

TUDOMÁNYOS DIÁKKÖRI KONFERENCIA

Tanítóminta-priorizálási módszerek vizsgálata és fejlesztése Megerősítő Tanulás esetére

Szerző:

Pelenczei Bálint



Budapesti Műszaki és Gazdaságtudományi Egyetem
Közlekedés- és Járműirányítási Tanszék



HUN-REN Számítástechnikai és Automatizálási Kutatóintézet

Konzulens:

Kővári Bálint

2023

ÚNKP-23-1-I-BME-19:

A Kulturális és Innovációs Minisztérium **ÚNKP-23-1-I-BME-19** kódszámú Új Nemzeti Kiválóság Programjának a Nemzeti Kutatási, Fejlesztési és Innovációs Alapból finanszírozott szakmai támogatásával készült.



KULTURÁLIS ÉS INNOVÁCIÓS
MINISZTERIUM



NEMZETI KUTATÁSI, FEJLESZTÉSI
ÉS INNOVÁCIÓS HIVATAL



Tartalomjegyzék

1. Bevezetés	1
1.1. Motiváció	1
1.2. Kapcsolódó kutatások, eredmények	3
1.3. Kontribúció	7
2. Algoritmusok – Gépi Tanulás	9
2.1. Felügyelt Tanulás	10
2.2. Felügyelet Nélküli Tanulás	11
2.3. Megerősítéses Tanulás	11
2.3.1. Policy-alapú módszerek	14
2.3.2. Modell-alapú módszerek	15
2.3.3. Érték-alapú módszerek	15
2.3.3.1. Q -tanulás	16
2.3.3.1.1. Deep Q -Network	16
2.3.4. Tanítóminta priorizálás	18
2.3.4.1. Priorizált Tanítóminta Visszajátszás	19
2.3.4.2. Upper Confidence Bound-alapú tanítóminta priorizálás . .	20
3. Környezet	23
3.1. Gym környezetek	24
3.1.1. Cartpole	25
3.1.2. Acrobot	26
3.1.3. Cliff Walking	26
3.1.4. Taxi	27
3.2. Tanítás	27
3.3. Kiértékelés	28
4. Eredmények	30
4.1. Első fázis – általános kiértékelés	30
4.2. Második fázis – statikus visszajátszási memória	33
5. Konklúzió	36

Kivonat

A közlekedés- és járműirányítás kutatásai az utóbbi időben egyértelműen rávilágítottak a Mesterséges Intelligencia (MI) alkalmazásának számos előnyére. Ez elsősorban annak köszönhető, hogy a felmerülő optimalizálási és irányítási problémák komplexitása nem mindig teszi lehetővé hagyományos módszerek használatát, vagy azok teljesítménye elmarad a Mesterséges Intelligencia-alapú megoldásokétól. Az elérhető számítási kapacitás rohamos növekedésének következtében sok kutatás fókuszába a Megerősítéses Tanulás (Reinforcement Learning – RL) került, amely kimagasló eredményeket képes nyújtani különféle szekvenciális döntéshozatali problémákra.

A Megerősítéses Tanulás módszere azonban rendkívül nagy idő- és számítási erőforrásigényt jelent a tanítási folyamatok során, amikor az ágensnek nagy számosságú tanítómintát szükséges feldolgoznia a végső teljesítmény kialakításához. A tapasztalatok visszajátaszása lehetővé teszi ugyan a múltbeli minták újrafelhasználását, azonban az állapotátmenetek mintavételezése nem minden esetben hatékony, mivel egyenletes eloszlást alkalmazva az egyes tanítóminták azonos gyakorisággal játszódnak vissza, függetlenül azok jelentőségétől. A tanítóminták megfelelő módszerrel való priorizálása által elérhetővé válhat a tanítás konvergenciájának felgyorsítása és stabilizálása, ami végül a tanítás erőforrásigényének drasztikus csökkenéséhez vezethet.

Ebben a kutatásban célunk egy olyan tanítóminta-priorizálási módszer kidolgozása volt, amely a vizsgált szakirodalomban található teljesítményét felülmúlja. A módszert több környezetben kiértékeltek, és összehasonlítottuk más megközelítésekkel. Eddigi eredményeink alapján kijelenthető, hogy az általunk javasolt módszer jelentős mértékben javítja a tanítás hatékonyságát, lehetővé téve gyorsabb és költséghatékonyabb tanulási folyamatokat az összehasonlításként választott módszerekhez képest.

Kulcsszavak

Gépi Tanulás, Megerősítéses Tanulás, Mély Tanulás, mintavételezés, Q tanulás, tanítóminta priorizálás, tanítóminta visszajátaszás

Rövidítések

AI	-	Artificial Intelligence (Mesterséges Intelligencia)
DL	-	Deep Learning (Mély Tanulás)
DNN	-	Deep Neural Network (Mély Neurális Hálózat)
DQN	-	Deep Q -Network (Mély Q -Hálózat)
ER(M)	-	Experience Replay (Memory) (Tanítóminta Visszajátszás(i Memória))
MDP	-	Markov Decision Process (Markov Döntési Folyamat)
ML	-	Machine Learning (Gépi Tanulás)
NN	-	Neural Network (Neurális Hálózat)
PER	-	Prioritized Experience Replay (Priorizált Tanítóminta Visszajátszás)
RL	-	Reinforcement Learning (Megerősítéses Tanulás)
SL	-	Supervised Learning (Felügyelt Tanulás)
TD	-	Temporal Difference
UCB	-	Upper Confidence Bound (Felső Konfidencia Határ)
USL	-	UnSupervised Learning (Felügyelet Nélküli Tanulás)

1. fejezet

Bevezetés

Az utóbbi időben számos tudományos kutatás mutatott rá a Mesterséges Intelligencia-alapú megoldások alkalmazása által nyújtott megannyi előnyre, amelyek a legkülönbözőbb területeken jelentkeznek, mint például a robotika [1], a gyártás optimalizálás [2], az előrejelzések [3], az erőforrás-gazdálkodás [4] és végül az autonóm járműirányítás [5]. Elsődlegesen az újonnan felmerülő, valós idejű alkalmazást igénylő, komplex optimalizálási és irányítási problémák esetében, a hagyományos megközelítések, klasszikus algoritmusok alkalmazása gyakran korlátozott vagy akár egyáltalán nem megvalósítható az ezekkel járó óriási számítási kapacitásigény miatt, ami az online beavatkozási időben jelentkezik.

Ellentétben ezzel, a Deep Learning-alapú módszerek esetében egy előre betanított Neurális Hálózat döntéshozatali ideje milliszekundumos nagyságrendbe esik, biztosítva ezzel a valós idejű alkalmazhatóságot. Azonban ezeknek a módszereknek is van egy jelentős hátrányuk, amely nem más, mint hogy az offline tanítás során jelentős idő- és számítási teljesítmény szükséglet realizálódik.

1.1. Motiváció

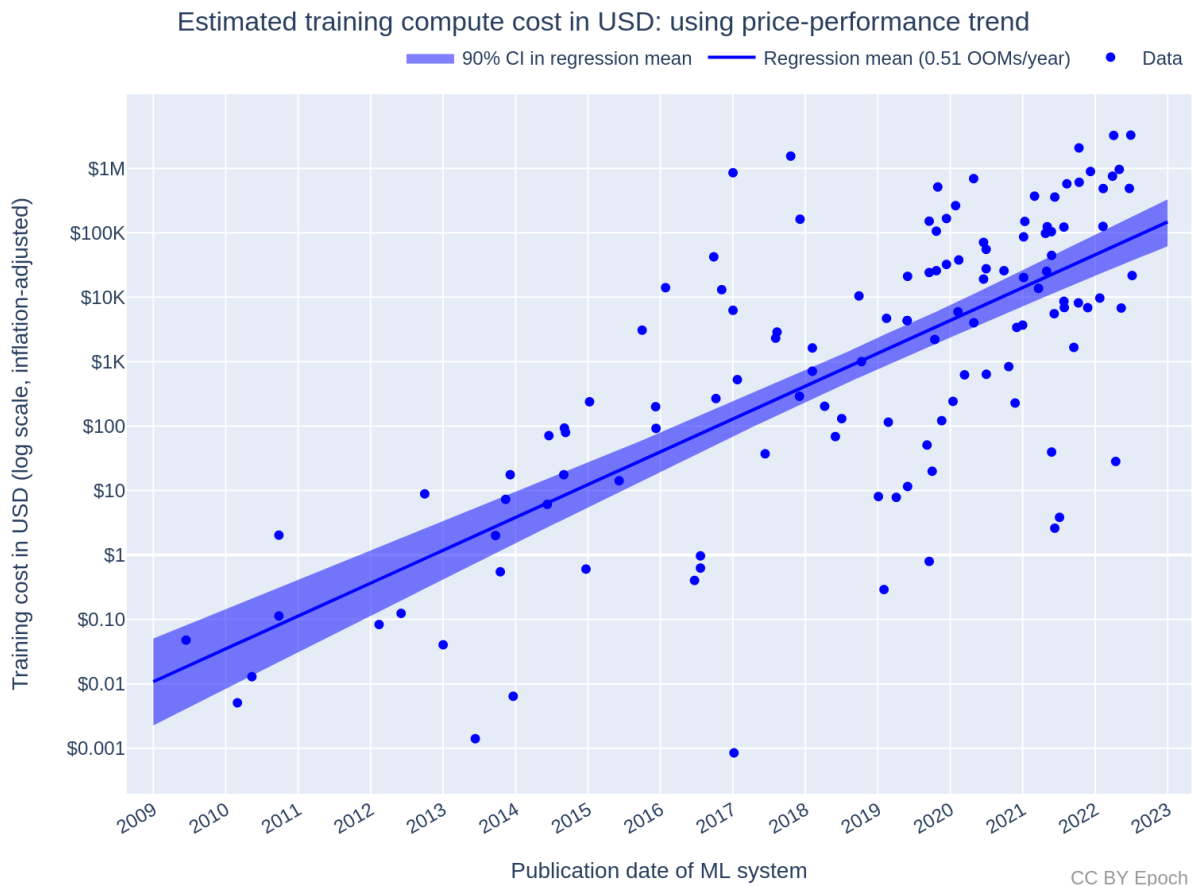
Annak érdekében, hogy az előzőekben bemutatott centrális problémát részletesebb analízisnek vessük alá, illetve hogy kiemeljük annak jelentőségét, érdemes megvizsgálni az abból fakadó további negatív hatásokat és a közöttük lévő ok-okozati összefüggéseket.

Először is, a [6]-ben elemzett adatbázis alapján egy átlagos korszerű Deep Learning modell tanításának számítási költsége, amelynél rendelkezésre állnak részletes tanítási adatok, körülbelül $2,44 \cdot 10^{24}$ FLOP. Azonban még a legerősebb AI szuperszámítógép platformon (amely jelenleg az NVIDIA HGX 16x NVIDIA A100 GPU modullal) is közel egy évbe telik egy ilyen átlagos state-of-the-art modell tanítása, amely 10 petaFLOPS számítási teljesítményre képes. Mivel a Mély Tanulás területén a kutatások és fejlesztések hatékonyságát leginkább a tanítási idő akadályozza – megnehezítve ezzel különböző architektúrák, hiperparaméter optimalizációs módszerek és adat-augmentációs technikák felfedezését –, egy kulcsfontosságú korlátnak számít.

Másodszor pedig fontos megérteni, hogy a Mély Neurális Hálózatok tanításával összefüggő valós költségek nem fejezhetők ki csupán a tanítás során felmerülő számítási szükséglettel, ezért általában jóval mélyebb, gazdasági megfontolások figyelembe vétele szükséges. Az érintett költségek pontos megértése és kvantifikálása létfontosságú mind a kutatók, mind a gyakorlati szakemberek, mind pedig a döntéshozó felek számára. Ennek célja, hogy képesek legyenek megalapozott döntéseket hozni és így optimalizálni a különböző Mély Tanulással összefüggő projektek erőforrás-elosztását.

Az 1.1 ábrán x tengelyen az idő látható 2009-től napjainkig (2023), y tengelyen pedig az adott időszakban state-of-the-art technológiának számító Deep Learning modellek inflációkorrigált tanítási költségei vannak megjelenítve logaritmikus skálán, *USD*-ben kifejezve. A diagram jól mutatja, hogy az elmúlt évek tendenciái a tanítási költségek erőteljes növekedésére mutatnak rá, amely miatt a Mély Neurális Hálózatok tanításáért felelős algoritmusok hatékonysága hosszútávon egyre kritikusabbá válik. Ezen okból kifolyólag a Mély Tanulási modellek tanítását végző algoritmusok hatékonyságának optimalizálását tekintettük ezen kutatás fő céljának.

Megkérdőjelezhetetlen tény, hogy az említett tanítási módszerek szuboptimális műkö-



1.1. ábra. State-of-the-art Mély Tanulási modellek inflációkorrigált tanítási költségei *USD*-ben kifejezve [7]

dése egyértelműen a tanító adathalmazok kezeléséből eredeztethető, a Neurális Hálózatok súlyainak frissítése során. Az első megoldások ezen jelenség kezelésére a neurobiológia egyes elméleteinek analógiájára épültek: tapasztalat visszajátszási módszerek bevezetésével próbálták elősegíteni Neurális Hálózatok memóriaképző és visszahívó képességét azáltal, hogy az eddigi tapasztalatokat felhasználva újraaktiváltak bizonyos neurális mintákat, ahogy az részletezésre kerül [8]-ban. Habár a tapasztalatok visszajátszása a Gépi Tanulás mindhárom fő ágazatában elterjedt, a metodológiának leginkább a Megerősítéses Tanulás területén van relevanciája, ahol az ilyen irányú fejlesztések szignifikánsan képesek javítani az ágens teljesítményét. Ez a performancia növekedés azáltal érhető el, hogy a tanító-minták megfelelő újbóli felhasználásával kiküszöbölhető, vagy legalább nagy mértékben csökkenthető az RL algoritmusok legkritikusabb problémájának, az ineffektív mintavételezésnek a hatása, amely ezen algoritmusok skálázhatóságát és valós életbeli alkalmazhatóságát egyaránt korlátozza.

Az elmúlt időszakban számos kutató igyekezett előrehaladást elérni ezen a területen, olyan koncepciókat kidolgozva, amelyek közös céllal a nem megfelelő mintavételezési hatékonyság dilemmájára keresik a megoldást. A következő fejezet rövid áttekintést nyújt a területen jelenleg is aktívan folyó kutatásokról és az eddig elért eredményekről, különös hangsúlyt fektetve a tanítóminta priorizálási módszerekre.

