

Homogén ágensek alkalmazkodóképessége dinamikusan skálázható forgalmi környezetben

Szerző:

Knáb István Gellért



Közlekedésmérnöki és Járműmérnöki Kar

Konzulens:

Kővári Bálint

OTDK dolgozat

Tartalomjegyzék

1. Bevezető	1
1.1. Probléma feltárása	1
1.2. Kapcsolódó kutatások	3
1.3. Megoldási kísérlet	5
2. Algoritmusok	6
2.1. Gépi Tanulás	6
2.2. Megerősítéses Tanulás	9
2.3. Deep Q-Network algoritmus	11
2.4. Multi-ágens Megerősítéses Tanulás	13
3. Környezet	15
3.1. Szimulátor	16
3.2. Gym környezet	17
3.3. Problémaspecifikus implementáció	17
3.3.1. Állapotreprézntáció	18
3.3.2. Action Space	18
3.3.3. Jutalmazási stratégia	19
4. Eredmények	20
4.1. Metrikák a kiértékelésben	20
4.2. Algoritmusok összehasonlítása	21
5. Konklúzió	25

Kivonat

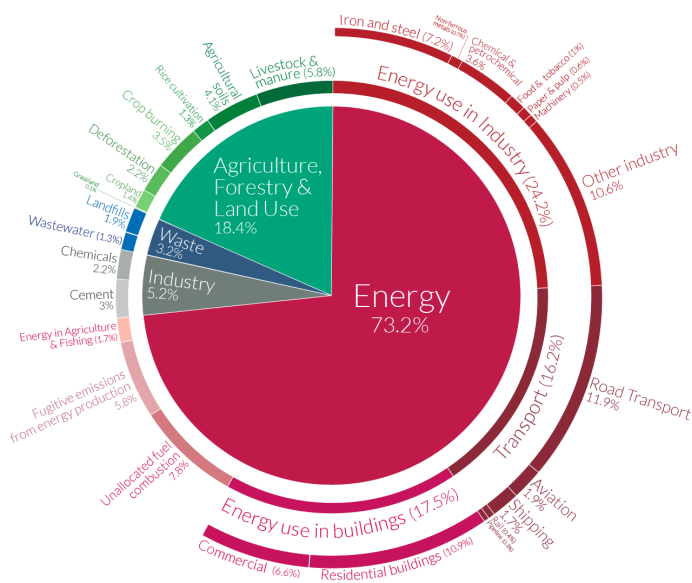
A modern városok arculatának formálásában központi téma a fenntarthatóság. Mivel a közlekedés jelentős részét teszi ki az üvegházhatású gázok emissziójának, az ezen a téren korszerű algoritmusok által meghozott beavatkozások nagy mértékben hozzájárulnak az élhetőbb városok kialakításához. Abból kifolyólag, hogy az infrastruktúra térbeli kiterjesztése nem minden esetben lehetséges, más jellegű megközelítésekben keresendő a megoldás. Itt jönnek képbe az intelligens forgalomirányító rendszerek, városi környezetek tekintetében többnyire forgalmi jelzőlámpák eseményvezérelt kapcsolásán keresztül. Az európai nagyvárosok jelentős részében már jelen vannak ilyen megoldások, de ezek többnyire szabályalapú implementációk, melyek nem képesek kognitív döntéshozásra. A forgalomban gyakran fellépő kumulatív jellegű hibák kiküszöbölésére viszont olyan prediktív döntéshozásra van szükség, ami nem csak feloldja, hanem meg is előzi a nem kívánatos forgalmi hullámok kialakulását. Az eddigi mesterséges intelligencia alapú megoldások számos alkalommal bizonyították alkalmasságukat a feladatra, akár több kereszteződés kooperatív irányítása esetén. A dolgozat témája az eddigi kutatások kiterjesztése egy általánosabb alkalmazhatóság felé, ahol egyetlen környezetben elvégzett tanítás során előálló ágens különböző számú kereszteződéssel rendelkező hálózatokon is alkalmazásra kerül, ezzel további lépéseket téve a fenntarthatóság felé mind a lerövidült tanítási időnek, mind pedig a bevethetőségig eltelt idő csökkenésének köszönhetően.

1. fejezet

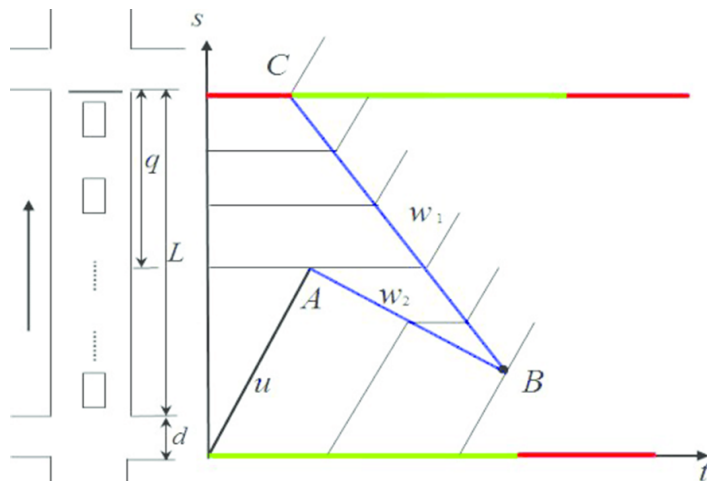
Bevezető

1.1. Probléma feltárása

Napjaink városaiban fő kihívás a jó minőségű levegő megtartása, különösen a sűrűn lakott övezetekben[1]. Ez természetesen több szektorral is összefüggésben van, mint az ipar, a szolgáltatások és nem utolsó sorban a közlekedés. Utóbbi jelentős részt vállal belőle, ahogy az az 1.1 ábrán látható. Ebből következően az itt eszközölt változtatások már a szektoron belüli százalékos nagyságrendű javulás esetén is szignifikáns tényezők lehetnek az összképet tekintve. Itt a növekvő számú elektromos és alternatív hajtásúkkal üzemelő járművek [2] mellett figyelembe veendő a forgalom irányítása is, amin keresztül a járművek összehangolt mozgása elősegíti a 1.2 ábrán látható reakcióidőkből eredő tranziensek[3] visszaszorítását.



1.1. ábra. Kibocsátás mértéke szektoronként csoportosítva. [4].



1.2. ábra. Várakozási idők szemléltetése[5].

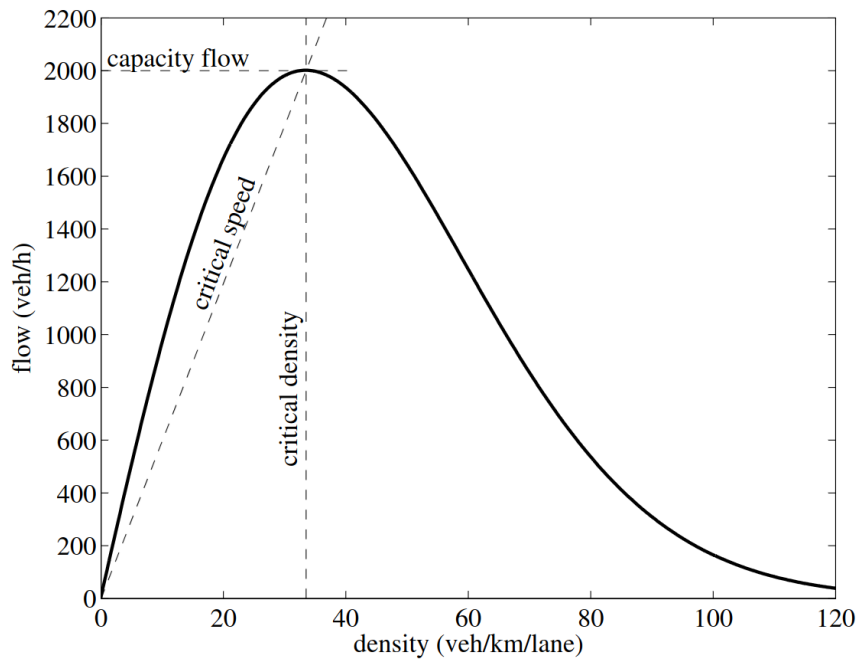
A reakcióidők mellett a járművek tehetetlenségéből is adódnak veszteségek egy forgalmi hálózaton, a kettőből együtt áll össze a kívánt mozgás eléréséig szükséges idő, a τ . Amint az 1.2 ábrán látható, egy lámpa zöldre váltása esetén ez az idő összeadódva ér végig a járműoszlopon, ezzel az adott jármű mozgásának megkezdéséig a jelzéstől eltelt idő az alábbi módon írható le:

$$\tau = \sum_{i=1}^n \tau_i, \quad (1.1)$$

ahol az n az előtte lévő járművek számát jelöli. Ebből tehát belátható, minél kisebb járműoszlopokat készítet a rendszer megállásra, annál hatékonyabb működés érhető el, hiszen minden τ_i részt vesz a mögötte érkező járművek mozgásának korlátozásában, ezzel egy kumulatív hátráltató tényezőként lép fel.

Bár a megoldás adódna a további forgalmi sávok létrehozásában, különösen nagy történelmi múlttal rendelkező városok esetén ez kivitelezhetetlen[6]. Ennek oka abban rejlik, hogy kialakulásuk idején nem a jelenlegi forgalomra tervezték a városokat, valamint az épületek jelentős része műemlékvédelem alatt áll, így nem bonthatóak le a forgalmi hálózat expanziójának érdekében. Emellett nem is érdemes minden esetben az infrastruktúra térbeli kiterjesztéséhez nyúlni, hiszen csak a kifejezetten frekvenciált időszakokban közelítené meg a kihasználtság a hálózat korlátait, ami anyagi szempontból nem bizonyul jövedelmezőnek.

