Primavera 2024 Pràctica 2 (Laboratori)

SISTEMES INTEL.LIGENTS DISTRIBUÏTS

UNIVERSITAT POLITÈCNICA DE CATALUNYA

DEPARTAMENT DE CIÈNCIES DE LA COMPUTACIÓ

Martí Recalde

Arnau Esteban

Oriol Ramos

ÍNDEX

1. Introducció

Durant les sessions de laboratori sobre aprenentatge per reforç, hem vist possibles maneres d'abordar l'entorn FrozenLake-v11 [1]: Iteració de valor, Estimació directa (model-based), Q-Learning i REINFORCE en un entorn amb un MDP accessible i amb espais discrets d'observacions i accions. En aquesta pràctica explorarem com entrenar un agent en un entorn de característiques similars.

L'objectiu d'aquesta pràctica és que dissenyem i executem experiments per a realitzar un estudi del rendiment en l'entorn Taxi-v3 [2] dels següents algorismes:

- Iteració de valor
- Estimació directa
- Q-Learning
- REINFORCE (extra)

2. Iteració de valor

2.1 Introducció

En primer lloc estudiarem el rendiment del mètode d'iteració del valor en el context de l'entorn Taxi-v3. S'analitzarà la variabilitat d'aquest en virtud de les modificacions realitzables als diferents paràmetres rellevants per l'algorisme. Per tal d'exemplificar aquesta variabilitat s'acompanyarà la breu descripció de l'algorisme que segueix amb un seguit d'experiments on s'explora aquesta variabilitat.

2.2. Paràmetres importants

1. Nombre d'episodis (NUM_EPISODES):

Aquest paràmetre determina quantes vegades s'executarà l'agent en l'entorn per recopilar dades i millorar la seva política. Un major nombre d'episodis permet a l'agent explorar més l'entorn i obtenir una millor estimació dels valors esperats de les accions en cada estat. En aquest estudi, es consideren 10 episodis per avaluar el rendiment inicial de l'agent.

2. Factor de descompte (GAMMA):

Aquest paràmetre (gamma) determina la importància de les recompenses futures en comparació amb les recompenses immediates. Un valor proper a 1 significa que es consideren fortament les recompenses futures, mentre que un valor proper a 0 fa que l'agent es concentri més en les recompenses immediates. En aquest cas, s'utilitza un factor de descompte de 0.95.

3. Senyal de recompensa (REWARD_THRESHOLD):

La funció de recompensa defineix com de bo o dolent és un estat o una acció per a l'agent. És crucial per guiar l'aprenentatge de l'agent cap a comportaments desitjats. En aquest context, un llindar de recompensa de 10 s'utilitza per definir què es considera una recompensa acceptable.

4. Nombre de trajectòries(T_MAX i num_trajectories):

En el context de l'estimació directa, el nombre de trajectòries o episodis és essencial perquè defineix la quantitat de dades d'experiència que l'agent utilitzarà per estimar els valors esperats de les accions en cada estat. En aquest estudi, es considera un màxim de 15 trajectòries per avaluar les estimacions.

2.3 Iteració del valor: Explicació general

Aquest és un algorisme d'aprenentatge per reforç basat en model que requereix d'una modelització del problema en la forma d'un Markov decission problem (MDP). A partir d'aquesta, l'algorisme d'iteració del valor cerca optimitzar la seva política seguint equacions

que estimen el valor dels estats a què es pot arribar des de l'actual a través de les diferents accions disponibles. A continuació en detallem alguns avantatges i inconvenients:

2.3.1. Avantatges

- Simplicitat: És fàcil d'implementar i entendre.
- Efectivitat en problemes petits: Per problemes petits que es puguin formalitzar com un MDP aquest algorisme té quasi sempre un bon rendiment degut a les seves condicions d'optimalitat.

2.3.2. Inconvenients

- Escalabilitat: A mesura que l'entorn creix en estats i accions disponibles, la quantitat de càlculs necessaris per usar aquest algorisme escala exponencialment de manera que en entorns grans, malgrat es puguin formalitzar com a MDPs, troba dificultats per rendir adequadament.
- Inestabilitat: Si diferents execucions poden generar conjunts d'estats diferents, l'algorisme troba dificultats per optimitzar la política generada. Això es deu principalment a les seves dificultats per generalitzar el problema a resoldre derivades, almenys en part, del fet de basar-se en una modelització rígida com la pròpia dels MDPs.