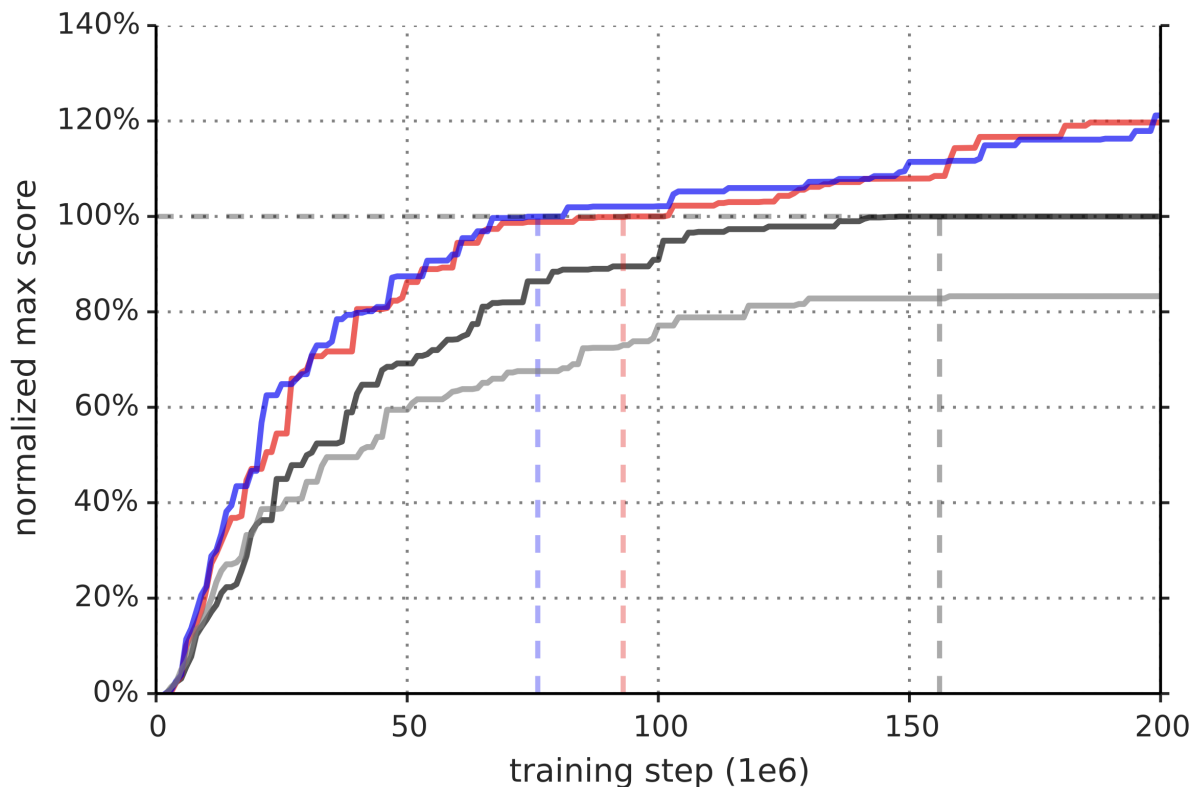
1.2. Kapcsolódó kutatások, eredmények

Nem sokkal a Megerősítéses Tanulás megjelenése után a fontosabb állapotátmenetek priorizálásának szükségessége egyértelművé vált. [9]-ben a kutatók egy úgynevezett priorizált söprési algoritmust mutattak be, amely az állapotátmenetek fontosságának meghatározásához az elnyelési valószínűségek (Δ paraméterek) változását használták fel. A magasabb Δ paraméter értékek az állapotot megelőző állapotok (másképpen az előd állapotok) abszorpciós valószínűségeiben bekövetkező potenciális változásokat jelzik előre, amelyek a Neurális Hálózat frissítése szempontjából így érdekesebbnek bizonyulnak. Az algoritmus két egymást követő valós megfigyelés között az állapotátmeneteket a prioritásuk alapján rangsorolja, amíg a feldolgozás el nem ér egy előre definiált számítási határt. Eredményként a fontosabb állapotátmenetek a prioritási rangsor elejére kerülnek, így végtelen számítási időt feltételezve kialakítva a teljes priorizált listát.

Az állapotátmenetek eltárolásának ötlete mindaddig megvalósíthatatlan maradt, amíg a koncepciót jóval később, [10]-ben nem publikálták. Ebben a publikációban került bemutatásra az úgynevezett tanítóminta-visszajátszási memória, amely lehetővé tette az ágensnek számára múltbeli tapasztalataik visszajátszását és ezáltal újbóli felhasználását azáltal, hogy az állapotátmenetet egyértelműen meghatározó, összetartozó elemeket elmentették. Ezt követően a Neurális Hálózat frissítése során a visszajátszási memória elemein sztochasztikus mintavételezés történik, egyenletes eloszlás alkalmazásával.

Azonban, ahogy az az egyenletes eloszlás alkalmazásából várható, ez a módszer minden elmentett tanítómintát ekvivalens fontosságúként kezel, amely erős megszorítást jelent az optimális megoldás kifejlesztése szempontjából. A fenti elméletek egyesítése, illetve az abból készült tanulmány került bemutatásra [11]-ben, ahol egy új algoritmust, a Priorizált Tanítóminta Visszajátszást demonstrálták a szerzők. Ennek a módszernek is azonos a lényege, hogy tanítás során a különböző fontosságú tapasztalatokat rangsorolt listába szervezze. Ebben az új megközelítésben minden egyes állapotátmenethez úgynevezett prioritás értéket rendelünk, ami alapján kialakításra kerül egy valószínűségi eloszlás az összes, memóriában aktuálisan eltárolt tapasztalatra vonatkozóan. Empirikus eredmények alapján sikerült kimutatni a tanítás hatékonyságának növekedését, amint azt az 1.2 ábra is szemlélteti: fekete színnel van jelölve az egyenletes eloszlás alapú mintavételezést alkalmazó ágens tanítási görbéje, míg a cikkben bemutatott két különböző priorizálási technikát felhasználó ágensek görbéi kék, illetve piros színnel jelennek meg.

Ahogy a PER algoritmus egyre inkább elterjedt a kutatók körében, hamarosan az RL területének vezető tanítóminta priorizálási módszerévé vált, ezáltal számos – később bemutatásra kerülő – technika alapjául szolgálva. Ugyanakkor fontos megjegyezni, hogy ezek a módszerek még mindig nem tudták teljesen kiváltani a PER-t, amely továbbra is a szakirodalom által a leggyakrabban alkalmazott algoritmus maradt ezen a területen, így a jelenlegi kutatás szempontjából is a legkorszerűbb és leghatékonyabb megközelítésnek

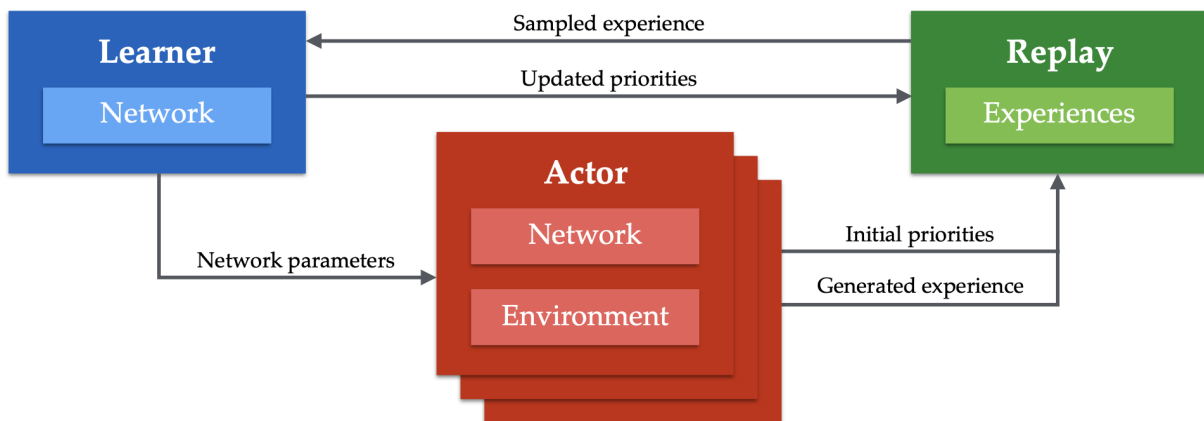


1.2. ábra. Priorizált tanítóminta visszajátszási algoritmus teljesítményét bemutató tanítási görbék [11]

tekinthető. Mivel ezen technológia a jelen tanulmányban is későbbi összehasonlítások alapját képezi, a módszertanról részletesebb ismertetőért lásd a 2.3.4.1 fejezetet.

A következőkben néhány PER metodológiára épülő algoritmus kerül ismertetésre: [12] szerzői a Deep Deterministic Policy Gradient ágenshez mutatták be a PER módszer adaptációját, [13]-ban pedig az Ape-X elnevezésű módszer került demonstrálásra, amely lényegében a PER kiterjesztése az elosztott ágens koncepciójának alkalmazásával. Az Ape-X keretrendszerben, az 1.3 ábrán látható módon, az ágens szerepe több különböző entitás között van megosztva, ezek között több beavatkozást végző és egyetlen tanulást realizáló komponens jelenik meg. Minden beavatkozónak saját, egymástól független környezete van, ugyanakkor a visszajátszási memória közös az összes entitásra nézve. A beavatkozók feladata elsődlegesen tanítóadatok generálása, míg a tanuló a hálózat frissítéséért és a prioritásértékek hozzárendeléséért felelős az egyes, memóriában található tapasztalatokhoz. [14] szerzői bemutatják az úgynevezett Prioritized Sequence Experience Replay kidolgozását, amely szintén tekinthető az eredeti PER koncepció kibővítésének. A szerzők teoretikusan alátámasztották, hogy az ilyen módon képzett tanítóminta szekvenciák prioritizálásával garantáltan gyorsabb konvergencia érhető el, mint a hagyományos PER esetében.

Számos alternatív megközelítés is létezik a szakirodalomban a mintavételezési hatékonyság növelésének céljából, ilyenek például a [15] cikkben szereplő speciális actor-critic algoritmusok is, amelyek a TRACER és az eNACER nevet viselik. A TRACER esetében egy már létező algoritmusnak, az Advantage Actor-Critic-nek a kiegészítéséről van szó, illetve fúziójáról egy tanítóminta visszajátszást alkalmazó úgynevezett Trust Region Policy Optimization algoritmussal. Az eNACER pedig a Natural Actor-Critic módszer speciális, módosított változata, amely esetében a gradiensek becsléséhez a legkisebb négyzetek módszere van alkalmazva, ezzel elkerülve a Fisher mátrix-ból származó explicit számításokat. Mindkét módszer esetében megállapították, hogy a mintavételezési hatékonyság jelentősen megnövekedett az összehasonlítás alapjául szolgáló más Megerősítéses



1.3. ábra. Elosztott ágens koncepció sematikus rajza a Distributed Prioritized Experience Replay algoritmusban [13]

Tanulás-alapú megközelítésekkel szemben. Egy másik módszert, a Stochastic Ensemble Value Expansion-t (STEVE) ismerteti [16], amely ötvözi a model-free megközelítések kiemelkedő teljesítményét a model-based algoritmusok előnyösebb mintavételi komplexitásával. Statisztikai analízis alapján kijelenthető, hogy a STEVE jelentős javulást eredményez a mintavételezési hatékonyságban számos összetett feladat kapcsán, összehasonlítva a model-free megközelítésekkel. Az ilyen, és ehhez hasonló, további lehetséges megoldásokról a mintavételezési hatékonyság problémájára, valamint egy teljeskörű áttekintésről ezen a területen lásd: [17].

Az eddig bemutatott módszerek mellett néhány, azon megközelítésektől teljes mértékben eltérő tantóminta priorizálási technika is létezik, mint például a [18]-ban kifejlesztett energia-alapú priorizálási keretrendszer (Energy-Based Prioritization). Ebben a publikációban a koncepció egy robotkar manipulációs környezetében került tesztelésre, illetve kiértékelésre. Az alapötlet nem más, mint hogy meghatározzuk a tervezett trajektória teljes energia felhasználását, a kumulatív állapotátmeneti energia kiszámításával, melyet a fizika alapelveiből határozunk meg, ahogyan azt az 1.1 egyenlet is mutatja:

$$E_{traj}(\tau) = E_{traj}(s_0, s_1, \dots, s_T) = \sum_{t=1}^T E_{tran}(s_{t-1}, s_t) \quad (1.1)$$

ahol τ a trajektória, s_i pedig az i -dik időlépés állapota. Az ágens ezután eltárolja ezeket az energiaértékeket a megfelelő állapotátmenetekhez, és ezáltal priorizálja az egyes trajektóriákat, amelyek később közvetlenül meghatározzák a valószínűségi eloszlást a mintavételezés során. A Curiosity-Driven Prioritization fogalom kerül bevezetésre [19]-ben, amely kifejezett hangsúlyt fektet az ágens által ritkán tapasztalt eseményekre. Azáltal, hogy az egyes tanítóminták sűrűségét ρ egy Variational Gaussian Mixture Model (V-GMM) segítségével értékeli ki, az ágens az alacsonyabb sűrűségű tapasztalatokat részesíti előnyben, ezzel kialakítva a prioritási sorrendet. A nagy sűrűségérték-oszcillációval rendelkező kiugró minták okozta zajok kiküszöbölése, ezáltal a rendszer robusztusságának növelése érdekében, a visszajátszási memóriában először a komplementer sűrűségek szerint ($\bar{\rho} = 1 - \rho$) sorrendet állítanak fel, majd az egyes minták sorszáma alapján történik maga a priorizálás. A [20] cikkben bemutatott Topological Experience Replay (TER) egy olyan módszer, amely a tanítás során gyűjtött állapotátmenetek rangsorolását egy gráf konstruálásával és reverse sweep algoritmussal végzi el, egy Q -függvény frissítésére. A generált gráf struktúráját tekintve a következő: a csomópontok a környezet egyes állapotainak reprezentációi, míg az élek az egyes állapotátmeneteket jelölik, másnéven a tapasztalatokat. Az algoritmus futása során számontartja a végállapotokat, illetve a kiindulási csúcsot, ezáltal a feladat már csak egy mintavételezési stratégia alkalmazása a kiválasztott állapotátmenet-szekvenciák által, a gráfban meghatározott úton. Ebben az esetben a stratégia a terminál állapotokból kiindulva, inverz Breadth-First Search keresés alkalmazását jelenti, amelyet a végállapotok elődállapotai követnek, és így tovább.

Végül pedig, [21]-ben definiálásra kerül az egyes tanítóminták tanulhatósági paramétere, valamint bevezetésre kerül egy algoritmus, amely szignifikánsan elősegíti az ágens információ-kihasználó képességét azáltal, hogy különbséget képes tenni a tanulhatatlan és a zajos minták között. A szerzők által felvázolt hipotézis szerint a tanulhatósági metrika a redukálható TD-hibából számítható, amely elméletileg azt mutatja meg, hogy a TD-hiba várhatóan mennyivel csökkenthető a tanítás folyamán. A redukálható veszteség (ReLo) formalizálása érdekében, amely prioritási értéként szolgál a súlyok frissítése során, két különböző veszteséget kell először meghatároznunk: egyet a beavatkozási hálózatra L_θ és egyet a célhálózatra vonatkozóan $L_{\hat{\theta}}$, amelyből a ReLo metrika az 1.2 egyenlet alapján számítható:

$$ReLo = L_\theta - L_{\hat{\theta}} \quad (1.2)$$

Összefoglalva, a Megerősítéses Tanulás területén több különböző megközelítést alkalmaztak sikeresen a mintavételezési hatékonyság problémájának kezelésére, köztük a Prioritized Experience Replay módszert. Ezen technikák használata által a tanítások hatékonysága és a mintavételezési teljesítmény egyaránt fejlődött, amely egyértelmű előrelépést jelent ezen a területen. Ugyanakkor érdemes megjegyezni, hogy a tanítóminta priorizálási módszerek körében a priorizált tanítóminta visszajátszás (Prioritized Experience Replay) továbbra is uralkodó marad a szakirodalomban, ezért a state-of-the-art technológiák közé sorolható, mind a mai napig.

1.3. Kontribúció

A konkrét tanítást végző algoritmusoktól függetlenül vitathatatlan, hogy a mintavételi stratégiának kulcsfontosságú szerepe van a tanítás hatékonyságában. Habár az utóbbi időben számos megközelítés került bemutatásra, ahogyan az 1.2-ben részletezésre kerül, azok a módszerek, amelyek a mintavételezés hatékonyságának növelésére összpontosítanak, jóval lassabb ütemben fejlődtek a Megerősítéses Tanulás más elemeihez képest, mint például az állapotrepresentációk, a Neurális Hálózati architektúrák vagy maguk az algoritmusok.

Ezen felül, az utóbbi időben nem történt jelentős áttörés a tanítóminta priorizálási módszerek terén, a legtöbb létező megoldás jelenleg is jelentős mértékben támaszkodik a Prioritized Experience Replay koncepciójára. Egyaránt tartós és kiemelkedő népszerűsége több tényezőnek is tulajdonítható, mint például a relatív implementációs egyszerűség, az adaptálhatóság és ezáltal az alkalmazhatóság különböző problémák esetére, valamint az integrálhatóság különböző algoritmusokhoz az ágens teljesítményének és stabilitásának javítása érdekében. Összefoglalva, figyelembe véve a fent említett indokokat, megállapítható, hogy a Prioritized Experience Replay továbbra is a Megerősítéses Tanulás, azon belül pedig a tanítóminta-priorizálás területének legszélesebb körben alkalmazott, ezáltal

a szakirodalomban uralkodó, state-of-the-art technológiájának minősül.