Egy alternatívaként kínálkozik az Intelligent Transportation Systems (ITS)[7], aminek célja különböző forgalmi helyzetekben olyan eseményvezérelt beavatkozások eszközölése, ami segíti az áramlás folytonosságát. Tekintve a 1.3 ábrát, látható hogy a sűrűség növekedése a forgalmi hálózaton egy idő után nem realizál nagyobb áteresztést, sőt, kontraproduktívan is hat. Ezeknek a rendszereknek a célja a sűrűséget a



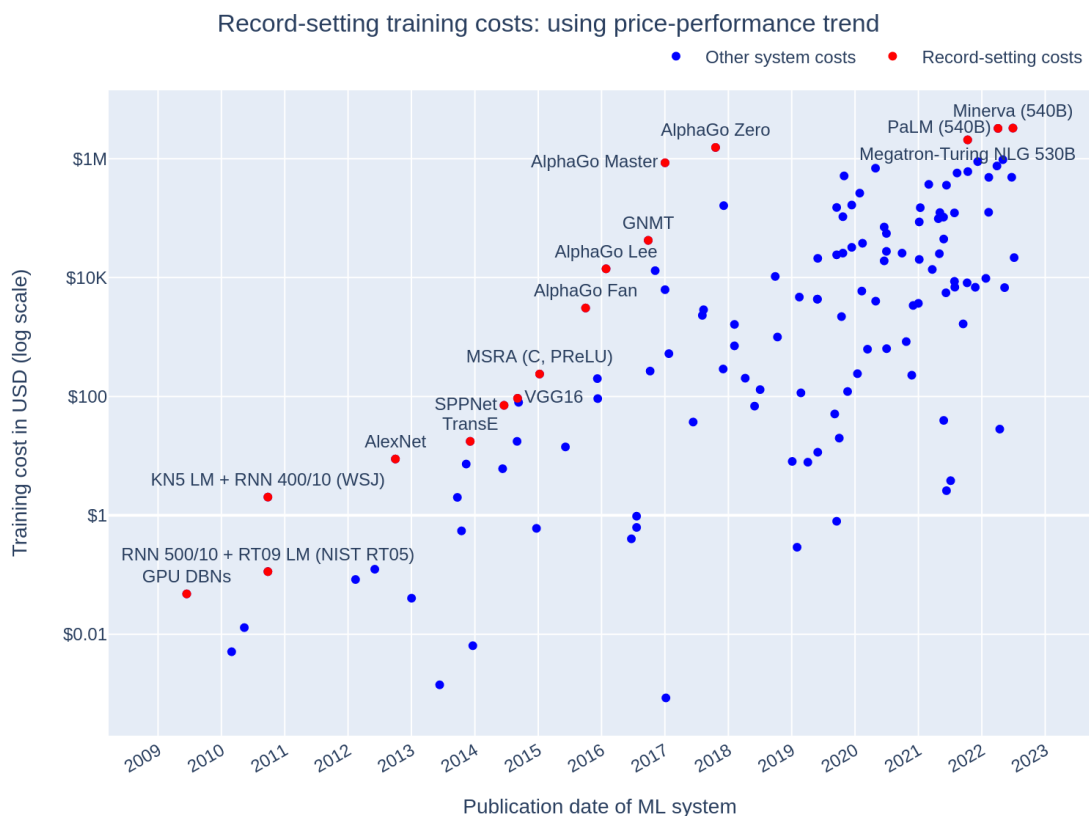
1.3. ábra. Fundamentális diagram[8]

maximális áteresztőképesség közelében tartani, ami a diagramon jelölt kritikus sűrűség környezetét jelenti.

ITS-re mind városi környezetben, mind pedig autópályák esetében szükség van. Előbbit tekintve a legnépszerűbb megközelítés a Traffic Signal Control (TSC) [9, 10], ahol a jelzőlámpák eseményvezérelt irányításán keresztül fokozható a hálózat áteresztőképessége. Nagysebességű hálózatokon előszeretettel alkalmazott megoldás a Variable Speed Limit Control (VSLC) [11, 12], ahol a cellákra osztott út egyéni sebességhatárok segítségével végzi el az adott cella ürítését vagy telítését. Mindkettő esetében olyan kooperatív döntéshozásra van szükség, ahol nem csupán a lokális környezet megfelelő irányítása a cél, hanem egy hálózati optimum keresése.

1.2. Kapcsolódó kutatások

Számos nagyvárosban fellelhetőek már a kezdetleges ITS megoldások. Algoritmusok terén több implementáció is megfigyelhető, melyek többnyire valamilyen szabály alapú viselkedést valósítanak meg. Ezek ismertetése során érdemes megemlíteni a SCATS (Sydney Coordinated Adaptive Traffic System) [13], GreenWave és RHODES (Real-time Hierarchical Optimized Distributed Effective System) [14] algoritmusokat, hiszen jelenleg is üzemelő rendszerek egy része ezek az algoritmusok alapján irányítja a lámpákat.



1.4. ábra. Korszerű DL modellek tanítási költsége[17]

Emellett elkezdtek megjelenni soft computingon alapuló realizációk is, és szimulációkban már több ízben is megmutatkozott, hogy van létjogosultságuk a Mély tanulást (Deep Learning - DL) alkalmazó rendszereknek is, köszönhetően a kognitív döntéshozásuknak. Fő előnyük, hogy nem csak a forgalom aktuális helyzetéről alkotott kép alapján indul el egy beavatkozási folyamat, hanem prediktíven hoznak döntést, ezzel nem csupán feloldva a már létrejött mozgó forgalmi hullámokat, hanem meg is előzve kialakulásukat. További előnye a hagyományos irányításelméleti megoldásokkal szemben, hogy nem szükséges a fizikai egyenletek meghatározása és a modell identifikáció bonyolult feladata, cserébe nem biztosít matematikai garanciát a hibátlan működésre, mivel black-box rendszerről van szó.

A szakirodalomban a Reinforcement Learning(RL) többkresztesződéses rendszerek kooperatív irányítását is lehetővé teszi[15][16], köszönhetően az ágensek azonos felépítésének könnyen tanítható rendszerek állnak elő. Viszont tekintve a 1.4 ábrát kirajzolódik, hogy ilyen metodológiák alkalmazása esetén is bizonyos megfontolásokkal kell élni a megfelelő megoldás eléréséhez. Bár általában az ilyen feladatok kis neurális hálózatokat használnak, mégis napokban mérhető sokszor a tanítás ideje. DL alapú

rendszereknél napjaink egyik fő törekvése ezt az időt lerövidíteni, akár a hatékonyabb tanítóminták felhasználásával[18], akár előre kiértékelt jutalmazási stratégiákkal[19], vagy éppen az univerzálisan felhasználható ágenseken keresztül.

1.3. Megoldási kísérlet

A tanítási idő csökkenésének reményében a felsoroltak közül a harmadik módszer alkalmazása kerül a középpontba. A megoldás egy csúszó ablak módszerben rejlik, ahol egyszerre egy kereszteződés dolgozik, viszont minden lépésben ez az ablak végigcsúszik az egész hálózaton, ezzel azonos alfeladatokra osztva a hálózat irányítását.

A megfelelően megfogalmazott absztrakciók lehetővé teszik a tetszőleges méreten való alkalmazhatóságot, mivel mindig csak az aktuálisan érintett szomszédos zónák képzik az állapotrepresentáció részét, viszont az átlapolások kapcsolatot teremtenek köztük. A kutatás hozzáadott értéke olyan állapotrepresentáció, beavatkozási tér és jutalmazási struktúra kialakítása, ami skálázhatóvá teszi a környezetet, és mindig egy releváns részt lefedve valósulnak meg a döntések. Ezek megfogalmazását követően egy homogén ágensekből álló hálózaton valósul meg a tanítás, aminek sikerességét követően 4 különböző számú kereszteződéssel rendelkező hálózaton kerül kiértékelésre, mind olyan forgalmat leíró makroszkópikus metrikákat illetően mint a várakozási idő vagy a feltartóztatott járművek száma, mind pedig üvegházhatású gázok kibocsájtása megfigyelére kerül.

A kutatás célja megmutatni, hogy a megfelelően megfogalmazott absztrakciók lehetővé teszik a tanítási idő csökkentését a környezet tesztelés során való újráskálázása mentén, amennyiben homogén ágensekből álló környezet beavatkozóinak számát változtatva azonos architektúrájú ágensek maradnak továbbra is. Az általános alkalmazhatóság tekintetében további feladat, hogy az alkalmazott szimulátor elérhető forgalomirányítási modelljeinél alkalmasabb megoldást kínáljon a dolgozat témáját is képző Multi-ágens megerősítéses tanulás(Multi Agent Reinforcement Learning - MARL) alapú rendszer, mind az üvegházhatású gázok csökkentésének tekintetében, mind pedig az áteresztőképesség növelését megfigyelve.

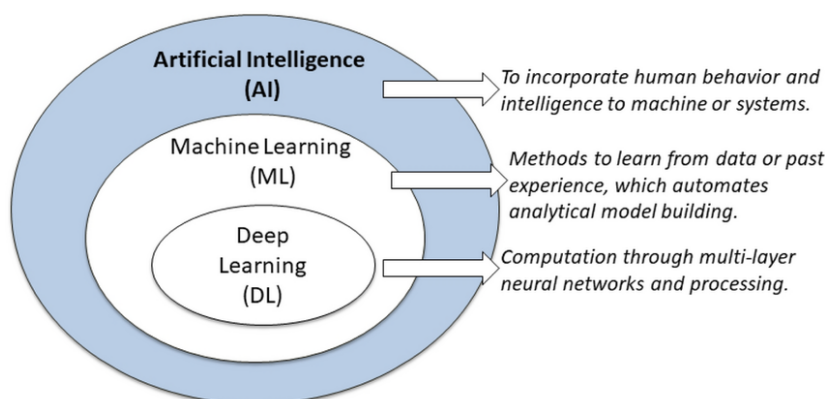
2. fejezet

Algoritmusok

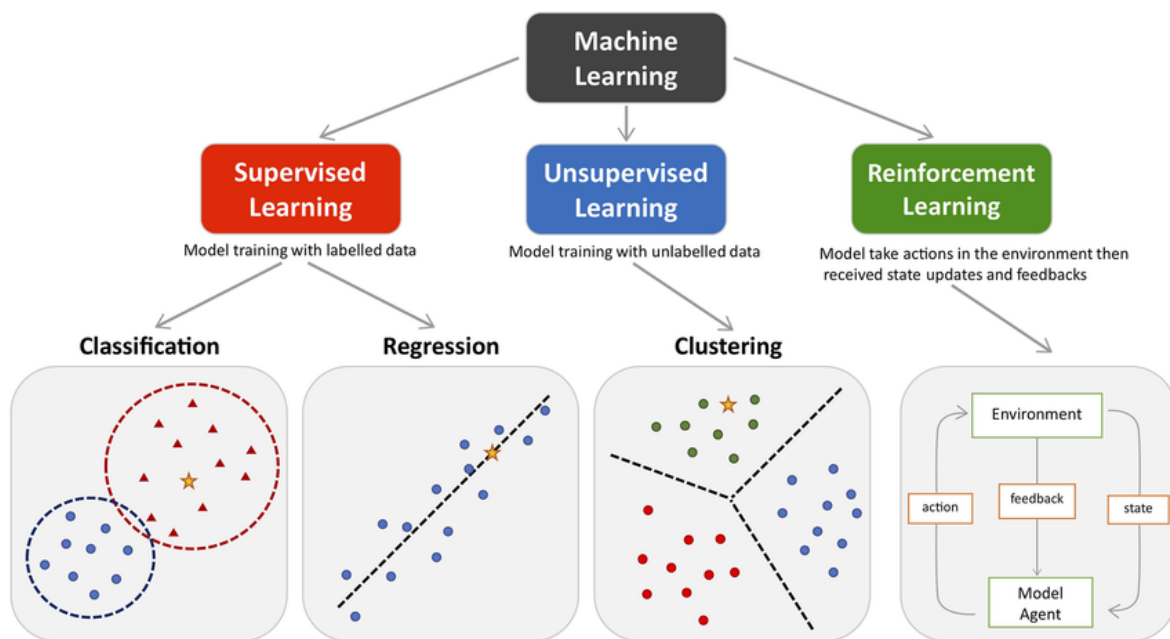
2.1. Gépi Tanulás

A mesterséges intelligencia (Artificial Intelligence - AI)[20] térnyerése napjaink egyik legjelentősebb témájává nőtte ki magát az elmúlt évek során. A területen érintett algoritmusok képesek olyan feladatok elvégzésére, amelyek hagyományosan emberi intelligenciát igényelnének, és gyakran nagyobb hatékonysággal és sikerrel, mint a klasszikus, fizikai modelleken alapuló megközelítések.