2.4 Iteració del valor a Taxi-v3

Iteració del valor troba algunes dificultats en el entorn Taxi-v3. Aquestes es deuen principalment als inconvenients associats a aquest algorisme que hem definit tot just fa un moment. Taxi-v3 pot ser modelitzat com un MDP però el seu espai d'estats i accions és considerablement gran. Tot i així, allò que més perjudica iteració del valor en aquest entorn és el fet que, a cada execució, la circumstància a resoldre canvia. A diferència de l'entorn FrozenLake que vam veure a les classes de laboratori, a Taxi-v3 unes vegades l'objectiu està en un lloc i unes altres en un altre. Així, malgrat al laboratori iteració del valor convergia ràpidament i fàcil a la política òptima, aquí li resulta quasi bé impossible d'esbrinar-la perquè l'entorn és variant i no el pot predir ni generalitzar proutament com per optimitzar els seus resultats del tot. Tot i així, com veurem a continuació, malgrat la seva simplesa implementativa iteració del valor fa una bona feina solucionant l'entorn Taxi-v3 i ofereix una bona opció algorísmica dagut al seu baix cost computacional.

2.5 Experimentació

Implementació i Disseny Experimental

Cada experiment consisteix en la concatenació de les següents fases:

Definició del Paràmetre a Variar: Triar el paràmetre a variar (nombre d'episodis, γ, senyal de recompensa).

Configuració dels Valors de Prova: Definir un rang de valors per al paràmetre elegit.

Execució d'Experiments: Executar l'entrenament de l'agent diverses vegades per a cada configuració de paràmetre.

Recopilació de Dades: Registrar el rendiment de l'agent i

Anàlisi de Resultats: Comparar els resultats obtinguts per determinar l'impacte de cada paràmetre en el rendiment de l'agent.

Experiments:

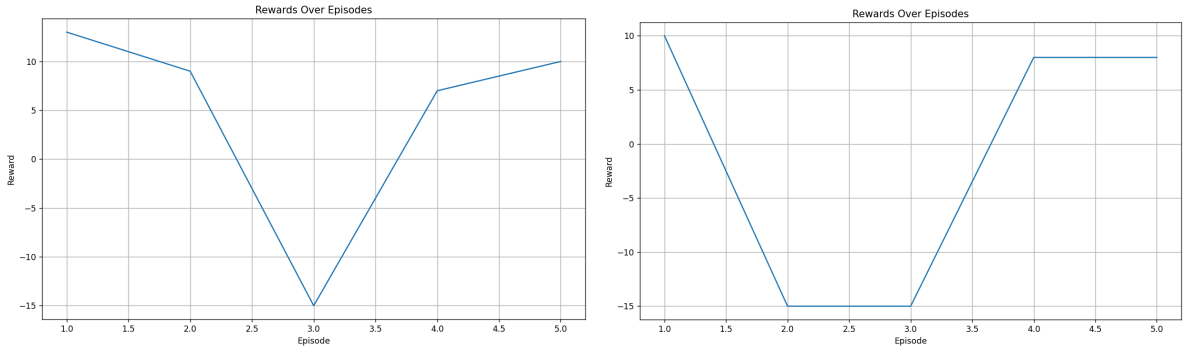
Variació del Nombre d'Episodis:

Objectiu: Avaluar com el rendiment de l'agent millora amb un major nombre d'episodis.

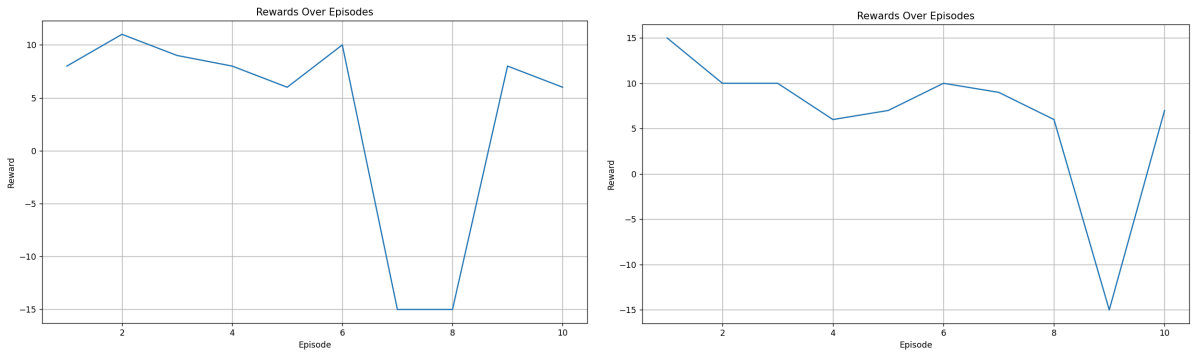
Hipòtesi: Hi ha un equilibri a trobar en el nombre d'episodis que l'agent entrena de tal manera que tant l'excés com el defecte d'aquests suposa un desajust en l'entrenament d'aquest respecte del seu rendiment òptim.