A kutatás kontribúciója több részből tevődik össze, ezek közül kiemelve a legfontosabbat bemutatásra kerül egy újszerű tanítóminta priorizálási megközelítés, illetve annak kiemelkedő teljesítménye a szakirodalomban alkalmazott jelenleg legkorszerűbb módszerrel szemben. Ez egy átfogó kiértékelés keretein belül kerül ismertetésre, a mintavételi folyamatok teljesítménymutatóinak részletes elemzésével. A dolgozat kontribúcióinak részletezése a következőkben olvasható:

- Ez a dolgozat egy új prioritásérték-hozzárendelési koncepciót mutat be, amely egy újfajta megközelítést hordoz magával az úgynevezett exploration-exploitation dilemma feloldásához. Az eredmények azt mutatják, hogy a fent említett, legkorszerűbb módszer limitációinak felszámolásával az ismertetésre kerülő módszerrel tovább javítható a mintavételezés hatékonysága.
- Négy-négy Deep Q -Network ágens került betanításra a bevezetett prioritás metrika felhasználásával, és a jelenleg legkorszerűbbnek tekinthető PER módszerrel négy, a szakirodalomban összehasonlításra és kiértékelésre egyaránt gyakran használt Megerősítéses Tanulási alapeladatra.
- Szigorúan azonos körülmények között, alapos kiértékelésnek lettek alávetve a fent leírt módon tanított ágensek, különös figyelmet fordítva a tanítás során alkalmazott különböző mintavételi stratégiák hatékonyságának értékelésére.

2. fejezet

Algoritmusok – Gépi Tanulás

A Gépi Tanulás területe megkérdőjelezhetetlen módon az elmúlt időszakban került a számítástechnika tudományának középpontjába, a Mesterséges Intelligencia vezető ágazataként. Ez a szakterület olyan algoritmuscsoportok és technológiák gyűjteményét foglalja magába, amelyek számítógépek számára lehetővé teszik, hogy adatokból valamilyen fajta visszacsatolás és optimalizálás alkalmazásával tanuljanak és azon alapuló döntéseket hozzanak vagy előrejelzéseket készítsenek, illetve ezáltal javítsák saját teljesítményüket anélkül, hogy kifejezetten egy adott programozási céllal lennének létrehozva. Ezért elmondható, hogy a Gépi Tanulás eszközei az adatvezérelt megoldások körébe tartoznak. A Gépi Tanulás további három fő típusra bontható, amelyek rendre a Felügyelt Tanulás (Supervised Learning), a Felügyelet Nélküli Tanulás (Unsupervised Learning) és a Megerősítéses Tanulás (Reinforcement Learning).

A Gépi Tanulásnak, lévén egy meglehetősen komplex és szerteágazó módszercsoportról beszélhetünk, számos alkalmazási területe létezik, beleértve többek között az objektum detekciót, a beszédfelismerést, autonóm járművek irányítását, különböző egészségügyi diagnosztikai módszereket és számos pénzügyi elemzési stratégiát is. Az autonóm járművek területén például legtöbbször Gépi Tanulást alkalmaznak a járható útfelület meghatározásához, amelyet szemantikus szegmentációval érnek el. Tehát egy útmenti képen a szemantikus szegmentáció képes meghatározni, hogy melyik pixel tartozik az út felületéhez, melyik a járdához, az autókhoz, a fákhöz és így tovább. Az egészségügyben Gépi Tanulás segíti az orvosokat pontosabb diagnózisok felállításában és kezelési tervek készítésében. Az adatok hatalmas szerepet játszanak tehát a Gépi Tanulásban, hiszen minél több, minél diverzebb és minél minőségibb adat áll rendelkezésre, annál pontosabb és hatékonyabb Gépi Tanulási modellek hozhatók létre. A Gépi Tanulás térnyerése máig folyamatos, amely még szélesebb körű alkalmazást biztosíthat a technológia területén. A következőkben a Gépi Tanulás három fő csoportja kerül bemutatásra, különös tekintettel a Megerősítéses Tanulásra.

2.1. Felügyelt Tanulás

A Felügyelt Tanulás a Gépi Tanulás egyik fő ága, amelynél az ágensek célja egy viszonylag kiterjedt, címkézett adathalmazból való tanulás, majd az így szerzett ismeretek felhasználása a nem címkézett adatok elemzésére, optimális predikcióra. A Felügyelt Tanulás magába foglal különböző algoritmusokat és módszereket, például a regressziót, a klasszifikációt, a multi-label tanulást (amikor egy képhez több címke is tartozik), egyes dimenzióredukciós eljárásokat, illetve a döntési fákat.

Felügyelt Tanulás során a döntéshozó objektumok feladata lényegében, hogy felfedezzék, majd megtanulják a tanítóminták és a címkék közötti összefüggéseket, gyakran komplex és egyaránt absztrakt kapcsolatokat, így kialakítva végső tudásukat és teljesítményüket. Példának véve a klasszikus objektum detekciót, a feladat egyaránt az objektumok észlelése, klasszifikálása és lokalizálása a bemenetként adott képek alapján. Először is a lokalizáció általában egy téglalap (bounding box) definiálásával történik, amely körülveszi az objektumot. Ezek után az ágensnek azonosítania kell a box-ban elhelyezkedő objektumok kategóriáját, másnéven osztályát. Természetesen az ágensnek képesnek kell lennie több objektum szimultán detektálására egy-egy képen, vagyis külön-külön kell azonosítania és elhatárolnia őket a környezettől.

Az autonóm járművek fejlesztése és elterjedése az utóbbi években jelentős figyelmet kapott a közlekedés és a technológia területén, amely a Felügyelt Tanulásnak is számos kulcsfontosságú alkalmazási területet kínál az autóiparban, melyek közül a legfontosabbak kerülnek részletezésre:

- **Környezetérzékelés:** Az autonóm járműveknek minden egyes időpontban pontosan kell tudniuk értelmezni az őket körülvevő környezetet annak érdekében, hogy biztonságosan közlekedhessenek. A Felügyelt Tanulás segíthet olyan problémák kezelésében, mint az objektumok, például gyalogosok vagy más járművek észlelése, vagy útszélek detektálása és jelzőtáblák azonosítása majd feldolgozása.
- **Trajektória tervezés, követés, valamint navigáció:** Ennél a kategóriánál beszélhetünk külön lokális, valamint globális trajektória tervezésről, amely során bizonyos optimalizálási célokon keresztül egy optimális útvonal kerül megkonstruálásra a térkép és a környezet adatainak elemzésével. Ezen kívül a Felügyelt Tanulás alapú ágensek képesek lekövetni a tervezett trajektóriát kameraképek alapján, amelyhez elengedhetetlen modul még a dinamikus akadályelkerülési stratégia is.
- **Döntéshozatal:** Az autonóm járműveknek jelentős mennyiségű bemeneti adat alapján szükséges egyaránt gyors és megbízható döntéseket hozniuk, például a közlekedési helyzetbe történő esetleges beavatkozásokhoz, és reagálni a változó körülményekre.

2.2. Felügyelet Nélküli Tanulás

A Felügyelet Nélküli Tanulás olyan Gépi Tanulási paradigma, ahol az ágensnek nincs hozzáférése a bemeneti adatokhoz tartozó címkékhez. A cél ebben az esetben a tanító-adatok egymással való összehasonlítása, relatív információ kinyerése, ezért ezt az eljárást gyakran alkalmazzák adattömörítésre, absztrakt adatmintázatok felderítésére, valamint az adatok struktúrájának megértésére. Az Unsupervised Learning két fő csoportra osztható: klaszterezésre és dimenzióredukcióra.

Klaszterezés során a feladat az adatok hasonlóságuk alapján történő különböző csoportokba (klaszterekbe) szervezése, amely egy látens n -dimenziós vektortérben kerül kialakításra. Ezek a csoportok lehetnek hierarchikusak vagy nem-hierarchikusak, de minden ebbe a csoportba tartozó algoritmus célja a klaszterek meghatározása és az objektumok valamelyik klaszterhez való hozzárendelése. A klaszterezés alkalmazható például szegmentációban, orvosi diagnosztikában és szociológiai elemzésekben is.

A dimenzióredukció egy másik jelentős Unsupervised Learning alkalmazás, amely során az n -dimenziós adatot egy kisebb dimenzióba transzformáljuk projekcióval vagy egyéb módszerekkel, általában vizualizációs cézzattal. Matematikailag az elsődleges cél az adatok komplexitásának csökkentése anélkül, hogy sok információt vesztenénk. A dimenzióredukció számos területen hasznos, például a képfeldolgozásban, zajcsökkentésben és szűrésben, illetve folytonos változók vizsgálatánál.

A Felügyelet Nélküli Tanulás gyakran komplex algoritmusok és modellek felhasználását igényli, mint például a generatív modellek, mivel azok képesek az adatok rejtett összefüggéseinek és sajátosságainak felfedezésére, ami hozzájárul az adatok gyors feldolgozásához és az információk hatékony kinyeréséhez.

2.3. Megerősítéses Tanulás

A rendelkezésre álló számítási kapacitás hirtelen növekedésének hatására a Megerősítéses Tanulás jelentős figyelmet kapott [22] számos, a Gépi Tanulásban érdekelt kutató által, ígéretes tulajdonságai és a már elért kiemelkedő eredményei miatt különböző szekvenciális döntéshozatali problémák esetében.

A Megerősítéses Tanulás a Gépi Tanulás területén belül egy speciális alterületként jelenik meg. A Felügyelt Tanuláshoz hasonlítva, azzal ellentétben egy Megerősítéses Tanulás-alapú ágens nem igényel hatalmas, címkézett adathalmazt, amelynek beszerzése, majd pedig feldolgozása gyakran korlátozott vagy bizonyos esetekben egyáltalán nem is megvalósítható. Ehelyett a tanításhoz szükséges adatot az ágens saját maga generálja le önmagának egy másik, úgynevezett környezet objektummal való interakció sorozat folyamán.

Az RL egyedisége és alapvető differenciája más Gépi Tanulási módszerekhez képest a

fent leírt interakciós folyamatból eredeztethető. Minden egyes iterációs lépésben egy, az állapotátmenetet egyértelműen definiáló változók halmaza kerül összeállításra, amelyben egy úgynevezett reward (jutalom) függvény szolgál a beavatkozások minőségének numerikus reprezentációjára, amely később a Neurális Hálózat súlyainak frissítése során kerül felhasználásra. Ezen tulajdonság következményeképp az ágens teljesítménye nem korlátozódik egy előre definiált és felcímkézett adathalmazra, hanem sokkal inkább a fejlesztő saját intuícióján, majd döntésein múlik, hogy milyen performanciát képes elérni az ágens a számára kulcsfontosságú absztrakciók meghatározásán keresztül. Ezen absztrakciók az adott feladat egy magasabb szintű reprezentációi, beleértve az állapotteret, a beavatkozási teret, valamint az említett jutalomfüggvényt is. A felsorolt komponensek átgondolt megválasztása központi szerepet játszhat a credit assignment (érték hozzárendelési) folyamatban, közvetve pedig az ágensek által elérhető teljesítményben is.

Mivel egy előre betanított Neurális Hálózat online döntéshozatali (predikciós) ideje milliszekundumos nagyságrendbe esik, az RL számos előnye közül az egyik a korlátozás nélküli valós idejű alkalmazhatóság, amely lehetővé teszi az ágensek számára, hogy gyorsan reagáljanak és alkalmazkodjanak a változó környezeti elemekhez és történésekhez, amely hasznos tulajdonság dinamikus környezetekben, mint például a robotikában vagy forgalomirányítási feladatokban [23] [24].

A Neurális Hálózatok nemlineáris függvény-approximátorként történő alkalmazásának további előnye a skálázhatóság, amely különösen hasznosnak bizonyul a több összekapcsolt komponensből álló komplex rendszereknél. Ennek a tulajdonságnak köszönhetően az RL-ágens képes a megszerzett ismereteit kiterjeszteni akár jóval nagyobb méretű, és ezáltal magasabb komplexitású rendszerek kezelésére, hatékony koordinációt és optimális döntéseket eredményezve, egyszerre akár több absztrakciós szinten keresztül. Ez különösen fontosnak bizonyul olyan területeken, mint a hálózat menedzsment, az ellátási lánc optimalizáció és az intelligens hálózatok (smart grid) [25] [26] [27].

Végül pedig az RL egyik, sokak által régóta vizsgált előnye a generalizáló képesség, amely képes feloldani a környezetben felmerülő változásokat, bizonytalanságokat, előre nem látható körülményeket és szélsőséges helyzeteket. Ez biztosítja többek között, hogy a döntések megbízhatóak és robusztusak maradnak abban az esetben is, ha az aktuálisan kialakult szituáció eltér a tanítás során közvetlenül tapasztaltaktól. Mindemellett ez képezi az úgynevezett a Transfer Learning technikák alapját [28], amelyek lehetővé teszik az RL ágensek számára, hogy a megszerzett viselkedésformát egyik feladatról vagy környezetről a másikra transzformálják, ezzel minimalizálva a jelentősebb extra tanítások szükségességét, közvetve pedig csökkentve az idő és a számítási erőforrások által generált finansiális vonzatokat, miközben további doméneken gyorsítja a tanítás folyamatát.

A Megerősítő Tanulás matematikai keretrendszerét a Markov Döntési Folyamatok (Markov Decision Processes) [29] határozzák meg, amelyre gyakran $\langle S, A, R, P \rangle$ -ként hivatkoznak, benne a döntések meghozásáért felelős objektumot ágensnek nevezik. Egy

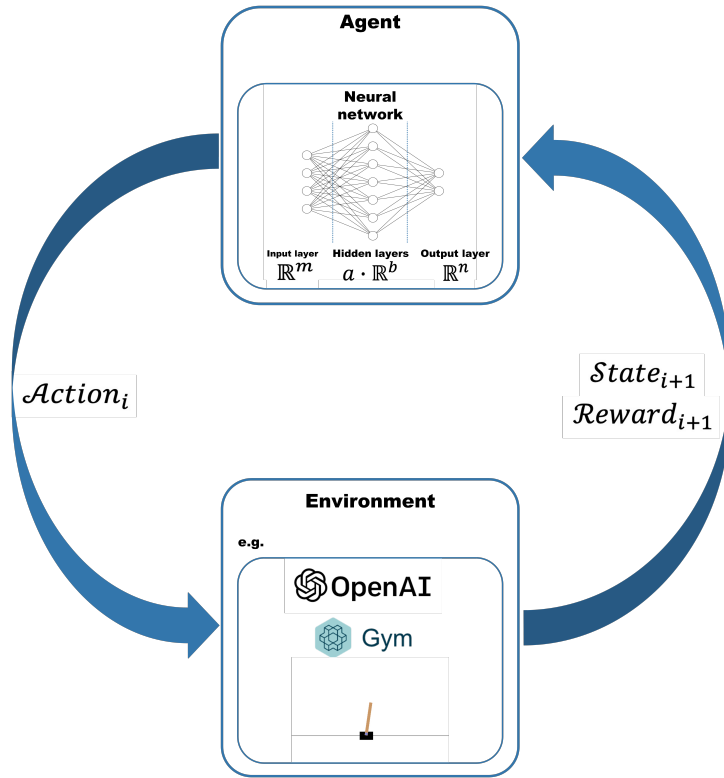
Markov Döntési Folyamat négy elem által definiált egyértelműen, amelyek rendre a következők:

- $s_t \in S$ a t -dik időlépésben vett állapot, ahol S az állapotteret jelöli
- $a_t \in A$ a t -dik időlépésben az s_t állapotból választott beavatkozás, ahol A a beavatkozási tér
- r_{t+1} a reward (jutalom) érték, amely az s_t állapotból az a_t beavatkozás hatására létrejövő állapotátmenet minőségét jelző skalár szám
- $P(s_{t+1}|s_t, a_t)$ annak a valószínűsége, hogy a t -dik időlépésben az s_t állapotból a_t beavatkozás választása által a $t + 1$ -dik időlépésben az s_{t+1} állapotba kerülünk

A fent leírtak alapján egy tanítás az ágens és a környezet között lejátszódó interakciók sorozataként jelenik meg, amely közben az adott állapotátmenetet leíró jelzők (s_t kiindulási állapot, a_t beavatkozás, s_{t+1} végállapot és r_{t+1} reward érték) által generálódik le a tanításhoz szükséges adathalmaz. Ebben a folyamatban a környezet kettős célt szolgál: egyrészt alapvető információkat szolgáltat a környezet aktuális állapotáról, lehetővé téve az ágens számára, hogy a megfigyelések alapján döntéseket hozzon. Másrészt számszerűsíti a választott beavatkozások hatékonyságát azáltal, hogy egy skalár visszacsatolási értéket, az úgynevezett jutalmat vagy reward-ot, juttat vissza az ágenshez. A fent említett kommunikáció, vagyis a tanítási ciklus egy iterációja látható a 2.1 ábrán, amely rendre a következőképpen játszódik le:

1. Az ágens az előző időlépésben kapott állapotrepresentációt döntéshozásra használja fel.
2. Ez a döntés beavatkozás formájában továbbítódik a környezet felé, amelynek aktuális állapota ennek hatására megváltozik.
3. A környezet állapotának megváltozása állapotátmenetet idéz elő, amelynek két lényeges eredménye jelenik meg: a következő időlépés állapotrepresentációja és az állapotátmenethez tartozó reward érték.
4. Ezek a metrikák visszakörülnek az ágenshez.
5. A következő állapotrepresentációt az ágens újbóli döntéshozatalra használja, míg a reward egyfajta visszacsatolásként szolgál, amely alapján bizonyos időközönként a Neurális Hálózat súlyainak frissítésére kerül sor.