Az AI alkalmazási területei rendkívül sokrétűek, és számos iparágban forradalmasítják életvitelünket. A hétköznapi megoldásokat tekintve a keresőmotorok és ajánló rendszerek jelentek meg elsőként a digitális világban, ahol a kedvelések és keresési előzmények alapján hangolt javaslatok segítik a kívánt oldal könnyebb elérését. Emellett a frissen berobbanó a nyelvi modellek[21] fejlődése jelenti az emberek nagyrészeének AI első igazi széleskörű használatának megnyilvánulását, hiszen sokkal egyszerűbb bonyolult problémákra gyors és összetett választ kapni.



2.1. ábra. A gépi tanulás[22]

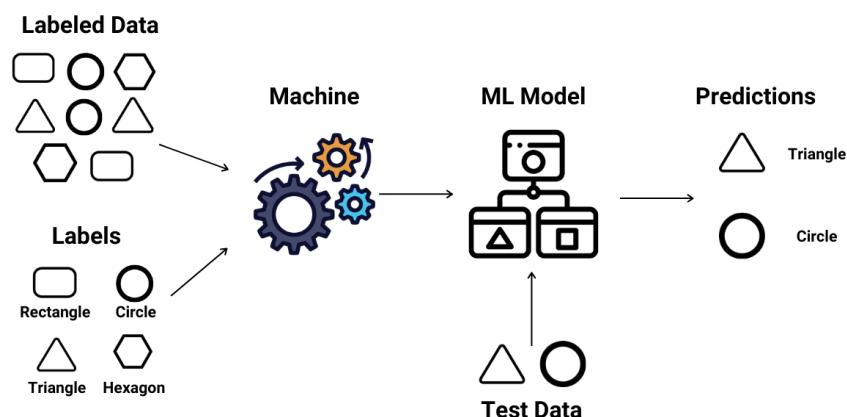


2.2. ábra. A gépi tanulás fajtái[24]

A hétköznapi felhasználás és a szórakoztatóipari megoldások mellett a műszaki világ is egyre nagyobb mértékben válik érintetté az AI alapú rendszerek integrálásában. Az AI egyik alcsoportját képző Gépi Tanulás (Machine Learning - ML) olyan problémák megoldását is lehetővé teszi, melyek a bonyolult leírhatóság és az elhanyagolásból keletkező hibák következtében még nem voltak a megfelelő teljesítménnyel elvégezhetőek. Ezen belül található a DL, ami az egyes feladatokat Mély Neurális Hálózatok segítségével valósítja meg, melynek a kimeneti rétegén megjelenő hibákat visszaterjesztve elvégezhető a tanítás. Az ML 3 további alcsoportra osztható ahogy az a 2.2 ábrán is látható, melyek a Supervised Learning (SL)[23], az Unsupervised Learning (USL) és a Reinforcement Learning. Ezek felhasználási területei rendkívül széles spektrumot ölelnek fel, ezáltal számos különböző feladat hatékony elvégzésére alkalmasak.

Az SL, amint a 2.3 ábrán látható előre címkézett adatokkal dolgozik, ahol a bemeneti adathoz hozzá van rendelve a megfelelő kimenet. Alapvető feladatai közé a regresszió és a klasszifikáció tartozik, ami a kimeneteken nyilvánul meg, ugyanis a regresszió folytonos kimenetet szolgáltat, és a célja a több dimenziós terekben egy folytonos függvényt illeszteni ami a legjobban leírja a problémát, míg klasszifikáció esetén a feladat az egyes osztályok megfelelő hozzárendelése a bemeneti adathoz. Az SL egyik legnépszerűbb felhasználása a számítógépes látórendszerek területe[25][26]. A legsűrűbben elvégzett feladatok a szegmentáció, detekció és klasszifikáció. A DL és NN jelentősége ezen a területen 2012-ben mutatkozott meg elsőként. Az ILSVRC[27] versenyen az azt megelőző évben győztes módszerhez képest, ami egyaránt használta a SIFT, HOG és SVM algoritmusokat az AlexNet 11% javulást volt képes felmutatni.

Supervised Learning

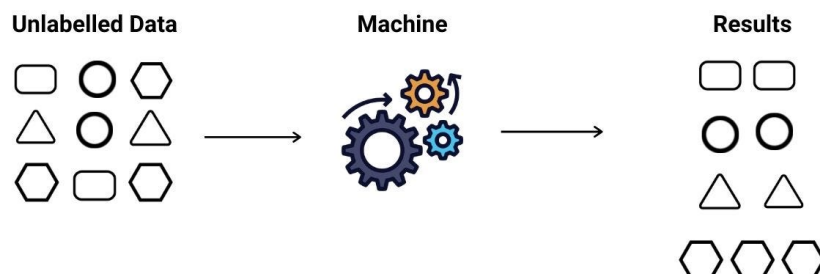


2.3. ábra. Supervised Learning[28]

A regresszió alkalmazása kedvelt mind a pénzügyi és ingatlanpiaci előrejelzésekben és elemzésekben, de egészségügyben is fellelhetőek a felhasználásai.

A USL szintén előre létrehozott adatokkal dolgozik, azonban azzal a különbséggel, hogy – amint a 2.4 ábrán is látható – ezekhez az adatokhoz nem tartoznak címkék. Ennek megfelelően az algoritmus célja olyan rejtett struktúrák feltárása, amelyek alapján az adatok klaszterekbe rendezhetők, vagy dimenziójuk csökkenthető. A DL a 2010-es évek első felében ezen a területen is jelentős fellendülést hozott, mivel az Autoencoder-ek újjáéledése, valamint a Generatív Adverzariális Hálózatok megjelenése új dimenziót nyitott a címkézetlen adatokból kinyerhető információk feldolgozásában és hasznosításában.

Unsupervised Learning



2.4. ábra. Unsupervised Learning[28]

Bár az RL az iparban a legkisebb részesedéssel bíró ágazat, mégis kedvelt kutatási terület. Ennek oka, hogy az adatok tanítás közben keletkeznek, az ágens tapasztalatai révén, és olyan szekvenciális problémák megoldására alkalmas, amelyekre az ML két másik fajtája nem használható.

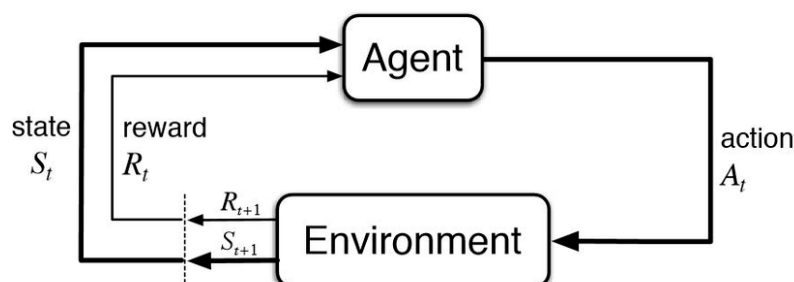
2.2. Megerősítéses Tanulás

Az RL alapvető működése két osztályt foglal magába, ahol az ágens feladata egy olyan π döntéshozási struktúra elsajátítása, ami alapján a legnagyobb jutalomhoz juthat hozzá adott környezetben. Amint a 2.5 ábrán látható, a diszkrétizált időlépésekben adott S_t állapotban az ágens meghoz egy A_t döntést, amire válaszul a környezettől egy S_{t+1} állapotot, valamint R_{t+1} rewardot kap, ami a döntés minőségét egy skalárral csatolja vissza az ágens bemenetére.

Ezekből a jellemzőkből állíthatóak össze az állapotátmenetek, melyek során az ágens tanítás során egy ideális döntéshozási szekvenciát tanul meg. Az állapotátmenetekhez valószínűségek rendelhetők és ezek segítségével foglalható a Markov Döntési Folyamat keretrendszerébe az RL. Ennek elemei a következők[30]:

- S : Állapotok halmaza
- A : Megléphető beavatkozások halmaza
- R : S állapotban A beavatkozásra kapott közvetlen jutalom
- P : annak a valószínűsége, hogy S állapotban meghozott A döntésre a rendszer S' állapotba kerül

Az egyes állapotokból τ trajektóriák épülnek fel, amik az ágens jellegétől függően vagy kiértékelésre kerülnek az aktuális π policy mentén, vagy pedig az állapotátmenetek eltárolásra kerülnek egy memóriában, és a későbbiekben ezekből mintavételezve valósul meg a tanulás.



2.5. ábra. Megerősítéses tanulás[29]

Mivel az ágens feladata nem csupán egy adott helyzetben egy ideális döntés meghozása, hanem az epizód során elérhető maximális reward realizálása, ezek kumulált értéke amit maximalizálni kell a tanítás végére. Érdekes megfigyelni, hogy az adott állapottól messzebb lévő értékek kevésbé fontos tapasztalatok, ugyanis amennyiben a korábbiak sérülnek lehet hogy el sem jutunk abba az adott állapotba. Ez instabilitáshoz vezethet a tanítás során, ezért az azonos súllyal való összeadás nem a megfelelő megoldás. Erre került bevezetésre a discount factor, ami egy 0 és 1 közötti szám, ami minden lépésben beszorozva csökkenti az egyre távolabbi állapotokhoz tarozó rewardot. Az ilyen módon létrejövő érték a return, ami az alábbi módon írható le:

$$G_t = \sum_{t=0}^T \gamma^t \cdot r_t, \quad (2.1)$$

priorizálva a közelebb lévő állapotok magasabb értékét.

Az állapotátmenetekhez minősége a $Q(s, a)$ értékfüggvénnyel jellemezhető, ami azt adja meg, az ágens adott állapotban meghozott beavatkozásra milyen jutalomra számíthat. Az egyes $Q(s, a)$ értékek egymás után a Bellman-egyenlet segítségével számíthatóak, ami az alábbi módon teremt kapcsolatot 2 állapotátmenet között:

$$Q(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) \max_{a'} Q(s', a'), \quad (2.2)$$

ahol a bal oldalon lévő $Q(s, a)$ érték az állapotátmenethez tartozó rewardból és a maximálisan elérhető következő $Q(s', a')$ értékek és hozzájuk tartozó $P(s'|s, a)$ állapotátmeneti valószínűségek szorzatösszegéből áll.

A $Q(s, a)$ értékek mentén már jellemezve vannak az egyes állapotátmenetek egy skalárral, valamint a γ segítségével a láncolatuk súlyozása is előáll, viszont a tapasztalatszerzési stratégia továbbra is ismeretlen. Mivel az ágens tapasztalatai alapján hozza meg döntéseit, viszont a kezdeti szakaszban még nem rendelkezik ezzel, fel kell fedeznie környezetét. Az RL központi problémája az exploration-exploitation dilemma, ugyanis amennyiben az ágens nem rendelkezik megfelelő tapasztalatokkal a környezetet illetően, szuboptimális megoldásokat talál majd mint legjobb döntés. Abban az esetben, ha teljes mértékben csak a tapasztalatok gyűjtésére fókuszál, egy véletlenszerű viselkedést valósít meg.