Farem execucions amb nombre d'episodis 5, 10 i 15 i veurem què passa:

Execucions amb nombre d'episodis = 5



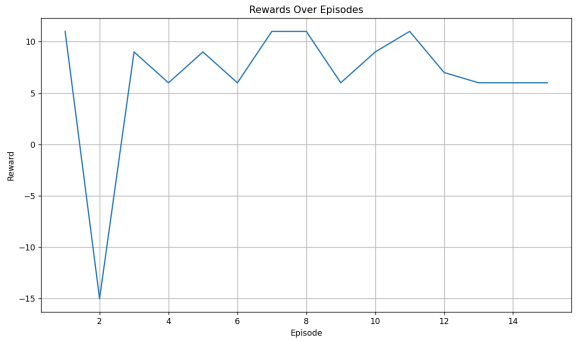
Execucions amb nombre d'episodis = 10

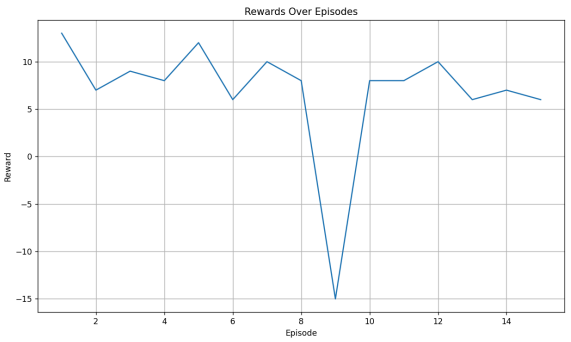


Execucions amb nombre d'episodis = 15

En aquest cas, seguint els valors estàndard definits anteriorment no trobàvem que l'algorisme convergís a una solució amb un rendiment mitjà de 10 (fet comprensible i coherent). En

conseqüència, hem reduït empíricament aquest threshold a 8 per poder oferir resultats sobre la variabilitat derivada d'ampliar el nombre d'episodis.





Analitzem les recompenses obtingudes:

1. NUM_EPISODES = 5:

Hi ha episodis en què el rendiment és tan bo com es podria desitjar i, en canvi, d'altres en que l'objectiu no s'assoleix. Veiem per tant que 5 episodis son prous perquè l'algorisme pugui trobar alguna solució però no ens garanteix consistència en el seu rendiment, com a mínim no a títol experimental.

Resulta evident que no hi ha prou per tal que l'agent trobi una política òptima

2. NUM_EPISODES = 10:

El rendiment de l'agent pel que fa a la seva recompensa ha millorat considerablement. S'ha estabilitzat, havent trobat una política que, en la majoria dels casos li permet resoldre el problema amb recompensa positiva. Si bé els entrenaments ara son una mica més llargs i calen més iteracions, la diferència temporal és nimia pel problema que es pretén resoldre. No s'ha obtingut una política òptima però, molt probablement, això serà impossible sense alterar altres paràmetres rellevants per l'algorisme.

3. NUM_EPISODES = 15:

Com s'ha comentat, aquest nombre d'episodis obliga a reduir el llindar de recompensa de 10 a 8 per tal de convergir regularment a una solució del problema. Tot i aquesta reducció el rendiment algorísmic pel que fa a recompensa és bo i, donat que aspira a un llindar menys exigent, necessita menys iteracions per convergir per la qual cosa es compensa el major nombre d'execucions del problema.

4. Conclusió:

En endavant, modifiquem l'estàndard per tal de realitzar els nostres experiments amb una versió millor que la original. Canviarem NUM_EPISODES de 5 a 10 degut a la millora clara que hem observat en el rendiment de l'algorisme en efectuar aquest canvi. Si no l'inicialitzem a 15 és degut que, malgrat també seria una opció algorísmica viable, ens obligaria a modificar

altres paràmetres amb què encara no hem experimentat (reduir el llindar de recompensa), cosa que preferim no fer per no influenciar prematurament els experiments.