A tanítás során a Neurális Hálózat súly- és bias paraméterei iteratív módon kerülnek frissítésre két további lényeges komponens felhasználásával, amelyek a költségfüggvény és a sztochasztikus gradiens süllyedési eljárás. Matematikai szempontból az ágens célja



2.1. ábra. Megerősítéses Tanulás-alapú ágens tanítását reprezentáló interakciós folyamat

a kumulált súlyozott reward értékének maximalizálása egy adott időhorizonton, amelyet formálisan 2.1 alapján a következőképpen lehet kifejezni:

$$G_t = \sum_{k=t+1}^T \gamma^{k-t-1} \cdot R_k \quad (2.1)$$

ahol t az időlépés, T a vizsgált időhorizont lépésekben kifejezve (vagy az epizód végállapotaig végbemenő lépések száma, amennyiben létezik ilyen), R_t a t -dik időlépésben szerzett jutalom értéke, γ pedig az úgynevezett discount faktor, amely az azonnali reward-ok (állapotátmenetek) fontosságát egyensúlyozza a jövőbeli jutalmakkal (állapotokkal) szemben.

2.3.1. Policy-alapú módszerek

A Megerősítéses Tanulás területén belül a különféle megközelítéseknek közös célja, hogy az ágens meg tanuljanak adott környezetben olyan optimális döntéseket hozni, amelyek maximalizálják a G_t kumulált súlyozott reward értéket, ezen lehetséges megvalósítások közül egy módszer a policy-alapú Megerősítéses Tanulás.

Policy-alapú RL esetben az elsődleges hangsúly egy egzakt policy (viselkedés) megtanulására helyeződik, amely egy leképezést jelent az állapotokból a beavatkozásokra nézve. Mivel maga a policy határozza meg, hogy az egyes állapotokban milyen beavatkozást érdemes végrehajtani a várható kumulatív jutalom maximalizálása érdekében, ezen mód-

szerek közvetlenül a policy-t paraméterezik valamely optimalizálási technika alkalmazása által, például a gradiens emelkedés, így frissítve a súlyok értékét a megfigyelt jutalmak függvényében. Ebbe a kategóriába tartozik többek között a REINFORCE, a Proximal Policy Optimization (PPO) és a Trust Region Policy Optimization (TRPO) algoritmus is.

A policy-alapú módszereket gyakran alkalmazzák olyan nagydimenziós, vagy folytonos beavatkozási térrel rendelkező problémák esetében, ahol komplikált lenne az értékfüggvények közvetlen meghatározása.

2.3.2. Modell-alapú módszerek

A modell-alapú módszerek a környezet modelljének direkt tanulására összpontosítanak. Ebben az esetben egy modell szimulálja a tényleges környezet várható viselkedését, lehetővé téve az ágens számára, hogy a modell alapján tervezzen és hozzon döntéseket anélkül, hogy közvetlenül kapcsolatba lépne a valós környezettel.

A modell-alapú módszerek gyakran tartalmazznak, ezáltal tanulnak átmeneti dinamikákat, reward modelleket, illetve állapotátmeneti valószínűségeket a környezetből. Ezeket az információkat később az ágens felhasználhatja tervezéshez például olyan módszerek segítségével, mint a Monte Carlo Tree Search (MCTS) vagy a Modell Prediktív Irányítás (MPC).

Ez a megközelítés ott bizonyul hasznosnak, ahol a valós környezettel való interakció lebonyolítása valamilyen szempontból költséges, ezért előnyösebb a környezet modelljéből való tanulás.

2.3.3. Érték-alapú módszerek

Az érték-alapú módszerek a Megerősítéses Tanulás területén olyan algoritmuscsoportot jelölnek ki, amelyek célja egy úgynevezett értékfüggvény minél pontosabb approximálása, amely az egyes állapotokhoz, vagy állapot-beavatkozás párokhoz kapcsolódó értékeket határozza meg. Az értékfüggvény kimenete az ágens által elérhető, várható hozam vagy kumulatív reward értéke, miközben egy adott viselkedést követ az adott epizód folyamán. Ezért a tanítás elsődleges feladata olyan optimális értékfüggvény meghatározása, amely maximalizálja a várható kumulatív jutalom értékét. Az állapotok vagy állapot-beavatkozás párok értékeinek közelítésével az ágens képessé válik arra, hogy megbízható döntéseket hozzon arra vonatkozóan, hogy milyen beavatkozások választása által maximalizálhatja hosszú távú jutalmát. Egy állapot értéke a 2.2 szerint számítható ki:

$$V_{\pi}(s) = \mathbf{E}_{\pi} [G_t | S_t = s] \quad (2.2)$$

ahol $\mathbf{E}$ a várható érték operátor, G_t a kumulált súlyozott reward, s az állapot, amely értékének a becslését végezzük el, és π az a viselkedés, amelyet az ágensnek a t -dik időlépés után követnie kell az approximált érték realizálásához.

Érdemes megemlíteni, hogy az érték-alapú módszerek elsősorban az értékfüggvény meghatározására fókuszálnak, mintsem hogy direkt módon törekednének egy viselkedésforma elsajátítására. Ettől eltekintve természetesen a becsült értékekből is konstruálható policy, például mindig az egyes állapotokon belüli legnagyobb értékkel rendelkező beavatkozás kiválasztása által.

2.3.3.1. Q -tanulás

Az érték-alapú Megerősítéses Tanulási algoritmusokon belül, a különböző megközelítések közül kiemelkedő teljesítményt nyújtanak számos doménban a Q -tanulási eljárások [30]. A Q -tanulás egyaránt off-policy és model-free megoldás, ami azt jelenti, hogy az ágens nem igényel előzetes ismereteket a környezet belső dinamikájáról, sem az állapotátmeneti valószínűségekről. A tanítás során folyamatosan frissítésre kerül az állapot-beavatkozás párokra vonatkozó értékfüggvény becslése, amelyet ebben az esetben Q -függvénynek neveznek. Hasonlóan az általános érték-alapú algoritmusokhoz, a Q -értékek számítása a 2.3 szerint történik:

$$Q_\pi(s, a) = \mathbf{E}_\pi [G_t | S_t = s, A_t = a] \quad (2.3)$$

ahol a 2.2 egyenlethez képest egy további a paraméter szerepel, amely az s állapotból választott beavatkozást jelöli.

A Q -értékek iteratív frissítése a környezetből megfigyelt reward értékeken és az interakció sorozat közben tapasztalt állapotátmeneteken alapul, egy kezdeti Q -táblázatból vagy Q -függvényből kiindulva, majd a beavatkozásokat valamely policy alapján kiválasztva egy exploration-exploitation stratégiát követve, mint például az ϵ -greedy. Ezen megfigyelések felhasználásával a Q -értékek frissítése a Bellman-egyenlet szerint történik, amely az egymást követő állapotok Q -értékei közötti kapcsolatot fejezi ki a 2.4 egyenletben látható módon:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \cdot [r + \gamma \cdot \max_{a'} Q(s', a') - Q(s, a)] \quad (2.4)$$

ahol α a tanítási ráta, r a jutalom értéke, γ a discount faktor és $\max_{a'} Q(s', a')$ a legnagyobb Q -érték, amely a következő s' állapotból érhető el.

2.3.3.1.1. Deep Q -Network

A Deep Q -Network, vagy röviden DQN, egy olyan specifikus Megerősítéses Tanulási algoritmus, amely ötvözi a Q -tanulás elveit a Mély Neurális Hálózatok kiemelkedő képes-

ségeivel. Habár a DQN algoritmust eredetileg Atari-játékokon demonstrálva mutatták be, azóta kutatók széles köre szentelt neki jelentős figyelmet rendkívüli teljesítménye miatt, amelyet nem csak a játékok területén, hanem később számos más alkalmazásban, például a robotikában, az ajánlórendszerekben és az autonóm járműirányításban is bizonyított.

A DQN lényege, hogy egy Mély Neurális Hálózatot alkalmazunk a Q -függvény approximálására, így a nagydimenziós, komplex bemeneti vektorterek kezelése is könnyen megvalósítható. Az optimális Q -függvény közelítését a prediktált-, és a cél (target) Q -értékek között felírható átlagos négyzetes eltérés alapú költségfüggvény minimalizálásával éri el, amelynek matematikai formalitása látható 2.5-ban:

$$\mathfrak{L}_{MSE} = \frac{1}{N} \cdot \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.5)$$

ahol N a tanítóminták darabszáma, y_i az i -dik indexű mintához tartozó cél (target) Q -érték, $\hat{y}_i$ pedig szintén az i -dik indexű mintához tartozó prediktált Q -érték.

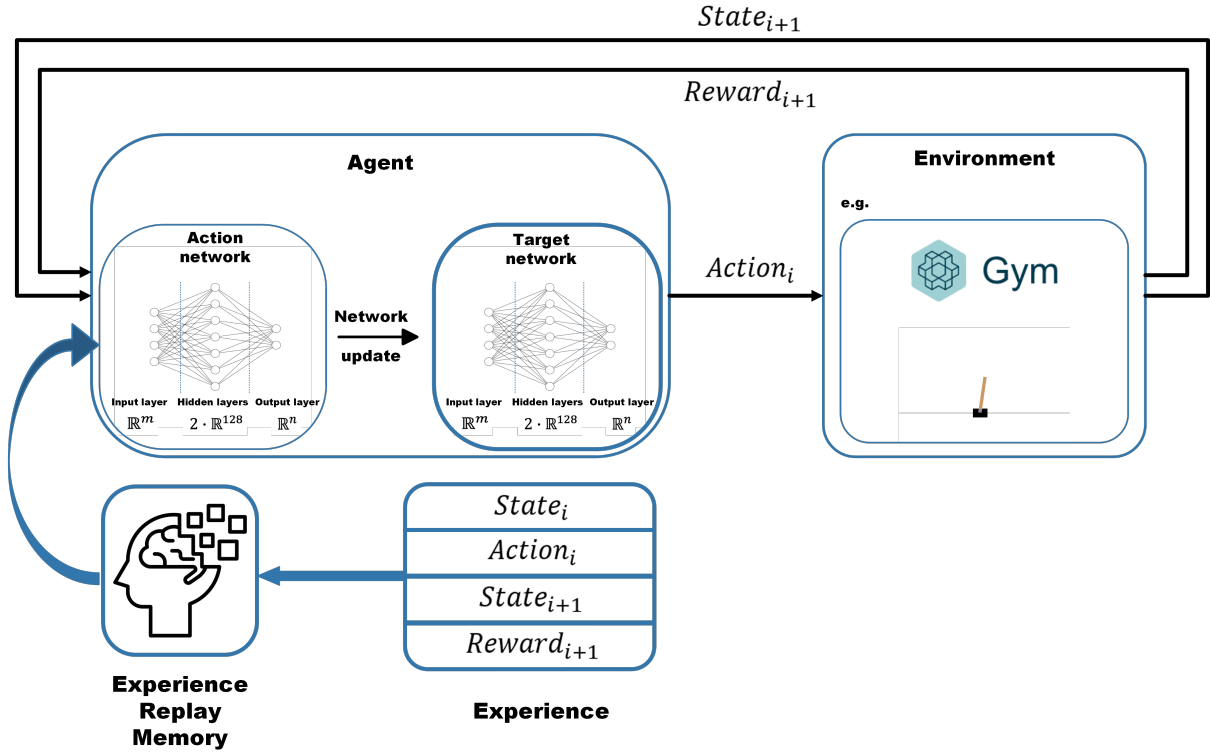
A DQN algoritmus bevezetése két fő újítást hozott a Megerősítéses Tanulás területére, amelyek egyike a két azonos struktúrájú Neurális Hálózat alkalmazása. Az elsődleges hálózat, az úgynevezett beavatkozási hálózat, magáért a döntéshozásért felelős, míg a másodlagos hálózat, amelyet célhálózatnak (target hálózatnak) nevezünk, a cél (target) Q -értékek kiszámítására szolgál. A Neurális Hálózat súlyainak frissítése közvetlenül a beavatkozási hálózaton történik, míg ezen paraméterek periodikusan, előre rögzített időlépésenként kerülnek átmásolásra a másodlagos hálózatra. Ez a megközelítés stabilizálja a tanulási folyamatot azáltal, hogy megbízhatóbb target értékeket biztosít a tanítás során, amely az egymást követő állapotátmenetek közötti korreláció megszüntetésével, vagy drasztikus lecsökkentésével érhető el. A cél (target) Q -értékek számítása a hagyományos Q -tanuláshoz hasonlóan, a Bellman-egyenlet segítségével történik 2.6 szerint:

$$Q(s_t, a_t; \theta_t) = r_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a_{t+1}; \theta_t^*) \quad (2.6)$$

ahol θ_t a beavatkozási (elsődleges) hálózat súlymátrixa, míg θ_t^* a másodlagos-, vagy célhálózat (target network) paraméterhalmaza.

A DQN másik újítása az úgynevezett tanítóminta visszajátszás technika alkalmazása, amely szintén növeli a tanulás hatékonyságát. Ebből kifolyólag az ágens tapasztalatait, elemenként az állapotátmenetet definiáló összetartozó [kiindulási állapot, beavatkozás, reward érték, végállapot] négyes értékeket, egy úgynevezett tanítóminta visszajátszási memóriában tároljuk el. A tanítás során frissítési lépésenként egy-egy batch-nyi tanítóminta kerül sztochasztikusan, egyenletes eloszlás szerint mintavételezésre, amely szintén segíti az állapotátmenet-szekvenciák dekorrelációját és a bias csökkentését a súlyok értékeinek frissítésénél.

Összefoglalva, az algoritmus működése szerint módosított Megerősítéses Tanulás ta-



2.2. ábra. Deep-Q Network tanítási folyamata kiemelve az algoritmus két fő újítását

nítási folyamata látható a 2.2 ábrán, a DQN két kulcsfontosságú újítását kiemelve: a tanítóminták későbbi felhasználására szolgáló memória használatát és a különálló, másodlagos/target Neurális Hálózatot, amelyek egyaránt hozzájárulnak a tapasztalatok közötti korreláció csökkentéséhez, ezáltal a tanítási hatékonyság növeléséhez.

2.3.4. Tanítóminta priorizálás

Ahogy az korábban ismertetésre került, a DQN algoritmusnál került bevezetésre az úgynevezett tanítóminta visszajátszási memória koncepciója, amely lehetővé teszi a korábban szerzett tanítóminták újrafelhasználását, felgyorsítva ezzel a konvergencia folyamatát és növelve a tanítás hatékonyságát. Azonban az egyes tapasztalatok kezelése ebben az esetben szuboptimálisnak mondható, mivel a memóriából történő mintavételezés sztochasztikusan, egyenletes eloszlás alkalmazásával valósul meg a tanítás folyamán. Ez természetesen a gyakorlatban azt jelenti, hogy minden egyes állapotátmenet ekvivalens valószínűséggel kerül kiválasztásra a hálózat frissítése során. Ennek következtében felmerül a kérdés, hogy valóban azonos-e a tanítóminták mindegyikének információsűrűsége. Míg a környezetben gyakrabban előforduló állapotátmenetek redundáns információtaralmat hordoznak magukkal, kialakulhatnak olyan ritka helyzetek is, amiből származó tanítóminták kritikusnak mondhatók az ágens tanulási folyamatának szempontjából.

Ennek okán kijelenthető, hogy az egyes tapasztalatok a tanítás egy adott állapotában nem egyenlő fontosságúak, illetve hogy azok értéke a tanítás során dinamikusan változ-

hat. A következő fejezet egy részletes áttekintést nyújt mind a szakirodalomban jelenleg leggyakrabban felhasznált módszerről, mind pedig az általunk bemutatásra kerülő megközelítésről, amelyek egyaránt a tanítómintákhoz frissítési lépésenként, egyénileg rendelnek hozzá úgynevezett prioritás értékeket, ezzel növelve a mintavételi hatékonyságot. Lényegében ezzel a technikával a visszajátszási memória elemeinek tanítás közbeni, dinamikus súlyozását valósítják meg, az elemek információtartalmának különböző módokon való rangsorolása által.