Az RL algoritmusok két fő nagy családja a Model-Based és a Model-Free RL, utóbbit vizsgálva az ágensek 3 fő csoportja áll elő. Az első esetben az ágens közvetlenül a π függvényt tanulja, míg az érték alapú rendszerekben egy értékfüggvényt tanul meg az ágens. Ezeket kombinálja az Actor-Critic megközelítés. Csoportosíthatóak a bemenetek és kimenetek alapján, ezek lehetnek diszkrét vagy folytonos értékek. A közlekedés irányításban általában diszkrét kimenetet igénylő problémák merülnek fel, így a Deep

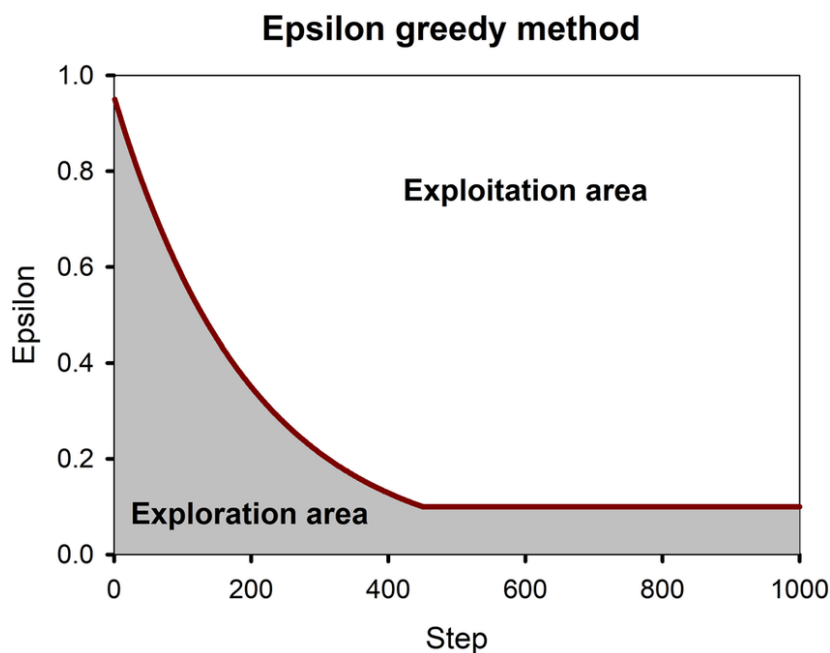
Q-Network algoritmus kerül használatra.

2.3. Deep Q-Network algoritmus

Az RL algoritmusok két nagy családba sorolhatóak, a modellalapú és modellenlélküli ágazatba. Ezek tovább csoportosíthatóak a tanítás módját tekintve, ugyanis egyes algoritmusok közvetlenül a π stratégiát tanulják, míg mások egy $Q(s, a)$ értékfüggvényt. Utóbbi egyik kedvelt algoritmus a Deep Q-Network (DQN), aminek 2015-ös megjelenése a DRL egyik legjelentősebb története volt[31]. Itt számos feladatban konvolúciós NN segítségével az emberi teljesítőképességet meghaladó minőségű döntéshozás került bemutatásra több tucat játékon. A DQN napjainkban is az egyik legkedvezőbb választás diszkrét kimenetet igénylő problémák esetére.

Az algoritmus két NN-t használ tanulása során, melyek közül egyik a modell hálózat, ami mindent iteráció során frissíti súlyait, valamint van egy target hálózat, ami bizonyos időközönként lemásolja a modell hálózat súlyait. Mivel egy folyamatosan frissülő hálózat instabil tanulást eredményezne, hiszen a súlyok módosítása torzítaná a referenciaértéket is, a célhálózat lassabb frissülése küszöböli ki ezt a problémát, csökkentve annak esélyét, hogy az algoritmus a rossz referenciák miatt nem megfelelő irányba konvergáljon.

Az algoritmus felépítését tekintve látható a 1.pseudokódban, hogy az inicializáló szakasz csupán a neurális hálózatok és a memória osztály létrehozására szolgál. Ezt követően minden egyes epizódban a környezet alaphelyzetbe kerül. Az egyes diszkrét

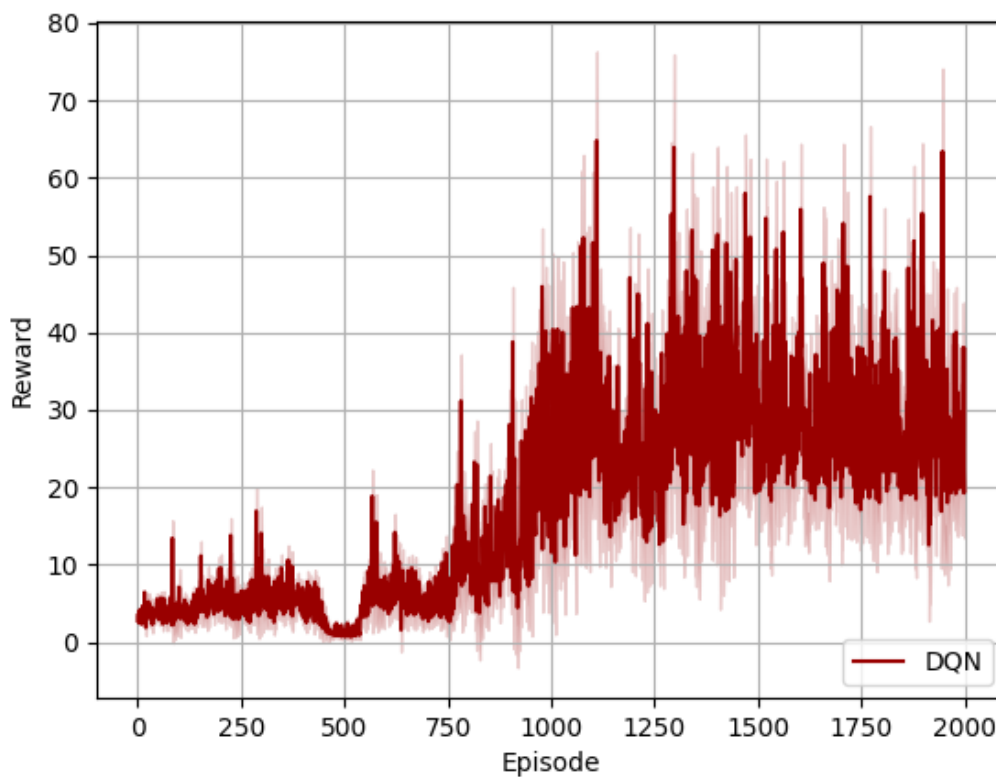


2.6. ábra. Az ϵ -greedy policy működése[32]

időlépésekben az ágens meglép beavatkozásokat, melyeket a modell hálózat szolgáltat. A döntések lehetnek véletlenszerű felfedező lépések vagy az ágens aktuális ismeretei szerinti optimális döntés. Ezt az 2.6 képen szemléltetett ϵ -greedy policy segítségével dönti el, ami az exponenciálisan csökkenő ϵ értékét hasonlítja össze egy véletlenszerűen generált számmal. A csökkenő érték biztosítja, hogy a kezdeti maximális felfedezés idővel az optimális döntések felé tolódjon, ahogy a diagram alatt és felett lévő területek arányából látható. A környezettel való interakciókat követően a memóriába eltárolásra kerülnek az egyes állapotátmenetek, amiből ezt követően egy meghatározott számú minta véletlenszerűen kiválasztásra kerül, hogy az algoritmus elvégezze rajta a hálózat hangolását. Ez a folyamat a Mean Squared Error(L_{MSE}) segítségével valósul meg:

$$L_{MSE} = \frac{1}{n} \sum_{i=1}^n (Q_{pred} - Q(s, a))^2 \quad (2.3)$$

ahol a target és modell hálózatok által szolgáltatott kimenetek közti különbség jelenti a hibát. Ennek a hibának egy értéken való beállása valamint a reward konvergenciája a környezet határát jelentő értékhez egyaránt indikátorai a sikeres tanítási folyamatnak.

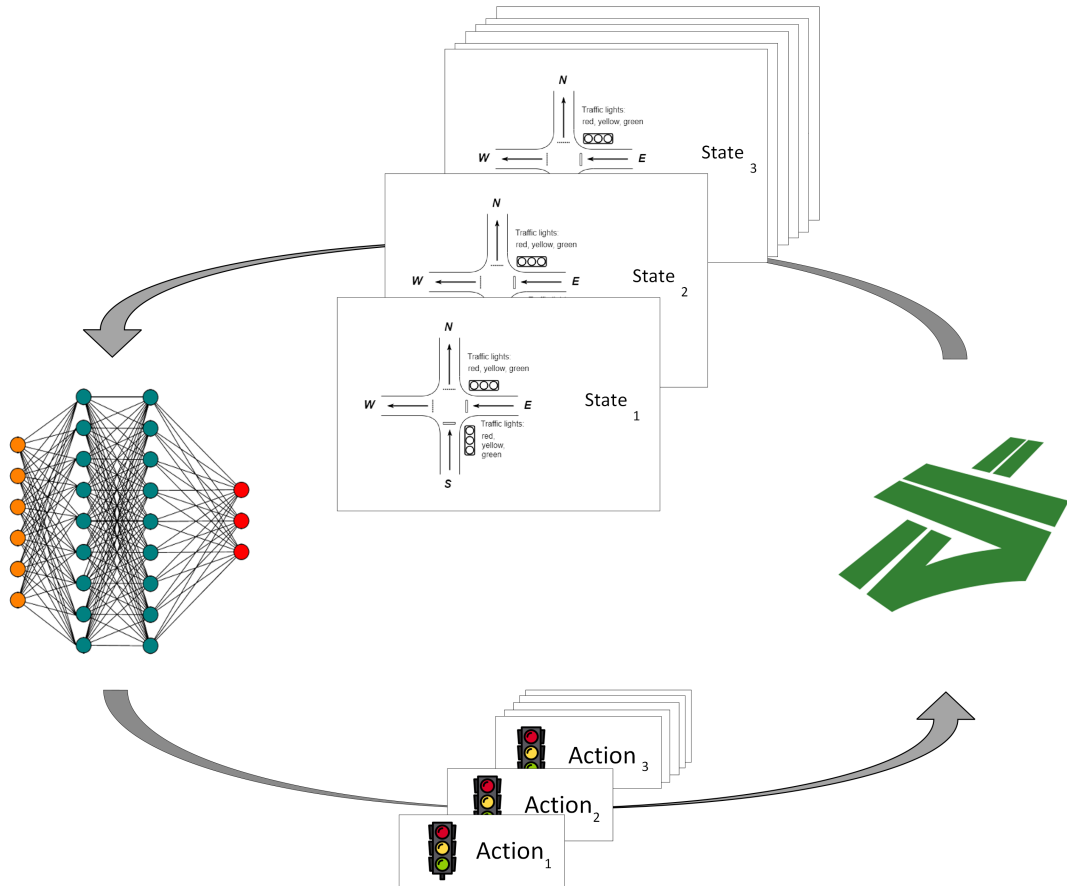


2.7. ábra. A tanítás konvergenciagörbéje.

2.4. Multi-ágens Megerősítéses Tanulás

Számos összetett probléma esetén célszerű a feladatot kisebb, jól definiált alfeladatokra bontani[33]. Az ilyen megközelítés a Multi-ágens Megerősítéses Tanulás (MARL), ami számos addig megoldhatatlannak tűnő feladatot képes elvégezni, valamint sokkal általánosabb működést szolgáltat újraméretezhető környezetekben. Ilyen összetett probléma egy robotkar irányítása, ahol a szervómotorok összehangolt mozgást végeznek, egyes logisztikai feladatok, valamint a forgalomirányításban is nagy mértékű javulás érhető el az irányítás ilyen módon való megfogalmazásával.