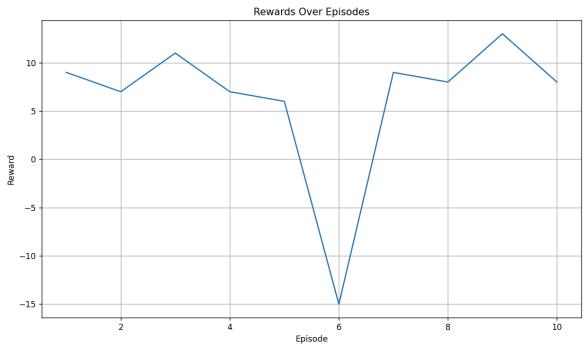
Variació del Factor de Descompte (γ):

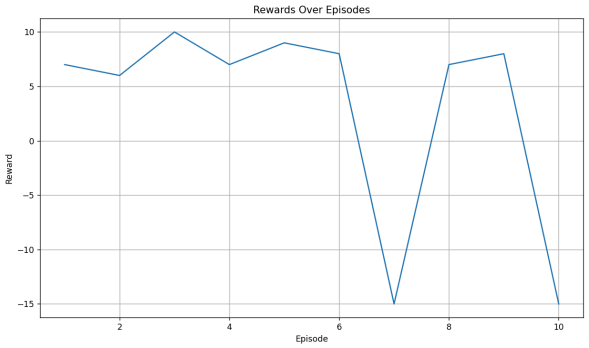
Objectiu: Determinar l'efecte del factor de descompte en el rendiment de l'agent.

Hipòtesi: Una reducció lleu del factor de descompte pot ser beneficial pel rendiment de l'agent.

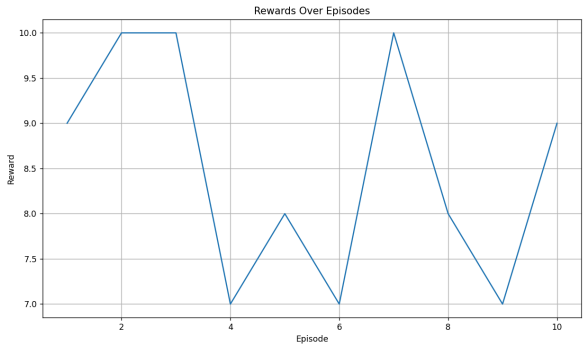
Farem execucions amb Gamma = 0.7, 0.8, 0.95.

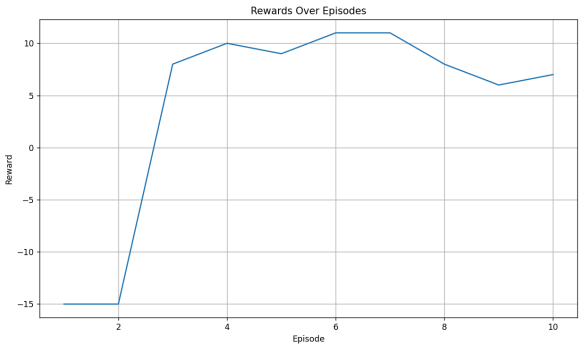
Execucions amb Gamma = 0.95



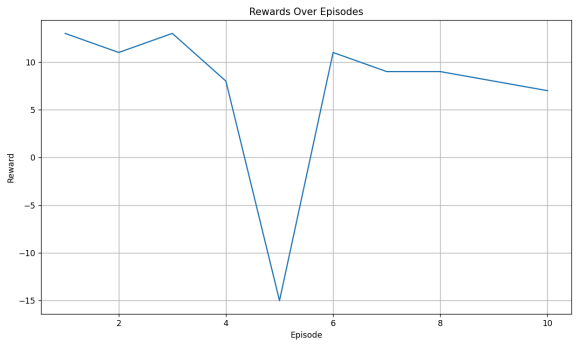


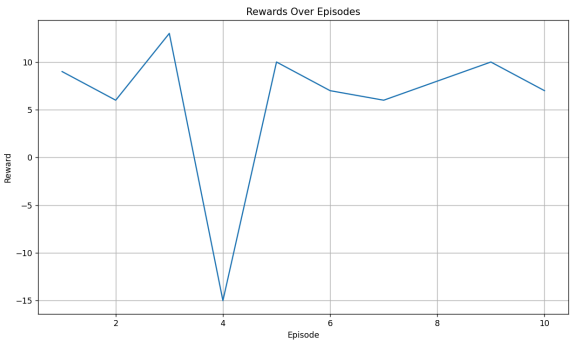
Execucions amb Gamma = 0.8





Execucions amb Gamma = 0.7





Ara, analitzarem les recompenses obtingudes per episodi en les diferents execucions amb GAMMA establert a 0.7, 0.8 i 0.95 respectivament.