2.3.4.1. Priorizált Tanítóminta Visszajátszás

A [11]-ben bemutatott Priorizált Tanítóminta Visszajátszás algoritmus a szakirodalom alapján továbbra is egyike azon leggyakrabban alkalmazott tanítóminta priorizálási módszereknek a Megerősítéses Tanulás területén, amely a korábban említett, tanítóminták közötti prioritásérték-beli különbség által indukált problémát hivatott megoldani.

Általánosságban a tanítóminta priorizálás mögött rejlő alapötlet triviális: a rendelkezésre álló (memóriában tárolt) állapotátmeneteket sorba rendezzük vagy egy prioritás metrikát rendelünk hozzájuk, így rangsorolva a tanítómintákat fontosságuk szerint az ágens aktuális tudásállapota szempontjából. Az elemenkénti prioritásérték hozzárendelése alapuló módszerek esetében a cél minden esetben egy prioritás metrika meghatározása oly módon, hogy minél fontosabbnak véljük az adott tapasztalatot, annál magasabb a hozzárendelt numerikus érték. A fő kihívás ezekben az esetekben, hogy meghatározzuk az(oka)t a metriká(ka)t/mérőszám(oka)t, amely(ek) alapján az egyes tapasztalatok prioritása megállapítható.

A már említett Priorizált Tanítóminta Visszajátszás esetében a rangsorolás alapját az úgynevezett Temporal Difference (TD) hiba adja, amely a környezetben az állapotátmenet közben megfigyelt reward érték és az állapotátmenet kiindulási állapotára adott prediktált érték közötti eltérést jelenti, ahogy látható 2.7-ben:

$$TD = \left[r_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a_{t+1}; \theta_t^*) \right] - Q(s_t, a_t; \theta_t) \quad (2.7)$$

ahol $Q(s_t, a_t; \theta_t)$ a t -dik időlépésben végbemenő állapotátmenet s_t kiindulási állapotra való predikciója a_t beavatkozás által, míg $r_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a_{t+1}; \theta_t^*)$ ugyanazon t -dik időlépésben vett környezeti megfigyelés. Lényegében minél magasabb az így definiált TD-hibatag, annál nagyobb a várható tanulási előrehaladás, vagyis az adott minta értékes információkat tartalmaz, ezért az ágens potenciálisan többet tud belőle tanulni. A sztochasztikus mintavételezéshez a teljes tanítóminta visszajátszási memóriára vetített valószínűségi eloszlás a 2.8 által adott képlet alapján számítható ki:

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha} \quad (2.8)$$

ahol $P(i)$ az i -dik indexű memóriában tárolt minta kiválasztásának valószínűsége, α a priorizálás mértékét meghatározó kitevő, p_i pedig az i -dik indexű állapotátmenet prioritásértéke.

Az eredeti publikációban két különböző megközelítés került ismertetésre a prioritásértékek számítására, amelyek közül az egyik egy indirekt, rangsor-alapú priorizálási módszer, amely a prioritásokat a 2.9 egyenletben látható módon formalizálja:

$$p_i = \frac{1}{\text{rank}(i)} \quad (2.9)$$

ahol $\text{rank}(i)$ az i -dik indexű minta numerikus rangjával egyezik meg, az abszolút TD-hibatagok $|\delta_i|$ alapján konstruált rendezett listában.

A másik, a szakirodalom által szélesebb körben alkalmazott koncepció egy direkt proporcionális módszer, amelynél a prioritásértékek meghatározása 2.10 szerint történik:

$$p_i = |\delta_i| + \epsilon \quad (2.10)$$

ahol $|\delta_i|$ az i -dik indexű állapotátmenet abszolút TD-hibaértéke, ϵ pedig egy kis értékű konstans, amely megakadályozza a zérus értékű TD-hiba szélsőséges eseteit.

A nem egyenletes eloszlást alkalmazó mintavételezésből adódó numerikus torzítások kezelésére, végül a Neurális Hálózati frissítések korrigálására további súlyok kerültek bemutatásra (importance sampling weights), amelyek biztosítják az egyensúlyt a tanítás folyamán. Matematikailag ezen súlyok definíciója látható 2.11-ben:

$$w_i = \left(\frac{1}{N} \cdot \frac{1}{P(i)} \right)^\beta \quad (2.11)$$

ahol β a numerikus torzításkompensáció mértékét szabályozó kitevő, N a visszajátszási memóriában lévő minták darabszáma, $P(i)$ pedig az i -dik indexű elem választási valószínűsége.

A Prioritizált Tanítóminta Visszajátszás jelentős előnyei a tanítás hatékonysága és az elért performancia szempontjából számos empirikus kísérlet által kerültek alátámasztásra különböző alkalmazási területeken, az Atari-játékoktól kezdve bonyolultabb folytonos irányítási feladatokig egyaránt [31] [32].

2.3.4.2. Upper Confidence Bound-alapú tanítóminta priorizálás

Megerősítéses Tanulásban az úgynevezett exploration-exploitation dilemma egy fundamentális problémaként jelenik meg, mivel egy bizonyos feladat vizsgálatakor fontos megtalálni az egyensúlyt a környezet ismeretlen állapotainak felfedezése és az ágens aktuálisan meglévő tudásának kihasználása között, a kumulatív reward maximalizálása végett.

Ez tehát egy kompromisszum realizálását követeli meg, amelyben az exploration (felfedezés) tag olyan viselkedésformát jelent, amely által az ágens a környezetből új, eddig még

nem ismert, vagy kevésbé felfedezett állapotokból generál újabb tanítómintákat magának. Ennek célja, hogy az ágens számára potenciálisan jobb állapot-beavatkozás párokat is felfedezzen, amelyek hosszú távon magasabb jutalomhoz vezethetnek. Az egyensúly másik fele, az exploitation (kihasználás) tag, olyan viselkedéshez vezet, amely a már megszerzett tudás felhasználásával választ ki olyan beavatkozásokat, amelyek által az ágens aktuális tudása szerint a legmagasabb jutalmat képes elérni korábbi tapasztalatai alapján. Az optimális egyensúly megtalálása kulcsfontosságú a tanítás és a mintavételezési hatékonyság szempontjából egyaránt.

Jelentős probléma lévén, ahogy korábban már említésre került, számos exploration (felfedezési) stratégia és algoritmus került kidolgozásra, amelyről [33] részletesebb áttekintést nyújt. Ilyen például a már ipari sztenderdnek számító ϵ -greedy, a softmax exploration, a Thompson mintavételezés és az Upper Confidence Bound (UCB) algoritmus is, amelyet [34]-ben mutattak be, ez utóbbi a későbbiekben részletezésre is kerül.

A fent ismertetett egyensúly fenntartása a két tag, az exploration és az exploitation, között ugyancsak fontos a Priorizált Tanítóminta Visszajátszás esetében is. Teoretikusan belátható, hogy a PER algoritmus esetenként túlmintavételezhet bizonyos tapasztalatokat TD-hibájuk alapján, mivel ez az egyetlen, exploitation tagként interpretálható metrika, amely a prioritásértékek formálásáért felelős. Az eddigi logika alapján pedig ez potenciálisan akadályozhatja a felfedezést, amely szuboptimális eredményhez vezethet komplexebb feladatok esetében. Ebben a dolgozatban az exploration-exploitation dilemma újszerű megközelítéssel való feloldása által kínálunk megoldást erre a problémára.

Az Upper Confidence Bound a Megerősítéses Tanulás területén belül egyike a népszerűbb exploration-exploitation dilemma kezelésére bemutatott algoritmusoknak. Az UCB működési elve, hogy úgynevezett konfidencia határokat szab meg és ezekkel igyekszik egyensúlyozni a két tagot annak érdekében, hogy megbecsülje a különböző állapotok és/vagy beavatkozások összértékét. Ennek megfelelően minden állapothoz és/vagy beavatkozáshoz tartozik egy felső konfidencia határ (Upper Confidence Bound), amely a bizonytalanságot számszerűsíti adott időpontban. Az UCB értékek meghatározása 2.12 szerint történik:

$$UCB = \bar{x}_j + \sqrt{\frac{2 \cdot \ln(n)}{n_j}} \quad (2.12)$$

ahol $\bar{x}_j$ reprezentálja az exploitation tagot, a j állapot vagy beavatkozás esetén megfigyelt átlagos reward értékén keresztül. Az exploration komponens megfogalmazása mögötti elv, hogy a nagyobb bizonytalansággal rendelkező állapotokat és beavatkozásokat prioritáljuk, mivel a bizonytalansági faktor miatt potenciálisan magasabb jutalmat is képesek lehetünk elérni általuk. Ebből adódóan n_j azt jelöli, hogy a j állapot vagy beavatkozás hányszor került kiválasztásra, míg n az eddigi összes választás összege.

A dolgozat fő kontribúciójaként egy újszerű megközelítés kerül bemutatásra tanító-

minták prioritizálásához, egy alternatív prioritásérték-hozzárendelési koncepció alkalmazása által. A metrika matematikai formalitása az Upper Confidence Bound algoritmus tanítóminta visszajátszási kontextusában való interpretációjával került definiálásra. Módszerünk a 2.13 egyenletben látható módon számolja ki az egyes tanítómintákhoz rendelt prioritásértékeket:

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha} + c_p \cdot \sqrt{\frac{2 \cdot \ln(\max_k n_k)}{n_i + \epsilon}} \quad (2.13)$$

ahol az exploration komponenst két metrika matematikai viszonya határozza meg: n_i frissítési számláló (fit count) azt jelzi, hogy a memóriában i -dik indexű állapotátmenet hányszor került felhasználásra a Neurális Hálózat súlyainak hangolási lépései során, és $\max_k n_k$, a memóriában lévő állapotátmenetek közül maximális frissítési számláló értéke. Továbbá ϵ egy kis értékű konstans, amely megakadályozza a nullával való osztást, c_p pedig szintén konstans, amely az exploration és exploitation tagok közötti egyensúlyt szabályozza.

A bemutatott algoritmus egyesíti a Priorizált Tanítóminta Visszajátszás hiányosságainak feloldásából származó performancia-beli előrelépéseket és az exploration-exploitation dilemma feloldásának hatékonyabb stratégiáját, amely a jelenlegi state-of-the-art módszerhez képest magasabb mintavételezési hatékonyságot és gyorsabb konvergenciát eredményez.

3. fejezet

Környezet

Ahogy az 2.3-ban ismertetésre került, az úgynevezett környezetek kulcsszerepet játszanak a Megerősítéses Tanulás alapú ágensek tanításában, mivel ezek felelősek a virtuális világ szimulálásáért, amellyel a döntéshozó entitások interakcióba kerülnek, ezáltal döntéseik eredményeként realizálva az állapotátmeneteket. Az ágens tanulása szempontjából alapvetően szükséges absztrakciókat, mint például az állapotrepresentációk és a jutalmak, a környezet biztosítja számára, amely révén képes lesz információt kinyerni saját döntéseinek következményeiből, végeredményül tanító adatot generálva önmagának.

Egy ilyen RL-környezet általános esetben több komponensből tevődik össze: az S állapotteréből, az A beavatkozási térből, az állapotváltozások dinamikájából, végül pedig az $R(s_{t+1})$ jutalomfüggvényből. Az állapotter magába foglalja a környezet összes lehetséges állapotát, amelyek bekövetkezési valószínűsége nem-zérus, míg a beavatkozási tér meghatározza az elérhető, engedélyezett beavatkozások halmazát, amelyeket az ágens adott kezdeti állapotból végre tud hajtani. Az állapotváltozások dinamikája egyértelműen definiálja az állapotok közötti átmeneteket a kiválasztott beavatkozások alapján. A környezet továbbá rendelkezik egy jutalomfüggvénnyel (reward), amelyen keresztül egy visszacsatolásnak szolgáló skalár szám kerül meghatározásra az ágens cselekvéseinek hatékonyságáról.

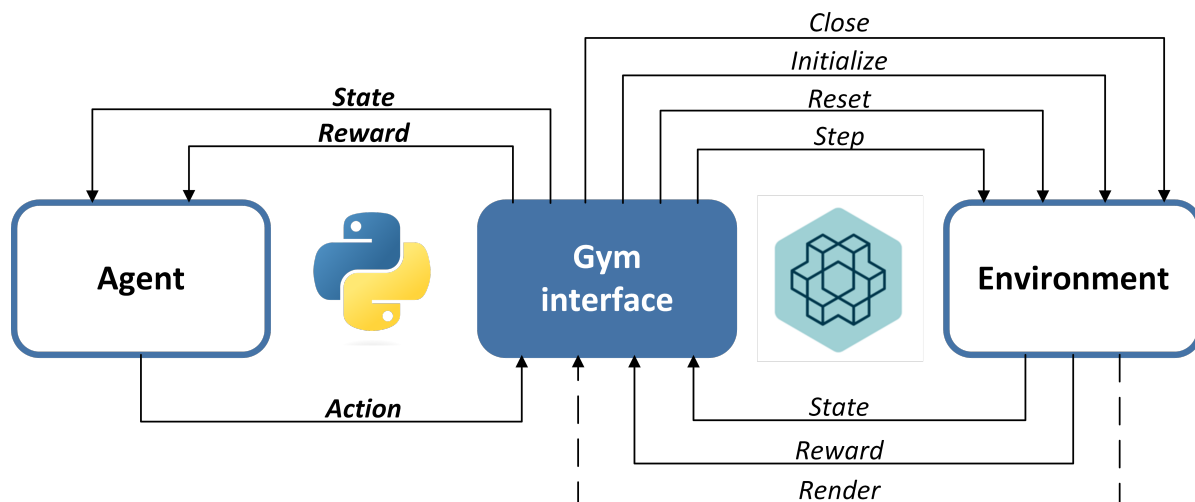
A környezetek strukturálisan felettébb sokszínűek lehetnek. Egyszerűbb esetben manifestálódhatnak például szimpla grid world-ökben, vagy társasjátékokban, de realizálhatnak akár komplex szimulációkat is a valóság egyes területeinek virtualizálása által, mint például a robotika, közlekedési hálózatok vagy a pénzügyi piacok. Ebben a dolgozatban a Gym keretrendszer által nyújtott, a Megerősítéses Tanulás kutatási területén belül széles körben ismert és alkalmazott sandbox-szerű, benchmark-nak számító környezeteket használtuk fel a különböző mintavételezési stratégiát alkalmazó algoritmusok mintavételi hatékonyságának vizsgálatához.

3.1. Gym környezetek

Ahogy az említésre került, a Gym az RL alkalmazások területén egy jelentős mértékben elterjedt eszközkészlet Megerősítési Tanulás-alapú módszerek fejlesztésére, kiértékelésére és összehasonlítására egyaránt. Eredetileg [35]-ben került bemutatásra a keretrendszer, amely számos előnyös tulajdonsága miatt azóta jelentősen elterjedt, ezek közül a legfontosabbakat kiemelve:

- *Szabványosság:* A Gym által nyújtott szabványosított interfésznek köszönhetően tetszőleges Gym-környezet alkalmazása esetén lehetővé válik a kutatók és egyéb felhasználók számára, hogy Megerősítési Tanulási algoritmusait hatékonyan, igazságszerűen, valamint könnyen reprodukálható módon értékeljék ki és hasonlítsák össze egymással. Az interfész következetes felépítése továbbá lehetővé teszi az algoritmusok könnyed alkalmazását különböző feladatokra (környezetekre), miközben biztosítja a megközelítések pontos összehasonlítását. Emellett fontos kiemelni, hogy az ágens és a környezet közötti interakció is szabványos módon zajlik, ennek következtében a környezettel való kommunikáció forráskódba való integrációja is hatékonyan elvégezhető.
- *Hozzáférhetőség:* A Gym keretrendszer egy mindenki számára ingyenesen hozzáférhető (open access) eszköz, amely a mindenre kiterjedően dokumentált API-val együtt járul hozzá a globális hozzáférhetőség egyszerűségéhez, lehetővé téve egy széles érdeklődői rétegnek – szakértelemtől és előzetes tapasztalatoktól függetlenül –, hogy felhasználják a környezetek által nyújtott funkciókat.
- *Flexibilitás, skálázhatóság:* A Gym környezetek struktúrája úgy van kialakítva, hogy az támogassa a hatékony és gyors utólagos testreszabását, illetve kibővítését egyes feladatoknak a felmerülő konkrét igényeknek és elképzeléseknek megfelelően. Ez a flexibilitás megkönnyíti a már meglévő, előre definiált környezetek módosítását, valamint teljesen új környezetek létrehozását és implementálását, ezáltal fellendítve a kutatás-fejlesztést az aktuális, innovatív megoldások terén.