Az ágensek két fő csoportra oszthatóak. Amennyiben az ágensek architektúrája teljes mértékben megegyezik, homogén ágensekről van szó, amennyiben eltérést mutatnak, heterogén ágensek[34]. Egy másik osztályozásban viselkedésük alapján lehet elkülöníteni őket. A MARL célja nem csak az összehangolt feladatvégzés lehet, hanem egy ágens olyan módon is tanulhat, hogy mások ebben hátráltatják, azaz a környezetben több azonos célú ágens osztozik. Ebből látható, hogy az alapvető kooperatív jelleg mellett kompetitív viselkedés is megvalósítható, ami a tanítómintákra jó hatással lehet.



2.8. ábra. A Multi-ágens feladat problémáspecifikus implementációja

Emellett léteznek olyan neutrális ágensek, melyek bár mind hatnak a környezetre, mégis nem szándékosan végeznek összehangolt döntést és nem is áll szándékukban a másik dolgának nehezítése[35]. Ez az independent learner koncepciója, ahol bár osztoznak a környezeten és hasonló dolgokat figyelnek meg, de nincs szándékos összehangoltság döntéseikben. A dolgozatban is egy ilyen jellegű megoldás van implementálva, ugyanis az egyes ágensek bemenetein nem jelenik meg a többiek állapota és döntése, csupán egy átlapolás van az állapotokban. Köszönhetően ennek az átlapolásnak, valamint a csúszó-ablak jellegű megvalósításnak azonban mégis megvalósul egy indirekt kooperáció.

A hálózatszintű optimalizáció érdekében egy további tényező a megfelelő jutalmazás alkalmazása. A kompetitív játékok esetében kedvelt a zero sum jutalmazás, ebben az esetben viszont ez nem lenne célravezető. Sokkal inkább az identical payoff, ahol az összes ágens együtt kap jutalmat, ezzel minden beavatkozót arra ösztönözve, hogy az egész környezeten igyekezzon várakozásokat csökkenteni, ne csak lokális környezetében.

1. Algorithm MARL DQN[31]

```

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

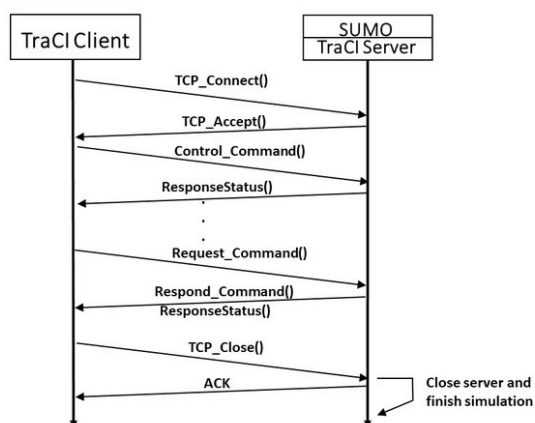
```

3. fejezet

Környezet

Reinforcement learning esetében, ahogy az a 2.2 részben tárgyalásra került két objektum kommunikációja valósul meg. Az ágens a metodológiában kifejtett DQN algoritmus, míg a környezet egy olyan forgalmi hálózat, ami dinamikusan változtatható számú beavatkozó pontokból áll.

A kommunikáció megvalósíthatóságának érdekében a szimulátorhoz csatolni kell egy Gymnasium struktúrájú Python környezetet, ami előkészíti a szenzoradatokat a beavatkozó számára és a kapott beavatkozó jeleket közli a szimulátor irányába. Ez a kommunikáció a TraCI-n keresztül valósul meg, ami TCP-IP kommunikációt megvalósító interfész, ami a 3.1 ábrán látható módon működik. A szimulátor és a python környezet együttesen képesek a forgalom állapotainak széleskörű leírására, valamint lehetőséget biztosítanak a MARL algoritmussal való irányításra.



3.1. ábra. A TraCI működése[36]

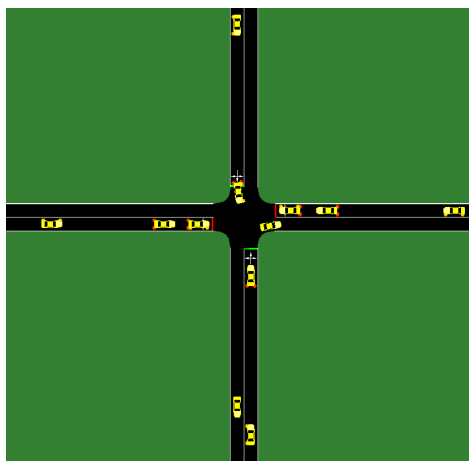
3.1. Szimulátor

A szimulációs környezetet a SUMO (Simulation of Urban MObility)[37] nyílt forráskódú szimulátor biztosítja. Számos előnye közül a legfontosabb motiváció az alkalmazására az, hogy kiválóan kommunikál a Pythonnal, valamint a makroszkopikus forgalmi adatok széles skálán elérhetőek a futás során. Ezek az értékek jellemzően sebességek, sűrűségek, valamint emissziók, melyek a bevezetőben kitűzött metrikák, így a szimulátor teljeskörűen alkalmazható a tanításra és kiértékelésre. Az adatok lekérhetőek forgalmi sávonként, csomópontként valamint járműszinten is.

A beavatkozó egységek egyszerű három fázist megvalósító jelzőlámpák, ahol minden sávból minden irányba lehet haladni és a kanyarodási szabály érvényes rájuk, nincsenek magasabb rendű utak. Mint a 3.2 képen is látható, minden esetben egy bemenő és egy kimenő sávval dolgozik a hálózat, ezzel az ágenseket teljes mértékben homogénné téve. Az alábbi kereszteződések közül került összeállításra 4 különböző méretű környezet, 1, 2, 3 és 4 kereszteződéssel.

Az infrastruktúra modellezésén túl fontos aspektus a forgalom modellezése is, hiszen olyan idősávban indokolt az ITS bevetése, ahol a forgalom mértéke meghaladja a hálózat kapacitását. Többféle eloszlás is generálható a SUMO-ban, melyek közül az óránként megjelenő járművek száma alapján került kiválasztásra, minden hálózaton egy kritikus értéket beállítva.

A lámpák egy rögzített fázisidő szerint kapcsolnak alapesetben, amire a SUMO beépített algoritmusai és a MARL gyorsabb beavatkozásokat eszközölhetnek amennyiben indokolt, viszont a minimális fázis ezekben az esetekben egyaránt 10 másodperc. Minden kiadott jelzés között a lámpa automatikusan sárgára vált.

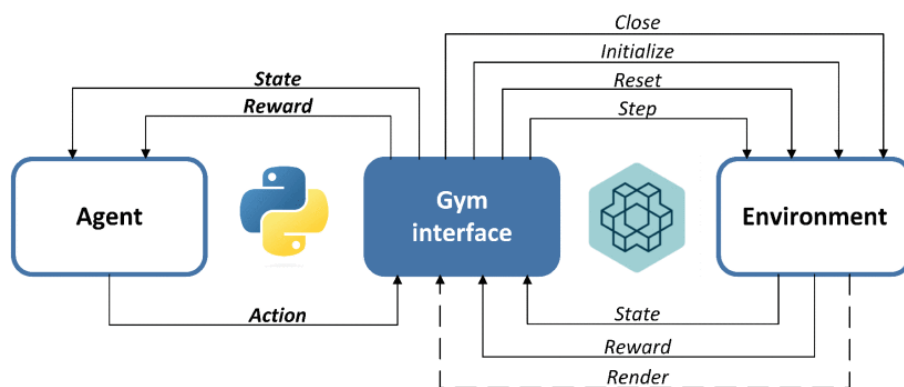


3.2. ábra. A SUMO kereszteződés felépítése

3.2. Gym környezet

Az RL algoritmusok összehasonlíthatósága érdekében az OpenAI létrehozta a Gym környezet architektúrát, ami szabványosságának köszönhetően jól kapcsolható különböző ágensekhez, ezzel metodológiákat és algoritmusokat hatékonyan és egyszerűen lehet kiértékelni. A Gym alapvető függvényei és kommunikált jelei a 3.3 ábrán láthatóak, ahol az inicializálás és bezárás mellett megtalálható egy függvény a környezet alaphelyzetbe állítására. Futás közben a lépések a `step()` függvénnyel vannak szimulálva, ahol az egyes meghozott döntésekre a környezet 5 értékkel tér vissza, melyek a következő állapot, a reward, az epizód végét jelző flag, a feltételek korai sérülését jelző flag valamint egy opcionális információkat tartalmazó dictionary. Mint a visszamenő ágban látható, rendelkezik egy opcionális `render()` metódussal, ami a vizuális kiértékelésért felel.

A Gymnasium oldalán nem csak a szabványos architektúra adott, hanem számos olyan környezet, ami a korszerű DRL kutatások terén meghatározónak számít. Ilyenek diszkrét beavatkozások esetén a `CartPole-v1`, az `Acrobot-v1` valamint az epizód végén jutalmazó algoritmusok esetén a `MountainCar-v0`. A folytonos beavatkozások terén is a környezetek széles skálája érhető el, az egyszerű `Pendulum-v1` egybeavatkozos környezettől a komplex Mujoco fizikai motort használó környezetekig.



3.3. ábra. A Gym interfész[38]

3.3. Problémaspecifikus implementáció

A korábbi megvalósításokkal szemben olyan RL absztrakciókra volt szükség, ami lehetővé teszi a tetszőleges méretű való alkalmazhatóságot. A dolgozatban bemutatott csúszóablakos rendszerben minden ágens lehetőséget kap 10 másodpercenként állapotot váltani, és amennyiben minden ágens döntött, a környezet újabb 10 lépést tesz a szimulátorban.

3.3.1. Állapotrepresentáció

Az állapotrepresentáció olyan makroszkópikus forgalmi adatokkal dolgozik, melyek elegendőek a shock wave detekciójára valamint a forgalom aktuális állapotának leírására. Ezek jellemzően a sűrűség és a sebesség szoktak lenni. Jelen esetben az utóbbi kerül alkalmazásra, minden kereszteződés esetén csupán a 4 lokális érték monitorozásával, az alábbi módon:

$$state_i = \begin{bmatrix} velocity_1 \\ velocity_2 \\ velocity_3 \\ velocity_4 \end{bmatrix} \quad (3.1)$$

Az állapotrepresentáció megfogalmazásának egyik fő célja volt, hogy olyan szűk lokális környezet kerüljön monitorozásra, ami minél több helyen bevezethető. A bemutatott módszer ismeretet ad az ágens számára az összes olyan tényezőről, amik közvetlen hatást gyakorolnak, viszont lehetővé teszi a használatot minden olyan hálózaton, ahol a beáramló utak száma megegyezik az ismertetettel. A homogenitáshoz való ragaszkodás így ebben az esetben egy hatalmas előrelépés az általános felhasználhatóság felé.