1. GAMMA = 0.95:

Resultats estàndard: l'algorisme convergeix regularment i, en la majoria dels episodis, és capaç de resoldre satisfactòriament el problema amb recompensa positiva.

2. GAMMA = 0.8:

Es pot constatar una visible millor respecte del valor anterior de GAMMA. L'algorisme segueix sent imperfecte i subòptim doncs encara hi ha episodis en què no és capaç de portar la persona a l'hotel. Tot i així, el seu nivell de recompensa obtingut quan triomfa s'estabilitza i pren, en mitjana, un valor més alt.

3. GAMMA = 0.7:

Aquesta reducció de gamma ens torna a apropar a un rendiment similar al del primer cas. Constatem doncs que reduir la gamma no és generalment positiu per l'algorisme sinó que el que ens ha d'orientar ha de ser cercar un valor adequat que optimitzi el rendiment del nostre agent.

4. Conclusió:

Sembla que iteració del valor sobre Taxi-v3 és moderadament sensible a la negació de la influència de futurs estats pel que fa a optimitzar la seva política. Tot i així, com hem vist, una reducció excessiva d'aquesta rellevància d'estats futurs condueix també a un estat pitjor que el que es pot aconseguir cercant un punt mig que s'adapti a les necessitats del model.

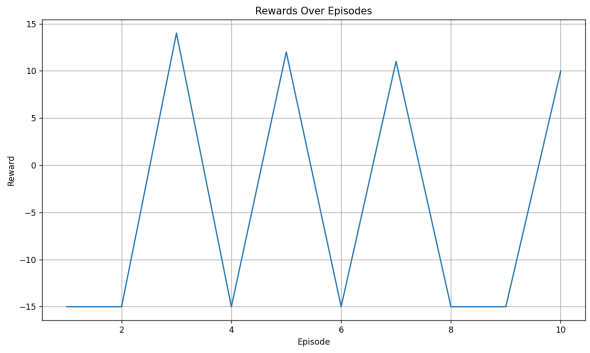
<u>Modificació del Senyal de Recompensa</u>:

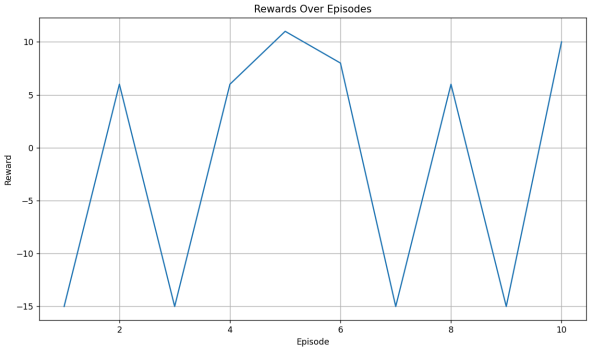
Objectiu: Provar diferents senyals de recompensa i el seu impacte en la política de l'agent.

Hipòtesi: Una senyal de recompensa adequada i realista pot ajudar a estabilitzar ràpidament el rendiment de l'agent cap a comportaments racionals desitjables.

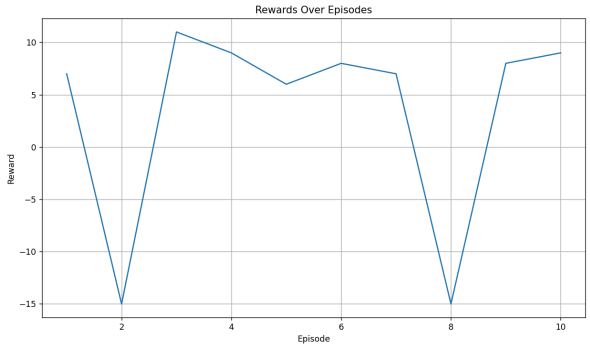
Farem execucions amb reward threshold 1, 5 i 10