A fent említett előnyök mellett a Gym csomag számtalan RL-alapfeladatot és problémát kínál a különböző megoldásokkal való kísérletezéshez, beleértve klasszikus irányítási feladatokat, robotikai szimulációkat és Atari-játékokat. Minden környezet egy úgynevezett Gym wrapper-be van foglalva, amellyel a továbbiakban jól definiált interfésszel lehet kapcsolatot teremteni. Ez teszi lehetővé az ágensek számára, hogy a környezet tetszőleges állapotából kiindulva gördülékenyen, a kiválasztott beavatkozásokon keresztül interakáljanak a környezettel, majd visszacsatolásként megkapják az állapotátmenet végállapotát, vagyis a következő állapotreprezentációt, illetve az állapotátmenet minőségét jelző jutalmat (reward). 3.1 az előzőekben részletezett Gym interfész sematikus ábráját mutatja be,



3.1. ábra. Gym interfész sematikus ábrája

ahol a központi komponens felelős mind a vezérlőparancsok (bezárás, inicializálás, visszaállítás és léptetés) kiadásáért, mind azok továbbításáért a tényleges környezet objektum felé, valamint a kommunikáló felek közötti adatátvitel megvalósításáért.

Annak érdekében, hogy problémák minél szélesebb körében vizsgáljuk meg az általunk bemutatott módszer teljesítményét, ezáltal diverzifikáljuk megoldásunk alkalmazhatósági területeit, négy alapjaiban különböző környezet került kiválasztásra a későbbi kiértékelés, majd pedig a részletes összehasonlítás fázisaihoz. A következőkben ezen környezetek rövid ismertetésére kerül sor.

3.1.1. Cartpole

A CartPole környezet egyike a Gym csomagban található, klasszikus irányításelméleti problémáknak, melyben a feladat egy kocsiszekrényhez kapcsolódó, instabil egyensúlyi helyzettel rendelkező rúd függőleges helyzetben tartása a kocsi egyenes vonalban való mozgatása által. Az eredetileg [36] által bemutatott specifikus RL-környezet definiálása az inverz ingán alapszik, amely az irányításelmélet egy alapvető benchmark feladatául szolgál különböző megközelítések tesztelésére. Az ágens célja a tanítási folyamat által nem más, mint hogy a rudat a függőleges helyzethez minél közelebb, a lehető leghosszabb ideig tudja megtartani.

A környezet állapotvektora ebben az esetben négy valós fizikai paraméterből áll: a kocsiszekrény pozíciója az egyenesvonalú pálya mentén x , a kocsi aktuális sebessége v , a rúd függőlegessel bezárt szöge θ és a rúd szögsebessége ω . A beavatkozási tér minden állapothoz két különböző beavatkozást definiál, amelyek rendre a kocsira ható konstans nagyságú erő irányát adják meg (a kocsi jobbra, illetve balra való elmozdítása).

Egy adott epizód akkor tekinthető sikeresnek, ha az ágens képes legalább 475 diszkrét időlépésig a limitációkon belül maradni, és csak ez után következik be végállapot. Az ágens

+1 jutalmat kap az epizód sikeres teljesítéséért, míg az ellenkező eset -1 értékű büntetést eredményez. Minden olyan időlépés, amely nem vezet a játék azonnali befejezéséhez, 0 reward értéket ér.

3.1.2. Acrobot

Egy másik gyakran alkalmazott környezetet mutattak be [37]-ben: az Acrobot szintén egy fizikai alapokon nyugvó szimuláció, amely az egyszerű kettős inga dinamikáját modellezi, a végtelenül merev rúdelemek között egy aktuált csuklóval. A cél ebben a környezetben nem más, mint a külső inga szabad végét, tehát a szélső tömegelemet, a lehető leggyorsabban fellendíteni egy előre meghatározott célmagasságba a középső csuklón elhelyezkedő aktuátoron keresztül, minden egyes időlépésben megfelelő irányú nyomatékot alkalmazva.

A megfigyelések a két rúdelem szögeinek szinusz és koszinusz értékeit tartalmazzák, amelyeket rendre $\sin \theta_1$, $\cos \theta_1$ és $\sin \theta_2$, $\cos \theta_2$ -vel jelölünk, valamint a rudak szögsebességeit ω_1 és ω_2 -vel jelölve (érdemes megjegyezni, hogy θ_1 a felső rúd függőlegessel bezárt szöge, míg θ_2 az alsó rúd relatív szöghelyzete a felső rúddhoz képest). A beavatkozások megfeleltethetők az aktuátor által kifejtett nyomatékoknak, amelyek a $\{-1; 0; 1\}$ Nm halmazból kerülnek kiválasztásra (lényegében egy konstans nagyságú nyomaték irányának megváltoztatására, valamint teljes megszüntetésére van lehetőség).

Az ágens minden egyes lépésért, amely nem vezet közvetlenül végállapothoz, -1 értékű negatív jutalomban részesül, teoretikusan ezzel ösztönözve a feladat minél dinamikusabb végrehajtására. Abban az esetben viszont, amikor a szabad vég eléri a célmagasságot, ezáltal sikeresnek tekinthető az adott epizód, 0 jutalomérték kerül kiosztásra.

3.1.3. Cliff Walking

A Cliff Walking elnevezésű grid-world (rácsvilág) környezet [38]-ból került kiemelésre, amelyben az ágensnek egy szakadékot is magába foglaló rács térképet szükséges felfedeznie azzal a céllal, hogy egy statikus kiindulási helyzetből, előre meghatározott célpozícióba navigáljon, miközben minimalizálja lépéseinek számát és természetesen elkerüli a szakadékba való leesést.

Az állapotrepresentáció ebben az esetben egyetlen skalár szám a $[0; 48)$ intervallumból, amely az ágens aktuális pozícióját jelöli a téglalap alakú 4×12 méretű rácsban, számítása 3.1 szerint történik:

$$s = curr_row * n_cols + curr_col \quad (3.1)$$

ahol $curr_row$ az aktuális sor indexét jelöli, n_cols az oszlopok száma (11, mivel az indexelés 0-tól kezdődik), míg $curr_col$ az aktuális oszlop indexe. Az ágens a térkép bal

alsó sarkában kijelölt kezdő pozícióból indul, a cél pedig a jobb alsó sarokban található. Az ágens a rácson belül a négy alapirányban képes beavatkozást végrehajtani: felfelé, jobbra, lefelé és balra. A rács legalsó sorában a kezdeti és végcellákon kívül szakadék található, amelyre történő elmozdulás azonnali büntetést, vagyis negatív jutalmat von maga után.

Ennek megfelelően általános esetben, minden egyes időlépésnél az ágens -1 jutalomértéket realizál, kivétel ez alól az előbb ismertetett állapot, amely -100 értéket eredményez. Egy epizód akkor ér véget, ha az ágens eléri a térképen lévő cél cellát, vagy egy előre definiált időlimitet.

3.1.4. Taxi

A negyedik és egyben utolsó, itt részletezésre kerülő környezet a Taxi, amelyet [39]-ben mutattak be először: egy taxi-üzem egyszerűsített szimulációja jelenik meg szintén egy grid-world keretén belül. Az ágens elsődleges célja az utasok elszállítása a kijelölt felvételi pontokból a megfelelő desztináció celláig, az utazási idő (a szimulációban elmozdulások) minimalizálása mellett.

A térkép egy 5×5 -ös rácsként van reprezentálva, lehetővé téve az ágens számára, hogy a 25 cella bármelyikét elfoglalhassa. Minden epizódban négy utas generálódik, ezáltal öt lehetséges allokáció jelenik meg általuk, beleértve egyet, amikor egyikük a járműben ül. Ezenkívül négy célállomás kerül előre definiálásra, így összesen 500 különböző állapotról beszélhetünk, amelyből a 3.2 egyenlettel adott az egyetlen skalár állapotjelző kiszámítási módja, amely végül az ágenshez jut:

$$s = ((t_r * 5 + t_c) * 5 + p) * 4 + d \quad (3.2)$$

ahol s az állapot skalár reprezentációja, t_r és t_c az ágens aktuális pozíciójának sor-, illetve oszlop indexét jelölik, p az utas helyzetét, d pedig a célállomás helyét jelöli. A beavatkozási tér hat elemből áll: felfelé, lefelé, balra és jobbra történő elmozdulás, valamint a fel- és leszállás végrehajtása.

Az ágens $+20$ jutalomban részesül, ha sikeresen végrehajtja az utas elszállítását a megfelelő helyre, de -10 értékű büntetést kap abban az esetben, ha a fel- vagy a leszállást nem a jó cellában hajtja végre. Továbbá az epizód során minden egyes időlépésnél egy -1 értékű büntetés kerül kiosztásra, amely a feladat elvégzésének másodlagos célját, az időhatékonyságot segíti elő.

3.2. Tanítás

A tanítási folyamat rendkívül meghatározó szerepet játszik Mély Tanulás alkalmazásakor, mivel az ágens teljesítménye a tanítás során fejlődik fokozatosan, ámde folyamatosan

a végső szintjére. Annak érdekében, hogy egy megfelelően diverz tanító adathalmaz kerüljön létrehozásra a környezettel való interakció folyamán, kulcsfontosságú a megfelelő fokú véletlenszerűsítés biztosítása. A kezdeti állapot minden környezet esetében, kivéve a Cliff Walking-ot, sztenderd módon kerül random meghatározásra a rendelkezésre álló érvényes kezdőpozíciók közül, a Gym keretrendszer által. Az így kialakított kezdő pozíciók által lehetővé válik egy jóval sokszínűbb tanítóminta halmaz, amely segíti az ágens robusztusságának kialakítását. Ezenkívül a végállapotok, kilépési feltételek pontos meghatározása is nélkülözhetetlen a vizsgált problémák megfelelő megoldásához, bár ez a Gym keretrendszer által szabványosított formában kerül alkalmazásra.

A később bemutatásra kerülő eredmények alapját képező kísérletek kettő megközelítés, a Prioritized Experience Replay-t és az exploration-exploitation dilemmát feloldó, UCB-alapú priorizációs algoritmust alkalmazó ágensek tanítása révén kerültek elvégzésre. Hipotézisünk további validálása érdekében mindkét módszerrel környezetenként 5-5 tanítást hajtottunk végre azonos feltételek mellett, a környezetekben alkalmazott determinisztikus pszeudo-véletlenszerű generátor seed-elésével.

Az első szakaszban szimpla tanítások kerültek elvégzésre a négy különböző környezeten, ugyanazon hiperparaméter halmazokkal és Neurális Hálózati architektúrákkal a két algoritmus esetén. Ezen tanításokra irányuló elemzések célja az volt, hogy általános performancia-indikátorok segítségével kerüljenek összehasonlításra a vizsgált algoritmusok, vagyis az általunk bemutatott módszer teljesítménye a jelenlegi state-of-the-art algoritmussal szemben.

A második szakaszban az előzőektől eltérően egy bemelegítési szakasz előzi meg az epizódokat, mielőtt a Neurális Hálózatok súlyainak frissítése megtörténne. Ebben a szakaszban a tanítóminta visszajátszási memóriák azonos állapotátmenet sorokkal kerülnek feltöltésre mindkét megközelítés esetében, amelyek a tanítás során változatlanok maradnak. Ennek az a célja, hogy a két mintavételezési stratégia hatékonyságbeli különbségeit kiemelve tovább erősítsük teoretikus állításainkat, mivel ugyanazon memóriából egyedül a mintavételezés milyenségétől függ a tanítás adott időpontban vett állapota, illetve a tanított ágens teljesítménye.

3.3. Kiértékelés

Az előzőekben ismertetett komplex tanítási folyamatokból kinyert nyers mérési adatok, monitorozott metrikák részletes kiértékelésnek lettek alávetve, miközben figyelemmel kísértük számos különböző tanítást leíró és az ágens aktuális teljesítményét minősítő jelző alakulását, különös hangsúlyt fektetve az alkalmazott mintavételezési stratégiák hatékonyságára. Ezen mutatók kiterjednek, de nem kizárólagosak a következő valamennyi metrikára:

- Időlépések száma valamely végállapot eléréséig adott epizódon belül
- Átlagos jutalom (reward) értéke adott epizódra nézve
- Kumulált jutalom az epizód valamennyi időlépésében szerzett reward értékekre
- Epizód abszolút tanítási ideje (GPU: NVIDIA Titan RTX 900-1G150-2500-000, CPU: Intel(R) Xeon(R) W-2123 CPU @ 3.60GHz, RAM: 64.0 GB)

Mivel tetszőleges tanítás abszolút ideje erősen korrelál a tanítás erőforrás szükségletével és a számítási kapacitásigénnyel, várhatóan mindhárom érték (habár közülük csak az egyik került megfigyelésre) hasonló arányban képes megváltozni az egyes tanítások között.

A tanítóminta priorizálási módszerek részletes értékelése és a különböző megoldások még alaposabb vizsgálata érdekében a második tanítási fázisban különböző kapacitású fix memóriák segítségével végeztük el az elemzést, ahol a bufferek mérete, így az ágens számára rendelkezésre álló statikus tanítóminták számossága, 10.000, 30.000, 50.000 és 100.000 darabra lettek konfigurálva.

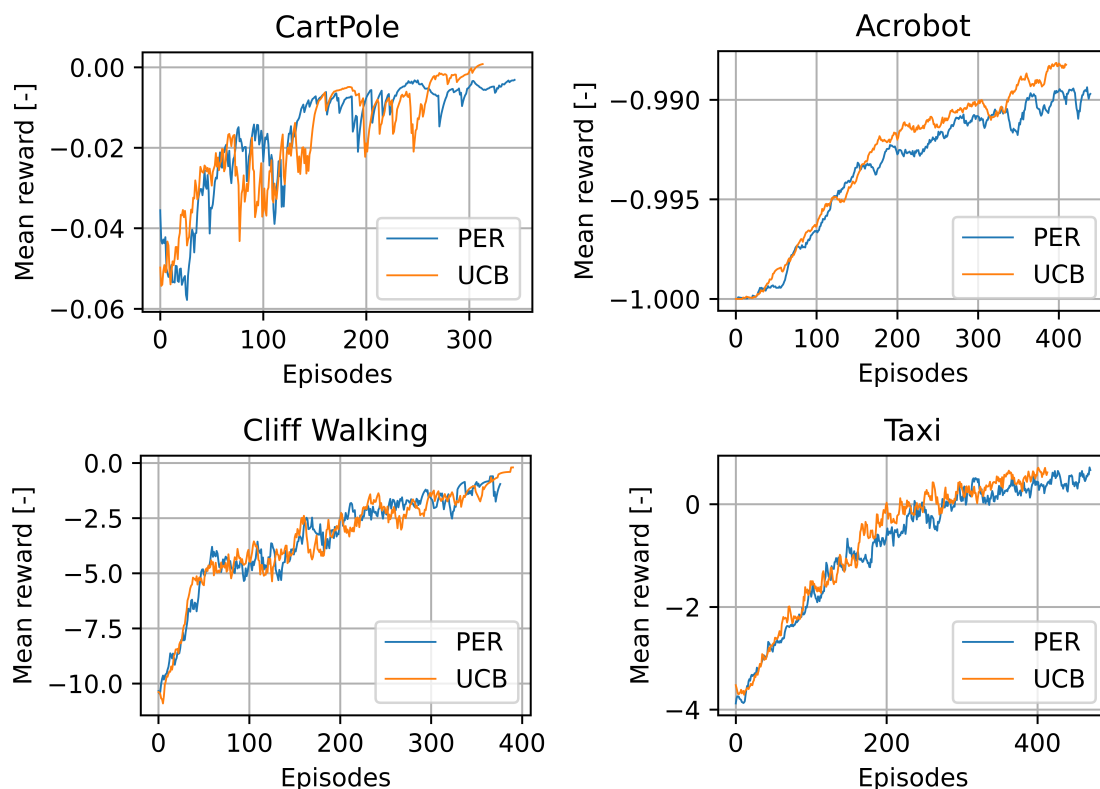
4. fejezet

Eredmények

4.1. Első fázis – általános kiértékelés

Ebben a fejezetben a feldolgozott tanítóminták részletes elemzésére kerül sor a Prioritized Experience Replay és a bemutatott Upper Confidence Bound-alapú priorizálási módszer tekintetében, amely során figyelembe vesszük és kiértékeljük az általánosan vizsgált tanító metrikák alakulását.