A célként megfogalmazott shock wave elkerülése érdekében megfigyelhető, hogy amennyiben az átlagsebesség hirtelen visszaesik, a környezetnek hirtelen kell változásokat eszközölni. Erre egyszerű szabályalapú algoritmusok is képesek lennének, viszont az igazi előny abban rejlik, hogy az ágens az ezt megelőző állapotokat is megtanulja, és mivel egy RL ágens egy optimális döntéshozási szekvenciára törekszik, ahol a G_t kerül maximalizálásra, alkalmas nem csak feloldani, hanem prediktív jellegének köszönhetően elkerülni is a jelenséget.

3.3.2. Action Space

A beavatkozások terén a forgalmi jelzőlámpák állapotai által adott kötöttségek okán a MARL megvalósításban rejlik a hozzáadott érték. A beavatkozások olyan csúszó ablak jelleggel valósulnak meg, ahol vagy a lámpák vízszintes vagy függőleges zöldre váltást engedélyeznek az alábbi módon:

$$action = \begin{bmatrix} North - South \\ East - West \end{bmatrix}, \quad (3.2)$$

minden sávból tetszőleges haladási irányt engedélyezve. A fázisok adott értékre vannak hangolva, de az ágensek a saját döntési időszakukban lerövidíthetik ezt a fázist, és megelőző beavatkozásokat hajthatnak végre. Minden váltás között legalább 10 másodpercnek el kell telni, mivel a forgalmi hálózatokon a hosszútávú stabilitás nem a

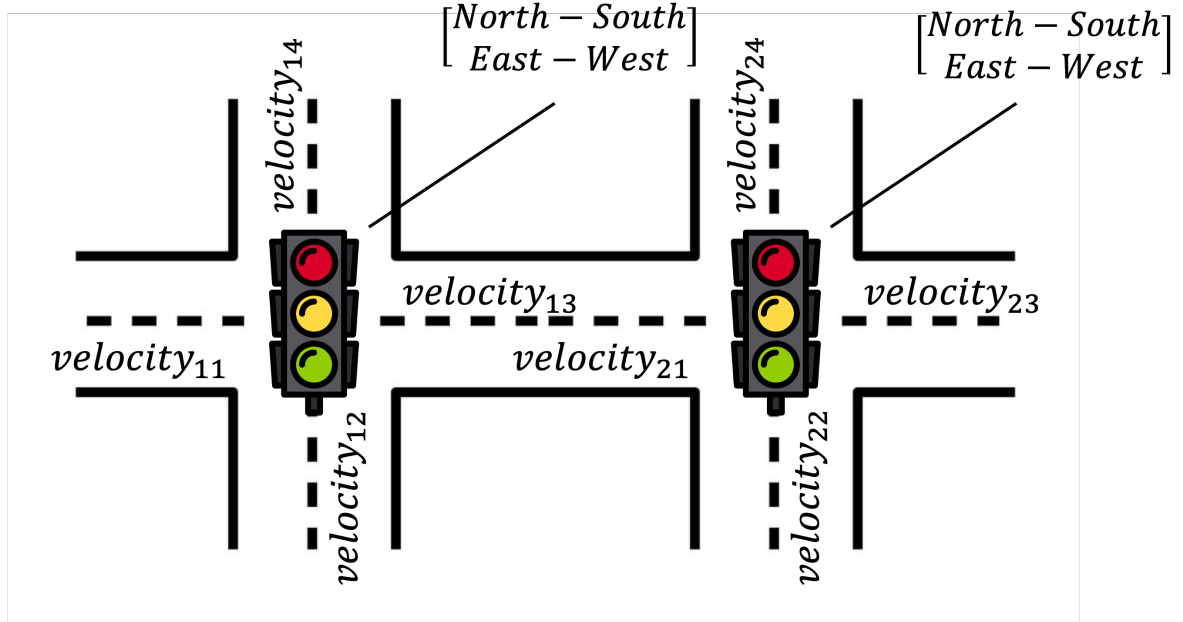
hirtelen döntések során érhető el.

3.3.3. Jutalmazási stratégia

A jutalmazási stratégia kialakítása során két fő motiváció vezérelte a fejlesztés irányát. Egyrészt a várakozási idők csökkentése volt a kitűzött cél, másrészt a nagyobb áteresztőképesség a realizált átlagsebesség növelésével. Ebből adódóan a sebesség a számlálóban kapott helyet, aminek minél nagyobb elért értéke egyre nagyobb jutalmakhoz vezet. A várakozási idők veszteségként értelmezettek a hálózaton, így ezek összege büntető tagként szerepel a jutalmazási eljárásban. A bevezetett metrikák segítségével az alábbi módon fogalmazható meg a jutalomfüggvény:

$$R = \frac{v_{avg}}{1 + w} \quad (3.3)$$

A kettő együttes alakulása a környezet méretétől és a forgalmi terheléstől függően egy meghatározott értékhez konvergál a sikeres tanítás esetén, ahol ideális esetben az elérhető legnagyobb sebesség feltartóztatott járművek nélkül valósul meg a hálózaton.



3.4. ábra. A környezet felépítésének szemléltetése

4. fejezet

Eredmények

4.1. Metrikák a kiértékelésben

A tanítási folyamat sikerességénél még nem ér véget a kiértékelés, ugyanis ezt más méreten is alkalmazni kell az univerzális felhasználhatóság bizonyításához. A 3 kereszteződéssel rendelkező csomóponton elvégzett tanítás által előálló modell 3 további méreten is ellenőrzésre kerül, ahogy az a kezdeti célkitűzésekben megfogalmazódott mint igény.

Az összehasonlító analízis két fajta metrikát fog érinteni. Először is a fenntartathatóság szempontjából nézve az emissziók alakulását, amit a becsült várakozási idők és feltartóztatott járművek száma követ mint folytonossági mutató. Ezekből tisztán leszűrhető a shock wave kialakulása, valamint az is látható, hány jármű vesz részt benne.

Az emissziók mérése a szimulátorban közelítő modellek segítségével történik. Erre több lehetőséget is kínál a SUMO, melyek közül a kiértékelés során az alapértelmezett HBEFA v2.1[39] kerül használatra, ami az alábbi módon számítja az egyes járművek üvegházhatású gáz kibocsátását:

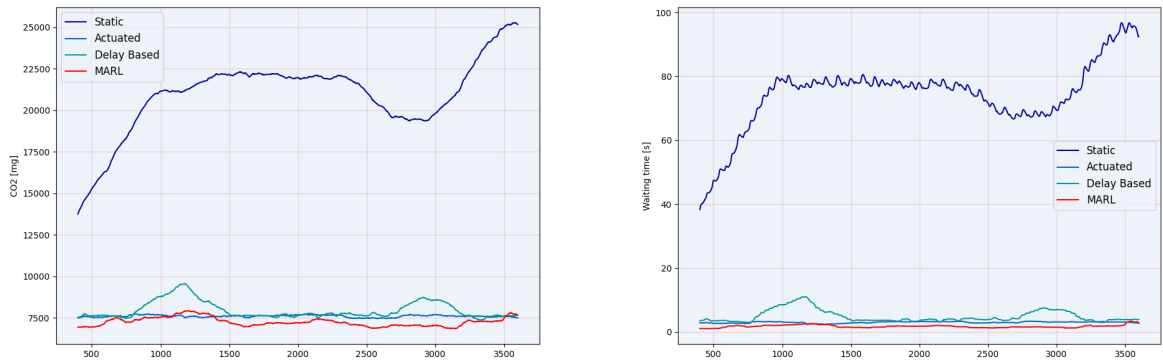
$$E = c_0 + c_1va + c_2va^2 + c_3v + c_4v^2 + c_5v^3,$$

ahol a c_i értékek az egyes metrikákhoz tartozó konstansok, míg a v a sebességet, az a pedig a gyorsulást jelöli. Mivel minden emisszió az alábbi modell segítségével kerül meghatározásra a szimulátorban, így a trend is hasonló jelleggel fog változni. Ennek köszönhetően kettő fontosabb mutató tekintetében kerülnek ismertetésre az eredmények, a CO_2 és NO_x vizsgálatában nyilvánul meg az emissziók összehasonlítása.

4.2. Algoritmusok összehasonlítása

A teljes körű összehasonlítás érdekében nem elegendő az előállt modellt csupán az ITS mentes környezettel összehasonlítani, így viszonyítási alapként két másik algoritmus is a kiértékelés részét képezi. Ezek a SUMO-ban beépített implementációk, melyek az *adaptive* és *delay based* nevekre hallgatnak. Míg az előbbi a kialakult sorok minimalizálásával valósítja meg a hatékonyabb áramlást, utóbbi a várakozási idők minimális értéken tartásával igyekszik optimális döntéseket hozni.

Az összehasonlító elemzés során az út terhelése a teljesítőképességének határát meghaladó értékere lett beállítva, hogy olyan tesztkörülmények álljanak elő, ahol szükség van az algoritmusok használatára. A forgalom eloszlása azonban nem a tanítás során



4.1. ábra. Eredmények 1 kereszteződés esetére

4.1. táblázat. 30 perces időablakban elvégzett teszt 1 kereszteződés esetére

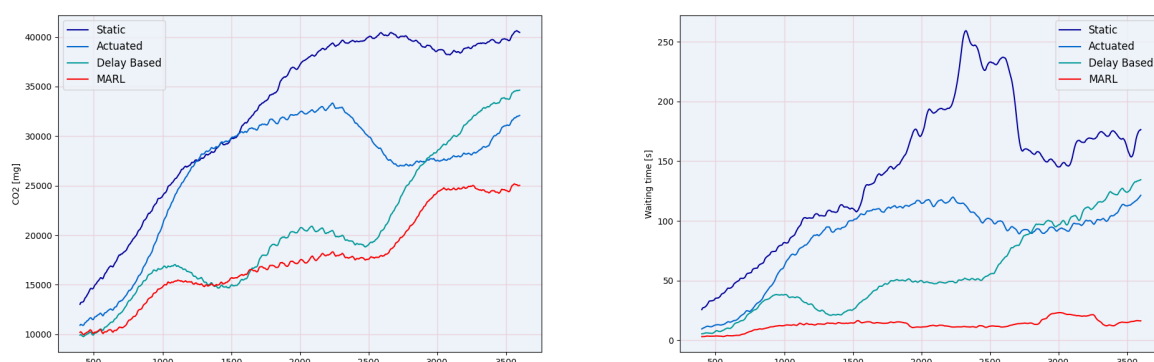
	Waiting Time	Average Speed	CO2	NOx	Halting Vehicles
Static	67.206	4.1012	19480.115	8.66188	7406.75
Actuated	2.932	7.4978	7596.198	3.20075	1046.0
Delay Based	5.127	6.9916	8013.949	3.39340	1320.0
MARL	1.717	7.2027	7290.073	3.06179	803.25

4.2. táblázat. 60 perces időablakban elvégzett teszt 1 kereszteződés esetére

	Waiting Time	Average Speed	CO2	NOx	Halting Vehicles
Static	72.856	3.9614	20694.588	9.21786	15991.5
Actuated	2.989	7.5899	7591.442	3.19872	2081.5
Delay Based	4.791	6.9903	7923.337	3.35206	2574.75
MARL	1.828	7.1689	7262.742	3.049596	1626.75

használt értékkel megegyezően került az útszakaszokra, ezzel is távolodva attól, hogy egy overfit legyen a feladat megoldása.