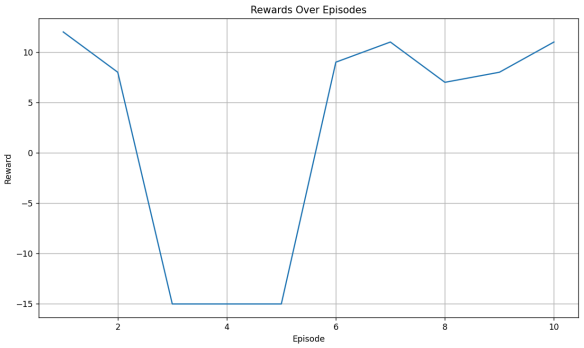
Execucions amb reward_threshold 1:



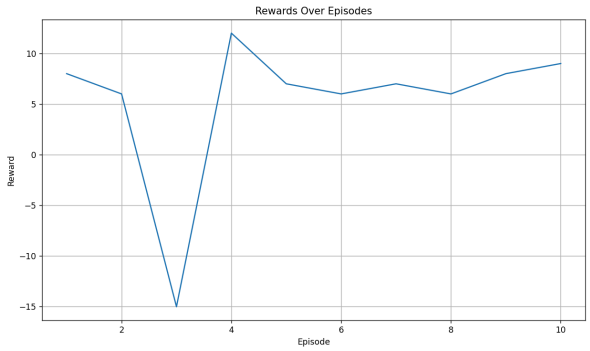


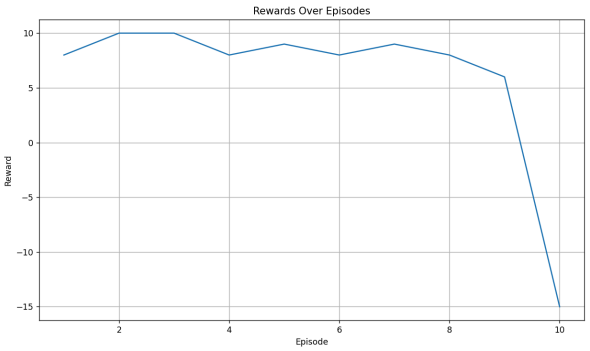
Execucions amb reward_threshold 5:





Execucions amb reward_threshold 10:





A continuació, analitzarem les recompenses obtingudes per episodi en les diferents execucions amb REWARD_THRESHOLD establert a 1, 5 i 10 respectivament.

1. REWARD_THRESHOLD = 1:

Les gràfiques mostren una gran inconsistència en el rendiment de l'agent, traduïda a una gran variabilitat en les recompenses obtingudes. Si bé en alguns episodis és capaç de resoldre el problema adequadament i amb recompenses altes, en tants o més episodis fracassa. Sembla doncs que aquest llindar de recompenses és massa baix per forçar l'agent a desenvolupar una política prou general com per resoldre adequadament la majoria dels casos.

2. REWARD_THRESHOLD = 5:

L'augment del llindar de recompensa mostra uns efectes clars, estabilitzant la majoria de les recompenses en un valor proper i, fins i tot, superior a ell però sense desempallegar-se d'entre un 20% i un 25% de casos de fracàs. Tot i així, l'augment del rendiment augura una bona dinàmica a aquest respecte.

3. REWARD_THRESHOLD = 10:

Com en el cas anterior, augmentar el llindar millora el rendiment. Les recompenses són més estables que quan el llindar era 5 i hi ha menys ocasions en les que l'algorisme no resol el problema. Aquest resulta, a més, ser el millor valor trobat donat que, d'augmentar-lo,

l'algorisme troba moltes dificultats per convergir donat el límit establert per la recompensa del problema.

4. Conclusió:

Hem trobat que pel problema Taxi-v3 la mètrica òptima amb què operar pel que fa al llindar de recompensa és 10, donat que és la que aconsegueix instar al model a optimitzar les seves recompenses, granatint alhora la convergència per tractar-se d'una aspiració prou realista en relació tant amb el disseny de l'agent com amb el problema.