Az epizódonként gyűjtött átlagos reward értékek eredményei a 4.1 ábrán láthatóak

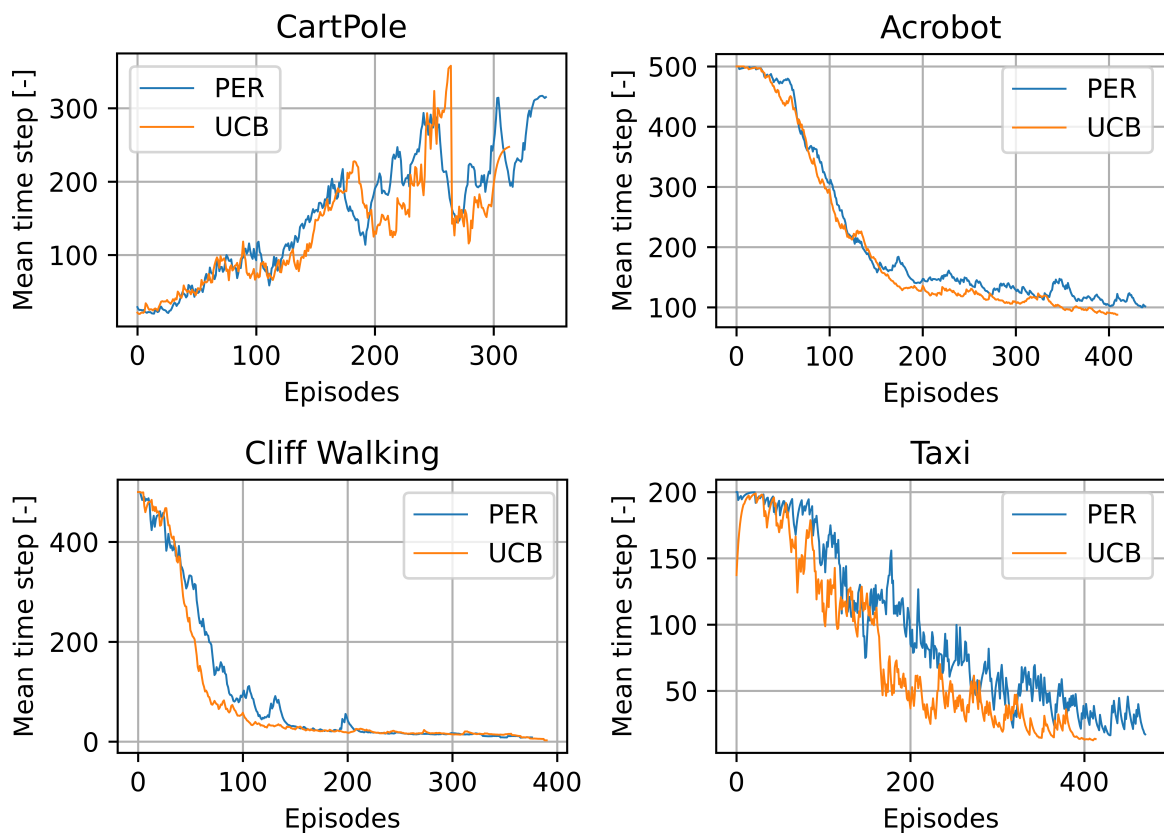


4.1. ábra. Epizódonként gyűjtött átlagos reward értékek

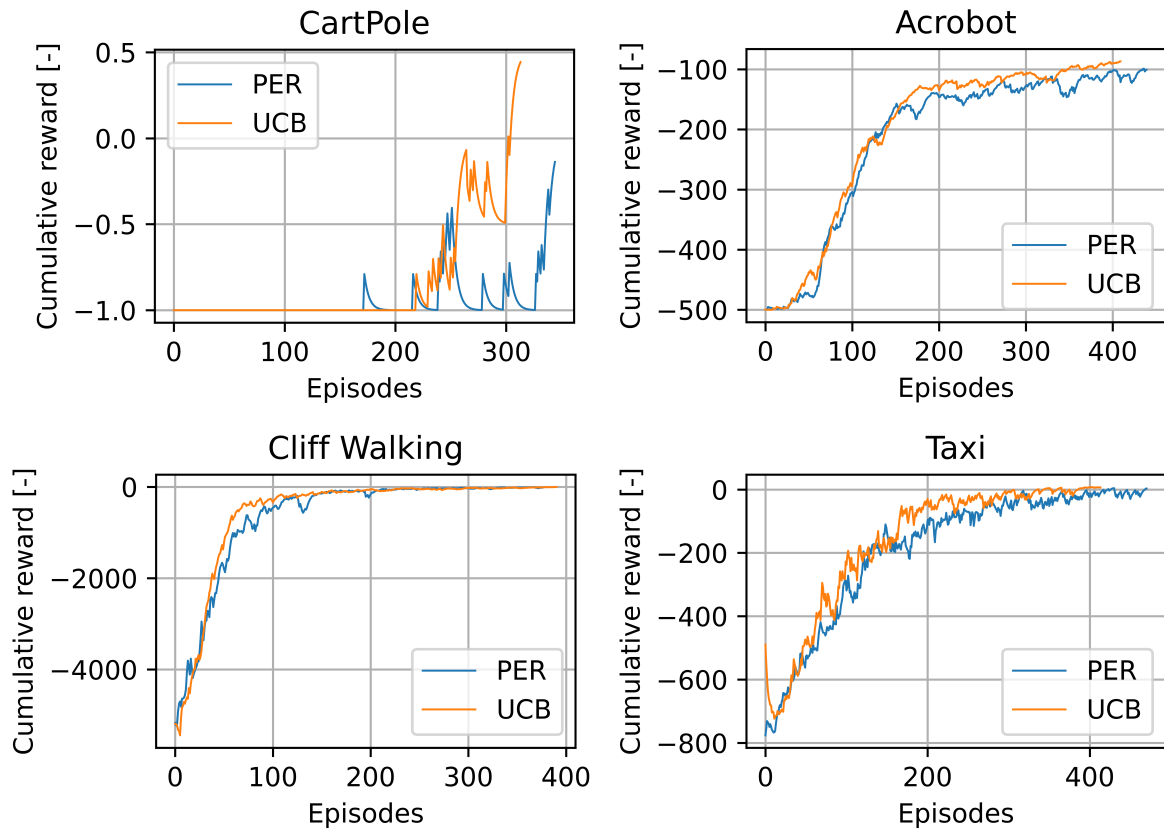
mind a négy vizsgált környezetre vonatkozóan, az 5 pszeudo-véletlenszerűen generált epizód futásai során. A feltüntetett értékek átláthatósága érdekében egyszerű aluláteresztő simító szűrő került alkalmazásra a megjelenítésnél.

Az ábrán a továbbfejlesztett prioritáserérték-hozzárendelési stratégiát alkalmazó ágens narancssárga színnel, míg a jelenleg legszélesebb körben használt tanítóminta prioritizálási algoritmus által tanított ágens kék színnel látható. Ezek az eredmények, valamint a 4.2 ábrán látható epizódonkénti átlagos időlépések száma és a 4.3 ábrán szemléltetett epizódonkénti kumulatív reward értékek által megjelenített konvergencia görbék egyenlő mértékben bizonyítják megközelítésünk egyértelmű teljesítményfölényét az összehasonlított koncepcióval szemben a mintavételezési hatékonyság tekintetében. Nemcsak hogy kisebb amplitúdójú és frekvenciájú oszcillációk figyelhetők meg a tanítás során, de az epizódok száma és ezáltal az adott feladat megtanulásához szükséges abszolút időszükséglet is jelentős mértékben lecsökkent.

A feladat sikeres végrehajtásáig szükséges Neurális Hálózati frissítések számának mérséklése egyfelől a konvergencia sebességének növekedését, másfelől az abszolút tanítási idő jelentős csökkenését eredményezi az újonnan bemutatott módszerrel, a PER algoritmus-hoz képest. Ennek a kijelentésnek a numerikus reprezentációját mutatja be a 4.1 táblázat,



4.2. ábra. Epizódonként futtatott időlépések száma



4.3. ábra. Epizódonként gyűjtött kumulált reward értékek

ahol a vizsgált probléma komplexitásától függően figyelemre méltó, 10 – 30%-os javulás figyelhető meg. Zárójelben feltüntetésre kerültek továbbá az átlagértékekhez kapcsolódó szórások, amelyek szintén jelzik az UCB-alapú prioritizálással elérhető jóval stabilabb tanulási folyamatokat. Hasonlóképpen, a 4.2 táblázat az átlagos epizódszámot mutatja, amely az adott feladat sikeres végrehajtásához szükséges az ágens tanítása szempontjából. A legtöbb esetben látható, hogy az általunk ismert algoritmus magasabb teljesítményt képes elérni ezen értékeket tekintve, kivéve a Cliff Walking környezetet, amelyen valószínűsíthetően a feladat trivialitása miatt az algoritmus nem képes kontrasztot képezni.

4.1. táblázat. Abszolút tanítási időszükségletek számtani közepének [s] és szórás értékek statisztikai összehasonlítása

Prioritization method	CartPole	Acrobot	Cliff Walking	Taxi
PER	218.6(±66.14)	704.2(±116.21)	140.4(±49.25)	4866.4(±2077.81)
UCB	154.4(±30.96)	493.6(±55.79)	129.4(±38.67)	4415.2(±1441.34)
Difference of UCB relative to PER	−29.4%	−29.9%	−7.8%	−9.3%

4.2. táblázat. Adott feladat megtanulásához szükséges epizódszámok számtani közepének statisztikai összehasonlítása

Prioritization method	CartPole	Acrobot	Cliff Walking	Taxi
PER	380.6	425.0	352.0	542.4
UCB	305	405.8	382.4	418.6
Difference of UCB relative to PER	−19.9%	−4.5%	+8.6%	−22.8%

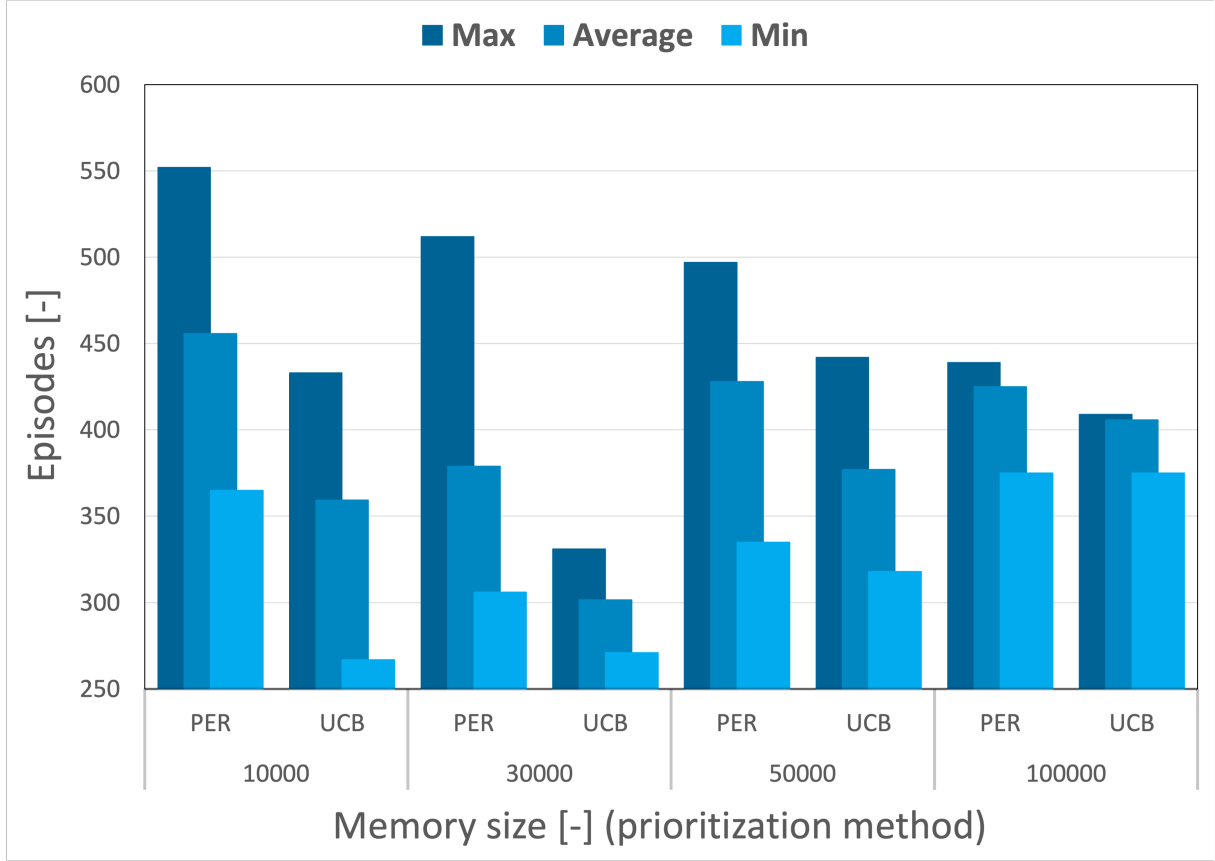
Ettől eltekintve azonban, csupán az abszolút időfelhasználást figyelembe véve ezen a környezeten is jobb eredményeket képes elérni az ebben a dolgozatban bemutatott módszer.

Következésképpen, mivel az abszolút tanítási idő erős korrelációban áll a számítási kapacitásigénnyel, ezen eredmények alapján arra lehet következtetni, hogy a szükséges számítási teljesítmény várhatóan hasonló arányban csökkenthető. Amint az a bevezetésben 1 részletezésre került, a tanítások teljes költsége kiszámítható a tanítás idejéből és számítási igényéből eredő indirekt költségek összegeként, ezáltal a megnövelt mintavételi hatékonysággal jelentős előrelépés érhető el.

4.2. Második fázis – statikus visszajátszási memória

Annak érdekében, hogy még inkább alá tudjuk támasztani elméletünket a továbbfejlesztett tanítóminta prioritizálási megközelítés magasabb mintavételezési performanciájáról, egy kollektív kiegészítő kísérletet végeztünk, amely során a két megoldás azonos állapotátmenetekkel rendelkező, azonos módon feltöltött, statikus tanítóminta visszajátszási memóriákkal rendelkezik. Ezt egy, a tanítást megelőző, úgynevezett bemelegítő fázison keresztül érjük el, amely hatására a tanítások időbeli kiterjedését egyetlen tényező befolyásolja: a mintavételi hatékonyság. Abban az esetben, ha az ágens jobban fel tudja használni az egyes memóriákban tárolt egymással megegyező tanító adatokat, vagyis nagyobb hatékonysággal mintavételez a memória bufferből, akkor kevesebb hálózati frissítésre és így kevesebb tanító epizódra van szükség az azonos szintű teljesítmény eléréséhez.

