, Roberto Natalini, and Benedetto Piccoli. A fluid-dynamic traffic model on road networks. *Archives of Computational Methods in Engineering*, 14:139–172, 2007.
- [6] Martin Treiber and Arne Kesting. Traffic flow dynamics. *Traffic Flow Dynamics: Data, Models and Simulation*, Springer-Verlag Berlin Heidelberg, pages 983–1000, 2013.
- [7] Péter Mándoki. Személyközlekedési rendszerek értékelési lehetőségei a városi és térségi közlekedésben. 2005.
- [8] Paris IEA. Iea (2020), the role of ccus in low-carbon power systems, iea, paris <https://www.iea.org/reports/the-role-of-ccus-in-low-carbon-power-systems>, license: Cc by 4.0. 2020.
- [9] Dominique Guerillot and Jeremie Bruyelle. Uncertainty assessment in production forecast with an optimal artificial neural network. 03 2017.

- [10] Bálint Kővári, László Szőke, Tamás Bécsi, Szilárd Aradi, and Péter Gáspár. Traffic signal control via reinforcement learning for reducing global vehicle emission. *Sustainability*, 13(20):11254, 2021.
- [11] Ellen F Grumert, Andreas Tapani, and Xiaoliang Ma. Characteristics of variable speed limit systems. *European transport research review*, 10(2):1–12, 2018.
- [12] Bidoura Khondaker and Lina Kattan. Variable speed limit: an overview. *Transportation Letters*, 7(5):264–278, 2015.
- [13] Johann Andersen and Steve Sutcliffe. Intelligent transport systems (its)-an overview. *IFAC Proceedings Volumes*, 33(18):99–106, 2000.
- [14] Andreas Hegyi, Serge P Hoogendoorn, Marco Schreuder, Henk Stoelhorst, and Francesco Viti. Specialist: A dynamic speed limit control algorithm based on shock wave theory. In *2008 11th international ieee conference on intelligent transportation systems*, pages 827–832. IEEE, 2008.
- [15] Andreas Hegyi and Serge P Hoogendoorn. Dynamic speed limit control to resolve shock waves on freeways-field test results of the specialist algorithm. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 519–524. IEEE, 2010.
- [16] Jaume Barceló. Models, traffic models, simulation, and traffic simulation. *Fundamentals of traffic simulation*, pages 1–62, 2010.
- [17] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pages 2575–2582. IEEE, 2018.
- [18] Ranju Mohan and Gitakrishnan Ramadurai. State-of-the art of macroscopic traffic flow modelling. *International Journal of Advances in Engineering Sciences and Applied Mathematics*, 5:158–176, 2013.
- [19] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. Sumo—simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind, 2011.

- [20] Axel Wegener, Michał Piórkowski, Maxim Raya, Horst Hellbrück, Stefan Fischer, and Jean-Pierre Hubaux. Traci: an interface for coupling road traffic and network simulators. In *Proceedings of the 11th communications and networking simulation symposium*, pages 155–163, 2008.
- [21] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.
- [22] DT Pham and PTN Pham. Artificial intelligence in engineering. *International Journal of Machine Tools and Manufacture*, 39(6):937–949, 1999.
- [23] Fabio Galbusera, Gloria Casaroli, and Tito Bassani. Artificial intelligence and machine learning in spine research. *JOR spine*, 2(1):e1044, 2019.
- [24] Zoltán Szabó. Dr. bokor, józsef dr. gáspár, péter. 2014.
- [25] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval*, pages 21–49. Springer, 2008.
- [26] Junjie Peng, Elizabeth Jury, Pierre Dönnès, and Coziana Ciurtin. Machine learning techniques for personalised medicine approaches in immune-mediated chronic inflammatory diseases: Applications and challenges. *Frontiers in Pharmacology*, 12, 09 2021.
- [27] Bing Liu and Bing Liu. Supervised learning. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*, pages 63–132, 2011.
- [28] Vladimir Nasteski. An overview of the supervised machine learning methods. *Horizons. b*, 4:51–62, 2017.
- [29] Bálint Kövári, Bálint Pelenczei, and Tamás Bécsi. Monte carlo tree search to compare reward functions for reinforcement learning. In *2022 IEEE 16th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pages 000123–000128. IEEE, 2022.
- [30] Zoubin Ghahramani. Unsupervised learning. In *Summer school on machine learning*, pages 72–112. Springer, 2003.
- [31] M Emre Celebi and Kemal Aydin. *Unsupervised learning algorithms*, volume 9. Springer, 2016.

- [32] Batta Mahesh. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR)*.*[Internet]*, 9(1):381–386, 2020.
- [33] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers & Geosciences*, 19(3):303–342, 1993.
- [34] Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.
- [35] Ayush Singh. Reinforcement learning: Markov decision process (part 1). *Towards data science*, 2018.
- [36] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [37] Máté Kolat, Bálint Kővári, Tamás Bécsi, and Szilárd Aradi. Multi-agent reinforcement learning for traffic signal control: A cooperative approach. *Sustainability*, 15(4):3479, 2023.
- [38] Afshin Oroojlooy and Davood Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence*, 53(11):13677–13722, 2023.
- [39] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. Multi-agent reinforcement learning: An overview. *Innovations in multi-agent systems and applications-1*, pages 183–221, 2010.
- [40] Dipti Srinivasan. *Innovations in Multi-Agent Systems and Application-1*, volume 310. Springer, 2010.
- [41] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316, 2017.
- [42] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [43] Daniel Soutner and Luděk Müller. Application of lstm neural networks in language modelling. In *Text, Speech, and Dialogue: 16th International Conference, TSD 2013, Pilsen, Czech Republic, September 1-5, 2013. Proceedings 16*, pages 105–112. Springer, 2013.
- [44] John Violos, Evangelos Psomakelis, Dimitrios Danopoulos, Stylianos Tsanakas, and Theodora Varvarigou. Using lstm neural networks as resource utilization predictors: The case of training deep learning models on the edge. In *Economics*

of Grids, Clouds, Systems, and Services: 17th International Conference, GECON 2020, Izola, Slovenia, September 15–17, 2020, Revised Selected Papers 17, pages 67–74. Springer, 2020.

- [45] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.
- [46] Daniel Krajzewicz, Stefan Hausberger, Peter Wagner, Michael Behrisch, and Mario Krumnow. Second generation of pollutant emission models for sumo. In *SUMO2014 - Second SUMO User Conference*, Reports of the DLR-Institute of Transportation Systems, May 2014.
- [47] Julio Sanguesa, J. Barrachina, Manuel Fogue, Piedad Garrido, Francisco Martinez, Juan-Carlos Cano, Carlos Calafate, and Pietro Manzoni. Sensing traffic density combining v2v and v2i wireless communications. *Sensors*, 15:31794–31810, 12 2015.