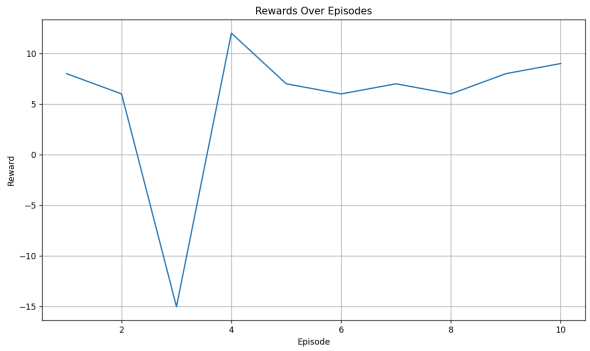
Variació del nombre de trajectòries:

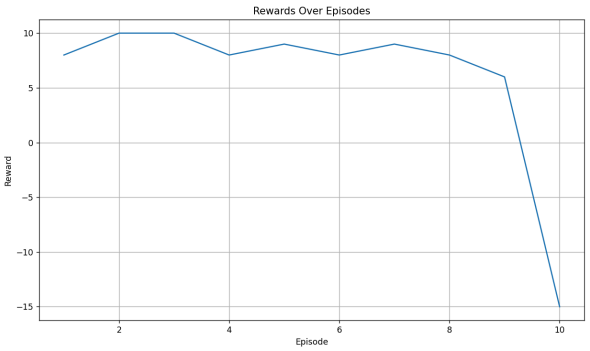
Objectiu: Provar diferents nombres de trajectòries i avaluar el seu impacte en el rendiment de l'agent.

Hipòtesi: Incrementar el nombre de trajectòries pot millorar consistentment el rendiment de l'agent sense gran cost d'entrenament si es restringeix aquest paràmetre a valors moderats.

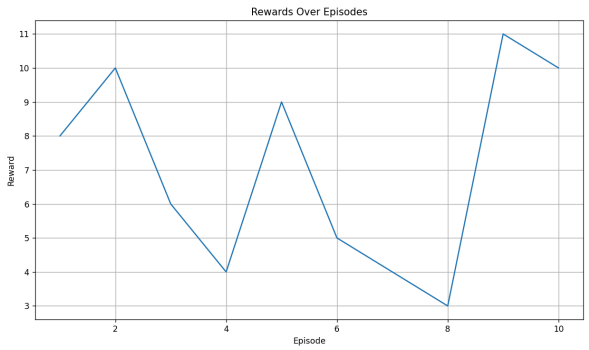
Experimentarem amb 15, 30, 50.

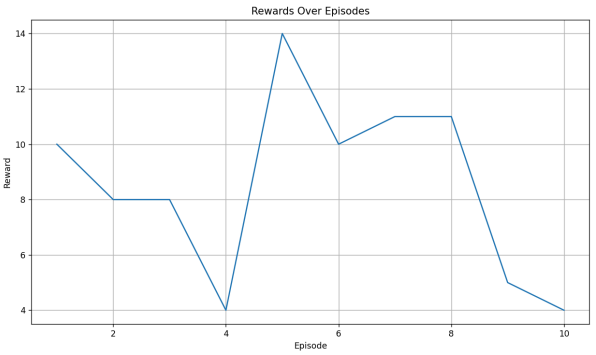
Execucions amb nombre de trajectories 15:



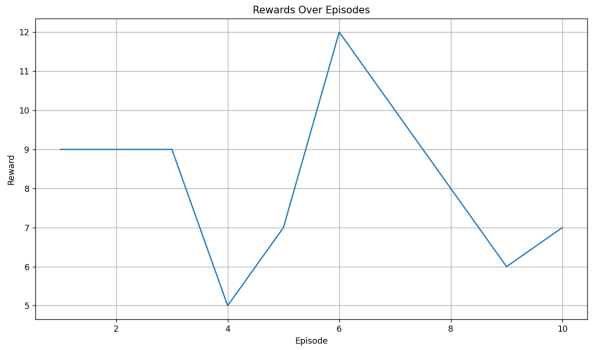


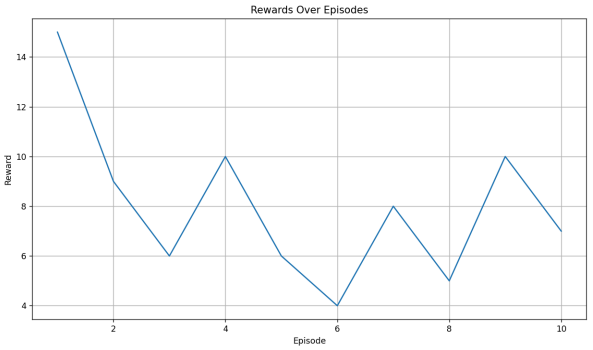
Execucions amb nombre de trajectories 30:





Execucions amb nombre de trajectories 50:





Per últim, analitzarem les recompenses obtingudes per episodi en les diferents execucions amb num_trajectories establert a 15, 30 i 50 respectivament.