A 4.4 ábrán 5 – 5 véletlenszerűen generált tanításhoz tartozó adat került megjelenítésre az Acrobot elnevezésű környezeten, miközben egymással megegyező tanítómintákat tartalmazó, statikus memóriák állnak mindkét ágens rendelkezésére. Az egyes oszlopokban különböző színekkel a sikeres tanításhoz szükséges minimum-, maximum- és átlag epizódszám-értékek szerepelnek (ebben az esetben sikeres tanításnak az elmúlt 10 epizódra számított csúszoátlag számít, amennyiben az eléri a 100 alatti időlépés számot). Jól



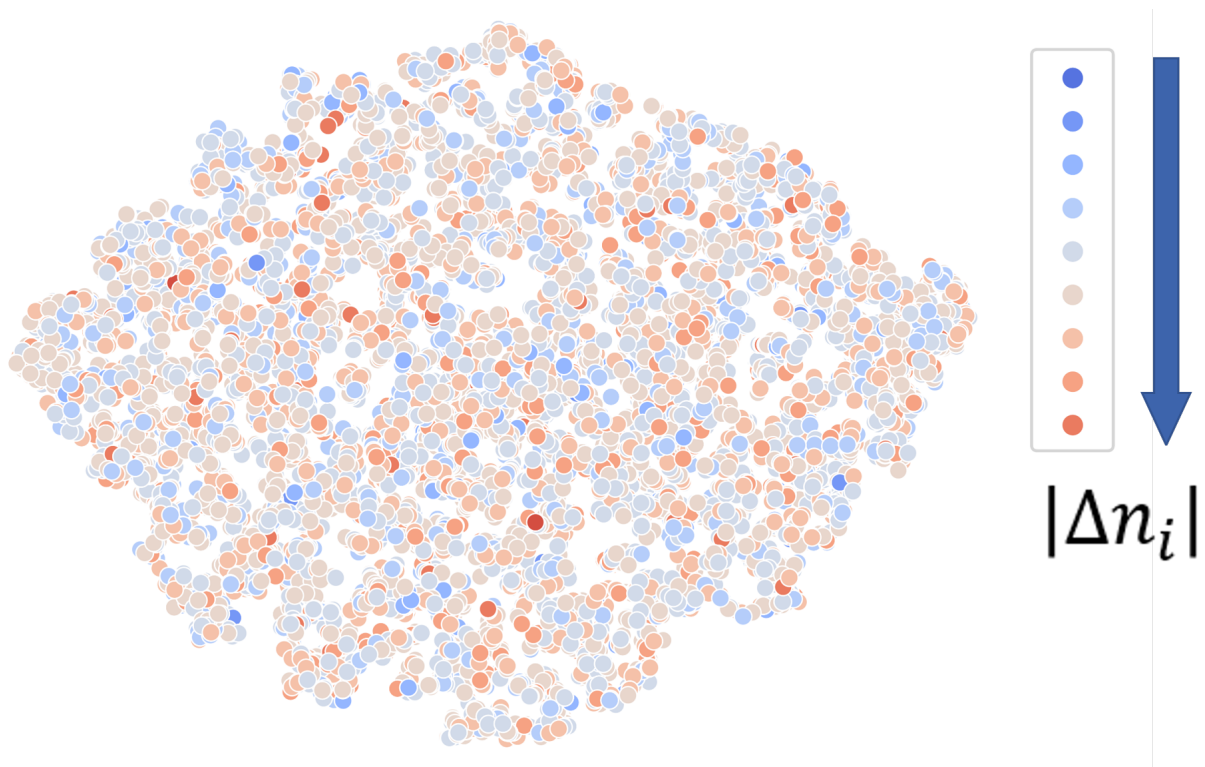
4.4. ábra. Azonos méretű, statikus visszajátszási memóriákat alkalmazó ágensek tanítási statisztikáinak összehasonlítása az Acrobot környezeten

látható az összes statisztikai jellemzőből, hogy a memória méretkonfigurációjától függetlenül, minden esetben az UCB-alapú priorizálási algoritmusunknak sikerült jobb eredményt elérnie, egy hatékonyabb mintavételi stratégia megvalósításával.

A 4.5 ábra további betekintést nyújt a megoldásunk által elért nagyobb teljesítmény részleteibe. Az absztrakt alakzat egyes pontjai tapasztalatokat jelölnek, míg a színezés kéktől a piros felé haladva az egyre növekvő, frissítésszámok közötti abszolút különbséget jelzi. Ez a különbség-metrika minden egyes tanítóminta esetében kiszámításra kerül, miután az ágens sikeresen megtanulta abszolválni az adott feladatot, ahogyan az 4.1 egyenletben definiálásra kerül:

$$RGB = |\Delta n_i| = |n_{i, PER} - n_{i, UCT}| \quad (4.1)$$

ahol n_i az i -dik állapotátmenethez tartozó frissítésszám. Ezáltal a kék színű pontokból a piros színűek felé elmozdulva a metrika kiemeli a két algoritmus között növekvő fókuszeltolódást a tapasztalatok tekintetében, a t-Distributed Stochastic Neighbor Embedding (t-SNE) dimenzióredukciós módszerrel, amelyet szintén az Acrobot feladaton való tanítás után a 10.000 memóriaméretű tanítóminta-visszajátszási memória kezdeti állapotaira alkalmaztunk.



4.5. ábra. Tanítóminták kezdeti állapotainak kétdimenziós t-SNE embedding-je a tanítás végén Acrobot környezetben

5. fejezet

Konklúzió

A Megerősítéses Tanulás továbbra is egy kihívásokkal teli, ugyanakkor rendkívül ígéretes terület marad, különösen szekvenciális döntéshozatali problémák kapcsán. Ez az állítás abból eredeztethető, hogy a betanított RL ágensek kiemelkedő performanciát képesek nyújtani más, potenciálisan alkalmazható módszerekhez képest. Ezek egyike a Felügyelt Tanulás, ahol az ágens teljesítménye a felhasznált tanító adathalmaz által korlátozódik, illetve egy másik alternatíva lehet különböző klasszikus algoritmusok alkalmazása, amelyek egyaránt jelentős számítási teljesítményt igényelnek az online döntéshozatali folyamatban, ezáltal nem teszik lehetővé a korlátozás nélküli valós idejű alkalmazhatóságot.

Azonban a Deep Learning-alapú megoldások – ezáltal természetesen a Mély Neurális Hálózatokat alkalmazó RL-algoritmusok – szignifikáns hátránya, hogy a tanítási fázis a tanítóadatok nem megfelelő kezelése miatt túlságosan nagy erőforrás-felhasználással jár. Az úgynevezett mintavételezési hatékonyság ilyen szempontból egy kulcsfontosságú metrikaként jelenik meg, amely mentén a fent említett módszerek teljesítményének értékelése megvalósítható. Számos különböző koncepciót publikáltak már ennek a problémának a kezelésére, amelyek közül egy a tanítóminta priorizálás, amelynek segítségével a mintavételezés hatékonysága azáltal javítható, hogy egy memóriában eltárolt valamennyi állapotátmenethez (tanítómintához) meghatározásra és hozzárendelésre kerül egy prioritás érték. Ezen értékek képezik a sztochasztikus mintavételezés alapjául szolgáló valószínűségi eloszlást, ezáltal egyfajta dinamikus súlyozást megvalósítva az egyes tanítóminták között a tanítás folyamán.

Ebben a dolgozatban egy újszerű megközelítés került bemutatásra, amely teljes mértékben újraértelmezi az exploration-exploitation dilemmát egy új prioritásérték-hozzárendelési koncepción keresztül. Az egyes minták prioritása az Upper Confidence Bound algoritmus matematikai formalitása alapján került definiálásra, amelyben a kevésbé felfedezett tapasztalatokat értékelő komponens (exploration) alapjául a frissítésszám szolgál. Ez az érték leírja, hogy az adott tapasztalat hányszor lett felhasználva a Neurális Hálózat hangolása céljából, ezáltal értékeli az aktuális állapotban az adott tanítómintáról meglévő információ bizonytalanságát. A másik tag, amely a pillanatnyilag legjobbnak várható

döntéseket részesíti előnyben (exploitation) továbbra is a normalizált TD-hiba marad. Ezáltal egy új trade-off stratégia került bevezetésre a Megerősítéses Tanulás területére, amely a mintavételezés hatékonyságának problémájára hivatott megoldást szolgáltatni.

A kutatás eredményei kimutatták, hogy az általunk bemutatott módszer jelentős mértékben javítja a mintavételezés hatékonyságát és növeli a tanulás stabilitását a jelenleg legelterjedtebben alkalmazott, Prioritized Experience Replay algoritmushoz képest, ahogy az az egyes konvergenciagörbék statisztikai elemzése során is megerősítésre került. Emellett kimondható, hogy ezen kiemelkedő mintavételi stratégia használatával az abszolút tanítási idő és a számítási erőforrásigény egyaránt szignifikánsan csökkenthető, ami hozzájárul a Megerősítéses Tanulás-alapú ágensek tanítási költségeinek csökkentéséhez.

Jövőbeli kutatási terveink közé tartozik ennek a tanítóminta priorizálási metodológiának az alkalmazása komplexebb, valós feladatokra annak érdekében, hogy demonstrálásra kerüljön képessége, amely lehetővé teszi az állapotátmenetek információinak hatékonyabb felhasználását. Másodszor, további célunk, hogy megvizsgáljuk és potenciálisan tovább finomítsuk az exploration-exploitation dilemma által definiált komponenseket, illetve még kifinomultabb absztrakciókat keressünk mind az exploration, mind pedig az exploitation tagnak a Megerősítéses Tanulás keretrendszerén belül. Végezetül pedig hiszünk abban, hogy egy hasonló trade-off elmélet alkalmazása a Felügyelt Tanulás területén is figyelemre méltó eredményeket hozhat, mivel a tanító adatok nem-egyenértékűségének problémája a Gépi Tanulás ezen ágán is teljes mértékben releváns, amely az általunk bemutatott megközelítést felhasználva egy megfelelő egyensúly kidolgozásával kezelhető lehet.

Irodalom

- [1] Daekyum Kim és tsai. „Review of machine learning methods in soft robotics”. *Plos one* 16.2 (2021), e0246102.
- [2] Bo-hu Li és tsai. „Applications of artificial intelligence in intelligent manufacturing: a review”. *Frontiers of Information Technology & Electronic Engineering* 18 (2017), 86–96. old.
- [3] Piotr Gorzelanczyk. „Forecasting the Number of Road Accidents in Poland Depending on the Day of the Week Using Neural Networks”. *Periodica Polytechnica Transportation Engineering* 51.4 (2023), 431–435. old. DOI: 10.3311/PPtr.22203.
- [4] Swati Garg és tsai. „A review of machine learning applications in human resource management”. *International Journal of Productivity and Performance Management* 71.5 (2022), 1590–1610. old.
- [5] Bálint Kóvári, Ferenc Hegedüs és Tamás Bécsi. „Design of a reinforcement learning-based lane keeping planning agent for automated vehicles”. *Applied Sciences* 10.20 (2020), 7171. old.
- [6] Jaime Sevilla és tsai. „Compute trends across three eras of machine learning”. *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2022, 1–8. old.
- [7] Ben Cottier. *Trends in the dollar training cost of machine learning systems*. Accessed: 2023-6-14. 2023. URL: <https://epochai.org/blog/trends-in-the-dollar-training-cost-of-machine-learning-systems>.
- [8] Tyler L Hayes és tsai. „Replay in deep learning: Current approaches and missing biological elements”. *Neural computation* 33.11 (2021), 2908–2950. old.
- [9] Andrew W Moore és Christopher G Atkeson. „Prioritized sweeping: Reinforcement learning with less data and less time”. *Machine learning* 13 (1993), 103–130. old.
- [10] Volodymyr Mnih és tsai. „Human-level control through deep reinforcement learning”. *nature* 518.7540 (2015), 529–533. old.
- [11] Tom Schaul és tsai. „Prioritized experience replay”. *arXiv preprint arXiv:1511.05952* (2015).

- [12] Yuenan Hou és tsai. „A novel DDPG method with prioritized experience replay”. *2017 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE. 2017, 316–321. old.
- [13] Dan Horgan és tsai. „Distributed prioritized experience replay”. *arXiv preprint arXiv:1803.00933* (2018).
- [14] Marc Brittain és tsai. „Prioritized sequence experience replay”. *arXiv preprint arXiv:1905.12726* (2019).
- [15] Pei-Hao Su és tsai. „Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management”. *arXiv preprint arXiv:1707.00130* (2017).
- [16] Jacob Buckman és tsai. „Sample-efficient reinforcement learning with stochastic ensemble value expansion”. *Advances in neural information processing systems* 31 (2018).
- [17] Yang Yu. „Towards Sample Efficient Reinforcement Learning.” *IJCAI*. 2018, 5739–5743. old.
- [18] Rui Zhao és Volker Tresp. „Energy-based hindsight experience prioritization”. *Conference on Robot Learning*. PMLR. 2018, 113–122. old.
- [19] Rui Zhao és Volker Tresp. „Curiosity-driven experience prioritization via density estimation”. *arXiv preprint arXiv:1902.08039* (2019).
- [20] Zhang-Wei Hong és tsai. „Topological experience replay”. *arXiv preprint arXiv:2203.15845* (2022).
- [21] Shivakanth Sujit és tsai. „Prioritizing samples in reinforcement learning with reducible loss”. *arXiv preprint arXiv:2208.10483* (2022).
- [22] Leslie Pack Kaelbling, Michael L Littman és Andrew W Moore. „Reinforcement learning: A survey”. *Journal of artificial intelligence research* 4 (1996), 237–285. old.
- [23] Songsang Koh és tsai. „Real-time deep reinforcement learning based vehicle navigation”. *Applied Soft Computing* 96 (2020), 106694. old.
- [24] Chao Yan, Xiaojia Xiang és Chang Wang. „Towards real-time path planning through deep reinforcement learning for a UAV in dynamic environments”. *Journal of Intelligent & Robotic Systems* 98 (2020), 297–309. old.
- [25] Joel Z Leibo és tsai. „Scalable evaluation of multi-agent reinforcement learning with melting pot”. *International conference on machine learning*. PMLR. 2021, 6187–6199. old.
- [26] Ming Zhou és tsai. „Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving”. *arXiv preprint arXiv:2010.09776* (2020).

- [27] Tianshu Chu és tsai. „Multi-agent deep reinforcement learning for large-scale traffic signal control”. *IEEE Transactions on Intelligent Transportation Systems* 21.3 (2019), 1086–1095. old.
- [28] Zhuangdi Zhu és tsai. „Transfer learning in deep reinforcement learning: A survey”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).
- [29] Martijn Van Otterlo és Marco Wiering. „Reinforcement learning and markov decision processes”. *Reinforcement learning: State-of-the-art*. Springer, 2012, 3–42. old.
- [30] Christopher JCH Watkins és Peter Dayan. „Q-learning”. *Machine learning* 8 (1992), 279–292. old.
- [31] Pengyu Zhai, Yingjun Zhang és Wang Shaobo. „Intelligent ship collision avoidance algorithm based on DDQN with prioritized experience replay under COLREGs”. *Journal of Marine Science and Engineering* 10.5 (2022), 585. old.
- [32] Daniel Fährmann és tsai. „Double deep q-learning with prioritized experience replay for anomaly detection in smart environments”. *IEEE Access* 10 (2022), 60836–60848. old.
- [33] Roger McFarlane. „A survey of exploration strategies in reinforcement learning”. *McGill University* 3 (2018), 17–18. old.
- [34] Peter Auer. „Using confidence bounds for exploitation-exploration trade-offs”. *Journal of Machine Learning Research* 3.Nov (2002), 397–422. old.
- [35] Greg Brockman és tsai. *OpenAI Gym*. 2016. eprint: [arXiv:1606.01540](https://arxiv.org/abs/1606.01540).
- [36] Andrew G Barto, Richard S Sutton és Charles W Anderson. „Neuronlike adaptive elements that can solve difficult learning control problems”. *IEEE transactions on systems, man, and cybernetics* 5 (1983), 834–846. old.
- [37] Richard S Sutton. „Generalization in reinforcement learning: Successful examples using sparse coarse coding”. *Advances in neural information processing systems* 8 (1995).
- [38] Richard S Sutton és Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [39] Thomas G Dietterich. „Hierarchical reinforcement learning with the MAXQ value function decomposition”. *Journal of artificial intelligence research* 13 (2000), 227–303. old.

Ábrák jegyzéke

1.1. State-of-the-art Mély Tanulási modellek inflációkorrigált tanítási költségei <i>USD</i> -ben kifejezve [7]	2
1.2. Priorizált tanítóminta visszajátszási algoritmus teljesítményét bemutató tanítási görbék [11]	4
1.3. Elosztott ágens koncepció sematikus rajza a Distributed Prioritized Experience Replay algoritmusban [13]	5
2.1. Megerősítéses Tanulás-alapú ágens tanítását reprezentáló interakciós folyamat	14
2.2. Deep-Q Network tanítási folyamata kiemelve az algoritmus két fő újítását .	18
3.1. Gym interfész sematikus ábrája	25
4.1. Epizódonként gyűjtött átlagos reward értékek	30
4.2. Epizódonként futtatott időlépések száma	31
4.3. Epizódonként gyűjtött kumulált reward értékek	32
4.4. Azonos méretű, statikus visszajátszási memóriákat alkalmazó ágensek tanítási statisztikáinak összehasonlítása az Acrobot környezeten	34
4.5. Tanítóminták kezdeti állapotainak kétdimenziós t-SNE embedding-je a tanítás végén Acrobot környezeten	35

Táblázatok jegyzéke

4.1. Abszolút tanítási időszükségletek számtani közepének [s] és szórás értékek statisztikai összehasonlítása	32
4.2. Adott feladat megtanulásához szükséges epizódszámok számtani közepének statisztikai összehasonlítása	33