Ahogy az ábrákon látható, az ITS mentes környezetben egyértelműen kialakul a shock wave, de a hálózat méretétől függően a többi módszer tekintetében is megfigyelhető, bár kisebb mértékben. A MARL alapú megoldás nem csak a csúcserkékek tekintetében mutat csökkenést, hanem az ingadozások is kisebb mértékűek, ezzel egy kevésbé hátráltatott forgalom képét visszaadva a diagramokon. Ez tulajdonképpen a vizuális igazolása a szimulációk grafikus megjelenítése mellett, hogy a shock wave jelenség megfelelő prediktív algoritmus segítségével adott terhelés esetén teljesen meg-



4.2. ábra. Eredmények 2 kereszteződés esetére

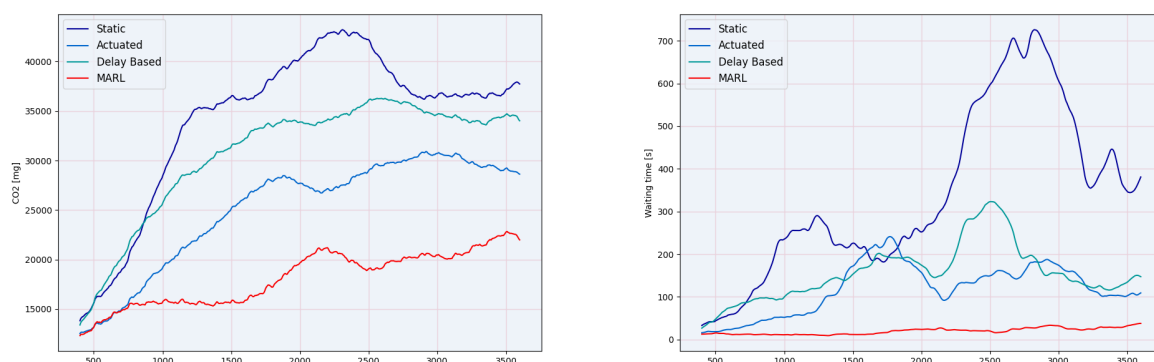
4.3. táblázat. 30 perces időablakban elvégzett teszt 2 kereszteződés esetére

	Waiting Time	Average Speed	CO2	NOx	Halting Vehicles
Static	86.028	5.1520	24209.949	10.67928	8876.5
Actuated	63.670	5.4120	22015.251	9.66317	7326.125
Delay Based	27.274	6.6790	14743.874	6.31286	3723.75
MARL	10.442	7.2813	13744.383	5.85558	2585.875

4.4. táblázat. 60 perces időablakban elvégzett teszt 2 kereszteződés esetére

	Waiting Time	Average Speed	CO2	NOx	Halting Vehicles
Static	137.594	4.7504	31886.470	14.21745	25823.75
Actuated	84.979	5.0486	26022.961	11.50417	18866.75
Delay Based	59.250	5.7190	20740.818	9.07874	13926.375
MARL	12.601	6.8569	17664.559	7.65427	6714.75

előzhető egy többbeavatkozás forgalmi hálózaton. Egy ponton az ITS mentes rendszerekben figyelhető visszaesés, viszont ez csupán annak köszönhető, hogy a hálózatra nem tudott több jármű belépni, így kevesebb jármű szolgáltatott adatot. A táblázatokról leolvasható, hogy 4 különböző forgalmi hálózaton 2 egymástól eltérő vizsgált időszak esetében is sikerült a legkisebb CO_2 és NO_x kibocsájtást elérni, ezzel teljesítve az emissziós elvárásokat. Emellett a várakozási időket és a feltartóztatott járműveket tekintve is hasonló trend mutatkozik meg az adatokban. A sebességet tekintve is, ahol érvényesül az identical payoff, a MARL alapú megoldás alkalmasabbnak bizonyul a többihez képest. A baseline algoritmusokat tekintve megfigyelhető, hogy bár a kötött



4.3. ábra. Eredmények 3 kereszteződés esetére

4.5. táblázat. 30 perces időablakban elvégzett teszt 3 kereszteződés esetére

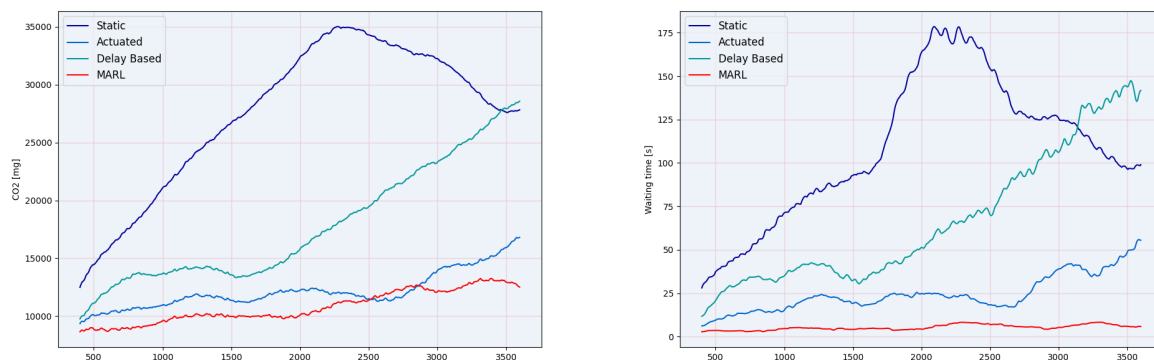
	Waiting Time	Average Speed	CO2	NOx	Halting Vehicles
Static	163.230	5.3928	27883.710	12.41850	11905.583
Actuated	96.199	6.3263	20162.268	8.84889	7260.666
Delay Based	114.739	5.7035	25289.744	11.21376	10104.166
MARL	14.065	7.5449	15173.915	6.53076	2960.583

4.6. táblázat. 60 perces időablakban elvégzett teszt 3 kereszteződés esetére

	Waiting Time	Average Speed	CO2	NOx	Halting Vehicles
Static	323.762	4.7415	33404.804	14.96324	30599.5
Actuated	114.715	5.8889	24615.000	10.89000	18852.25
Delay Based	149.480	5.1378	29977.941	13.36289	24615.083
MARL	20.282	7.1369	17934.456	7.79766	7765.5

fázisidejű kapcsoláshoz képest jobb eredményt érnek el, nem közelítik meg a MARL alapú irányítást, valamint kisebb oszcillációk is megfigyelhetők helyenként.

Összefoglalva az eredményeket, a kiértékelés a várt eredményt mutatja, minden algoritmus az egyes metrikákat tekintve alkalmasabbnak bizonyult annál a rendszernél, amiben nem került sor ITS használatára, és ezek közül is a MARL alapú rendszer érte el a legjobb eredményt, köszönhetően a kognitív döntéshozásra való képességének.



4.4. ábra. Eredmények 4 kereszteződés esetére

4.7. táblázat. 30 perces időablakban elvégzett teszt 4 kereszteződés esetére

	Waiting Time	Average Speed	CO2	NOx	Halting Vehicles
Static	75.367	6.0456	21628.051	9.50294	7924.75
Actuated	16.962	7.7325	10934.342	4.57993	2286.875
Delay Based	32.135	7.2594	12978.820	5.51974	3435.8125
MARL	3.888	8.5647	9487.481	3.90885	1188.8125

4.8. táblázat. 60 perces időablakban elvégzett teszt 4 kereszteződés esetére

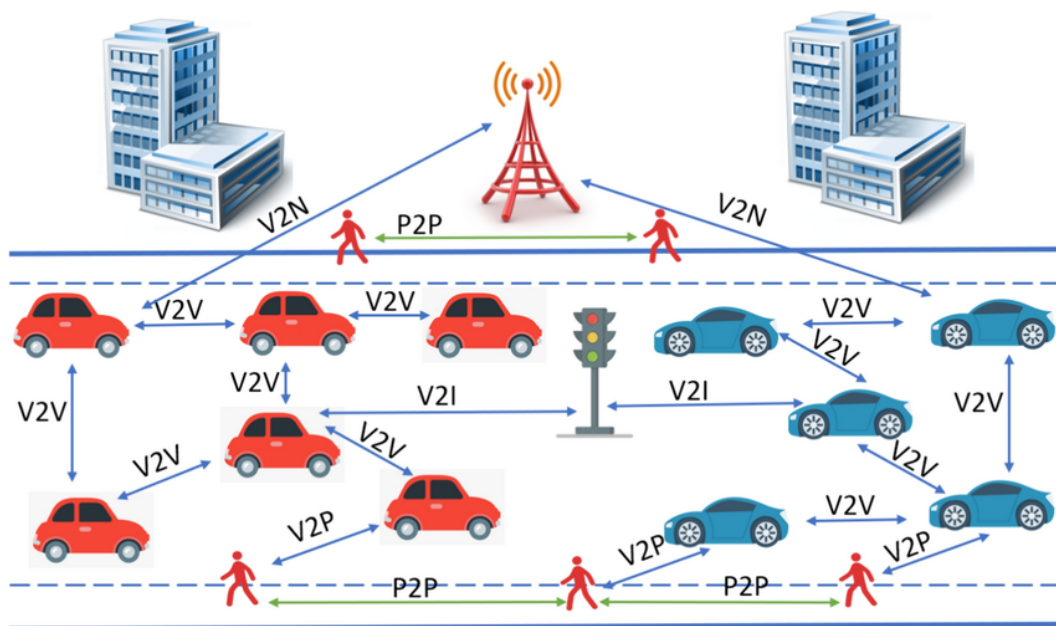
	Waiting Time	Average Speed	CO2	NOx	Halting Vehicles
Static	103.552	5.6438	26662.658	11.81623	21105.1875
Actuated	25.294	7.3910	12265.355	5.187802	5929.75
Delay Based	68.154	6.3610	17919.312	7.78420	12042.75
MARL	5.200	8.1890	10737.716	4.47625	3080.0625

5. fejezet

Konklúzió

A dolgozatban bemutatott metodológia egy olyan MARL rendszert tárgyal, amely nemcsak az ITS nélküli megközelítésekhez viszonyítva mutat előrelépést, hanem a SUMO szimulációs platform beépített algoritmusainak teljesítményét is felülmúlja. Ez a javulás nemcsak az emissziók csökkenésében mutatkozik meg, hanem a megnövekedett átlagsebesség révén a hálózat nagyobb áteresztőképességét is eredményezi. A homogén ágensek révén megvalósuló újraszakálázhatóság jelentős előrelépést jelent az általános felhasználhatóság irányába, lehetővé téve az univerzális bevethetőséget, ami jelentős mértékben csökkenti a tanítási erőforrások igényét. A dolgozatban bemutatott csúszó ablakos módszer egy innovatív megközelítést képvisel, amely a megfelelően definiált RL absztrakciókkal kombinálva biztosítja az alkalmazhatóságot tetszőleges méretű hálózaton.

A bemutatott előrelépés számos új kutatási irányt nyit meg, amelyek további hatékonyságnövelést eredményezhetnek a jövőbeli fejlesztések során. Első lépésként, a forgalmi hálózatok területén maradván, a heterogén ágensek vizsgálata, valamint azok számának és eloszlásának elemzése további előrelépést jelenthet a módszer valós hálózatokon történő alkalmazhatósága felé. Ez különösen fontos, mivel kevés város rendelkezik homogén keresztezésekkel, ahol minden csomópont azonos architektúrájú. Egy másik izgalmas kutatási irány lehet a bemenő sávok számának és a kanyarodási szabályok változtatásának vizsgálata, ami szintén hozzájárulhat a rendszer további finomításához és alkalmazhatóságához. Bár a kutatások eddig is foglalkoztak a VSLC hasonló alkalmazásával, az anomáliák véletlenszerű fellépése, több sávlezárás egyidejű kezelése egy hálózaton belül, valamint a különböző szegmenshosszal rendelkező rendszerek irányítása még kutatásra váró területek maradtak. A forgalomirányításban további kihívást jelent a mikroszkopikus és makroszkopikus irányítás hatékony összekapcsolása a V2X kommunikáció révén. Ennek során a forgalmi hálózathoz esetlegesen kiegészítő keresztezések információi közvetíthetők lennének a járművek számára egy központi



5.1. ábra. V2X kommunikáció szemléltetése, ahol az egyes résztvevők pontosabb képet kapva a hálózatról képesek hatékonyabb közlekedést megvalósítani[42]

kommunikációs csatornán keresztül. Ez lehetővé tenné a globális útvonal dinamikus újratervezését, biztosítva az optimális forgalmi áramlást még hálózati zavarok esetén is.

Szintén ígéretes megközelítés lehet a megerősítéses tanulás egyéb eredményeinek bevonása a fejlesztésekbe, amelyek gyorsabb és hatékonyabb konvergenciát eredményeznek. Ilyen módszer például a tanítóminták priorizálásának alkalmazása[40][38], amely számos esetben bizonyította a minták hatékony visszajátszásának fontosságát buffert használó algoritmusok esetén. A priorizált mintavisszajátszás lehetővé teszi, hogy az ügynök a tanulási szempontból legfontosabb, legrelevánsabb tapasztalatokra összpontosítson, így javítva a tanulás sebességét és hatékonyságát. Egy másik potenciális irány a reward függvény előzetes kiértékelése a MCTS segítségével[19][41], aminek segítségével adott függvények közül megállapítható a tanításra legalkalmasabb.

Az elért eredmények alapján jól látható, hogy még számos terület kínálkozik lehetőség a fenntarthatóbb városok kialakítására. A felsorolt módszerek együttes alkalmazása további jelentős előrelépéseket biztosíthat, amelyek hozzájárulnak a célként kitűzött élhetőbb nagyvárosok megvalósításához.

Rövidítések

ITS	-	Intelligent Transportation Systems
TSC	-	Traffic Signal Control
VSLC	-	Variable Speed Limit Control
SCATS	-	Sydney Coordinated Adaptive Traffic System
RHODES	-	Real- time Hierarchical Optimized Distributed Effective System
DL	-	Deep Learning
RL	-	Reinforcement Learning
MARL	-	Multi Agent Reinforcement Learning
AI	-	Artificial Intelligence
ML	-	Machine Learning
SL	-	Supervised Learning
USL	-	Unsupervised Learning
ILSVRC	-	ImageNet Large Scale Visual Recognition Challenge
SIFT	-	Scale Invariant Feature Transform
HOG	-	Histogram of oriented gradients
SVM	-	Support Vector Machine
DQN	-	Deep Q-Network
DRL	-	Deep Reinforcement Learning
NN	-	Neural Network
SUMO	-	Simulation of Urban MObility
V2X	-	Vehicle-to-everything

Ábrák jegyzéke

1.1. Kibocsátás mértéke szektoronként csoportosítva. [4].	1
1.2. Várakozási idők szemléltetése[5].	2
1.3. Fundamentális diagram[8]	3
1.4. Korszerű DL modellek tanítási költsége[17]	4
2.1. A gépi tanulás[22]	6
2.2. A gépi tanulás fajtái[24]	7
2.3. Supervised Learning[28]	8
2.4. Unsupervised Learning[28]	8
2.5. Megerősítéses tanulás[29]	9
2.6. Az ϵ -greedy policy működése[32]	11
2.7. A tanítás konvergenciagörbéje.	12
2.8. A Multi-ágens feladat problémáspecifikus implementációja	13
3.1. A TraCI működése[36]	15
3.2. A SUMO kereszteződés felépítése	16
3.3. A Gym interfész[38]	17
3.4. A környezet felépítésének szemléltetése	19
4.1. Eredmények 1 kereszteződés esetére	21
4.2. Eredmények 2 kereszteződés esetére	22
4.3. Eredmények 3 kereszteződés esetére	23
4.4. Eredmények 4 kereszteződés esetére	24
5.1. V2X kommunikáció szemléltetése, ahol az egyes résztvevők pontosabb képet kapva a hálózatról képesek hatékonyabb közlekedést megvalósítani[42]	26

Táblázatok jegyzéke

4.1.	30 perces időablakban elvégzett teszt 1 kereszteződés esetére	21
4.2.	60 perces időablakban elvégzett teszt 1 kereszteződés esetére	21
4.3.	30 perces időablakban elvégzett teszt 2 kereszteződés esetére	22
4.4.	60 perces időablakban elvégzett teszt 2 kereszteződés esetére	22
4.5.	30 perces időablakban elvégzett teszt 3 kereszteződés esetére	23
4.6.	60 perces időablakban elvégzett teszt 3 kereszteződés esetére	23
4.7.	30 perces időablakban elvégzett teszt 4 kereszteződés esetére	24
4.8.	60 perces időablakban elvégzett teszt 4 kereszteződés esetére	24

Irodalomjegyzék

- [1] Jes Fenger. Urban air quality. *Atmospheric environment*, 33(29):4877–4900, 1999.
- [2] Hannah Ritchie. Tracking global data on electric vehicles. *Our World in Data*, 2024. <https://ourworldindata.org/electric-car-sales>.
- [3] Arne Kesting and Martin Treiber. How reaction time, update time, and adaptation time influence the stability of traffic flow. *Computer-Aided Civil and Infrastructure Engineering*, 23(2):125–137, 2008.
- [4] Hannah Ritchie. Sector by sector: where do global greenhouse gas emissions come from? *Our World in Data*, 2020. <https://ourworldindata.org/ghg-emissions-by-sector>.
- [5] Aoxiang Wu, Liqun Qi, and Xiaoguang Yang. Mechanism analysis and optimization of signalized intersection coordinated control under oversaturated status. *Procedia - Social and Behavioral Sciences*, 96:1433–1442, 11 2013.
- [6] Junping Zhang, Fei-Yue Wang, Kunfeng Wang, Wei-Hua Lin, Xin Xu, and Cheng Chen. Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1624–1639, 2011.
- [7] Kashif Naseer Qureshi and Abdul Hanan Abdullah. A survey on intelligent transportation systems. *Middle-East Journal of Scientific Research*, 15(5):629–642, 2013.
- [8] András Hegyi. *Model predictive control for integrating traffic control measures*. Netherlands TRAIL Research School, 2004.
- [9] Salah Bouktif, Abderraouf Cheniki, Ali Ouni, and Hesham El-Sayed. Deep reinforcement learning for traffic signal control with consistent state and reward design approach. *Knowledge-Based Systems*, 267:110440, 2023.

- [10] 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.
- [11] Bidoura Khondaker and Lina Kattan. Variable speed limit: an overview. *Transportation Letters*, 7(5):264–278, 2015.
- [12] Md Hadiuzzaman, Tony Z Qiu, and Xiao-Yun Lu. Variable speed limit control design for relieving congestion caused by active bottlenecks. *Journal of Transportation Engineering*, 139(4):358–370, 2013.
- [13] Jaja Kustija et al. Scats (sydney coordinated adaptive traffic system) as a solution to overcome traffic congestion in big cities. *International Journal of Research and Applied Technology (INJURATECH)*, 3(1):1–14, 2023.
- [14] Pitu Mirchandani and Fei-Yue Wang. Rhodes to intelligent transportation systems. *IEEE Intelligent Systems*, 20(1):10–15, 2005.
- [15] KJ Prabuchandran, Hemanth Kumar AN, and Shalabh Bhatnagar. Multi-agent reinforcement learning for traffic signal control. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 2529–2534. IEEE, 2014.
- [16] Xiaoqiang Wang, Liangjun Ke, Zhimin Qiao, and Xinghua Chai. Large-scale traffic signal control using a novel multiagent reinforcement learning. *IEEE transactions on cybernetics*, 51(1):174–187, 2020.
- [17] Ben Cottier. Trends in the dollar training cost of machine learning systems, 2023. Accessed: 2023-12-18.
- [18] Csanád Levente Balogh, Bálint Pelenczei, Bálint Kővári, and Tamás Bécsi. Strategic data navigation: Information value-based sample selection. 2024.
- [19] Bálint Kővári, Bálint Pelenczei, István Gellért Knáb, and Tamás Bécsi. Beyond trial and error: Lane keeping with monte carlo tree search-driven optimization of reinforcement learning. *Electronics*, 13(11):2058, 2024.
- [20] P Russel Norvig and S Artificial Intelligence. A modern approach. *Prentice Hall Upper Saddle River, NJ, USA: Rani, M., Nayak, R., & Vyas, OP (2015). An ontology-based adaptive personalized e-learning system, assisted by software agents on cloud storage. Knowledge-Based Systems*, 90:33–48, 2002.

- [21] Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023.
- [22] Iqbal Sarker. Ai-based modeling: Techniques, applications and research issues towards automation, intelligent and smart systems. *SN Computer Science*, 3, 02 2022.
- [23] 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.
- [24] Junjie Peng, Elizabeth Jury, Pierre Dönnes, 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.
- [25] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018(1):7068349, 2018.
- [26] Jenny Raggett and William Bains. *Mesterséges intelligencia, AZ*. Akadémiai Kiadó, 1994.
- [27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [28] R Raj. Supervised, unsupervised, and semi-supervised learning with real-life usecase. *Enjoy Algorithm*, 24 May, 2021.
- [29] Richard S Sutton, Andrew G Barto, et al. Reinforcement learning. *Journal of Cognitive Neuroscience*, 11(1):126–134, 1999.
- [30] Ayush Singh. Reinforcement learning: Markov decision process (part 1). *Towards data science*, 2018.
- [31] 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.

- [32] Iman Sajedian, Heon Lee, and Junsuk Rho. Double-deep q-learning to increase the efficiency of metasurface holograms. *Scientific Reports*, 9:10899, 07 2019.
- [33] Afshin Oroojlooy and Davood Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence*, 53(11):13677–13722, 2023.
- [34] Dipti Srinivasan. *Innovations in Multi-Agent Systems and Application-1*, volume 310. Springer, 2010.
- [35] 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.
- [36] Cristina Olaverri Monreal, Javier Errea-Moreno, Alberto Díaz, Carlos Biurrun Quel, Luis Serrano, and Markus Kuba. Connection of the sumo microscopic traffic simulator and the unity 3d game engine to evaluate v2x communication-based systems. *Sensors*, 18, 12 2018.
- [37] 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.
- [38] Bálint Kővári, Bálint Pelenczei, and Tamás Bécsi. Enhanced experience prioritization: A novel upper confidence bound approach. *IEEE Access*, 2023.
- [39] 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.
- [40] Tom Schaul. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.
- [41] Bálint Kővári, 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.
- [42] Abhinav Singh and Bikash Chandra Sahana. Full duplex-non-orthogonal multiple access for v2x communications in 5g millimeter wave. *Wireless Personal Communications*, 136:1–24, 06 2024.