1. num_trajectories = 15:

Les gràfiques mostren que l'agent és capaç de definir una política adaptable que li permet aconseguir recompenses positives en la majoria dels casos. Tot i així, encara hi ha episodis en què no aconsegueix resoldre el problema.

2. num_trajectories = 30:

L'augment de les trajectòries ha estabilitzat mitjanament el rendiment de l'agent. Sembla sempre arribar a una solució pel problema malgrat les recompenses obtingudes oscil·len en un interval d'entre 3 i 15. És una millora considerable dels resultats especialment pel que fa a la consistència del comportament de l'agent.

3. num_trajectories = 50:

Obtenim uns resultats similars al del cas anterior. S'aprecia un increment en el límit inferior de l'interval de recompenses obtingudes però no necessàriament aquest és prou substantiu com per justificar l'overhead computacional addicional.

4. Conclusió:

La hipòtesi que l'increment de trajectòries ajudaria a millorar el rendiment s'ha mostrat vertadera però limitada, doncs ràpidament hem trobat que aquesta millora és reduïda i no escala de forma lineal.

3. Estimació directa

3.1. Introducció

En aquesta pràctica, com a segona part, s'analitzarà el rendiment del mètode d'Estimació Directa en l'entorn Taxi-v3 de Gymnasium. S'explicaran els paràmetres més importants per a aquest mètode, es detallaran els experiments proposats per avaluar el seu rendiment, i es discutiran els avantatges i desavantatges del mètode en general i en el context específic de l'entorn Taxi-v3. A més, s'abordarà la inestabilitat de l'algorisme tant en aquest entorn com en general. No introduirem, com si hem fet en el cas anterior, els paràmetres rellevants, per ser els mateixos que per l'algorisme d'iteració del valor.

3.2. Estimació directa: Paràmetres importants

3.3. Estimació directa: Explicació general

L'estimació directa és un mètode basat en models on l'agent utilitza un conjunt de dades d'experiències per estimar el valor esperat de prendre una acció en un estat determinat. Aquest mètode pot ser senzill d'implementar i entendre, però té certes limitacions i avantatges específiques.

3.3.1. Avantatges

Els llistem:

- Simplicitat: És fàcil d'implementar i entendre.
- Efectivitat en Espais Petits: Pot ser molt efectiu en entorns amb un espai d'estats i accions petits.

3.3.2. Inconvenients

Llistem els inconvenients:

- Requereix Moltes Trajectòries: Per obtenir bones estimacions, es necessita una gran quantitat de dades d'experiència, cosa que pot ser costosa en termes de temps i recursos computacionals.
- Escalabilitat: En entorns amb grans espais d'estats i accions, la quantitat de dades necessàries pot ser prohibitiva.
- Dependència del Model: La qualitat de les estimacions depèn en gran mesura de la qualitat del model i de les dades d'experiència.
- Inestabilitat: Diferents execucions poden generar trajectòries significativament diferents, cosa que pot portar a estimacions inconsistents dels valors esperats. Ademés, el rendiment de l'algorisme pot variar dràsticament amb petits canvis en els paràmetres d'entrenament (per exemple, nombre d'episodis, factor de descompte). Per últim, existeix el risc que el model es sobreajusti a les trajectòries específiques utilitzades per a l'estimació, cosa que pot afectar negativament la seva capacitat per generalitzar a noves trajectòries.

3.4. Estimació directa aplicada a l'entorn Taxi-v3

L'estimació directa és un mètode senzill i intuïtiu per abordar problemes d'aprenentatge per reforç. En l'entorn Taxi-v3, aquest mètode es pot aplicar recopilant trajectòries en les quals l'agent executa diferents accions en diversos estats del joc. Aquestes trajectòries s'utilitzen per calcular el valor esperat de cada acció en cada estat, basant-se en les recompenses observades.

No obstant això, l'estimació directa presenta diversos problemes quan s'aplica a entorns complexos com Taxi-v3. En primer lloc, l'algoritme és inherentment inestable. La variabilitat en les trajectòries recopilades i l'exploració aleatòria de l'entorn fan que les estimacions dels valors esperats siguin inconsistents, resultant en un rendiment fluctuante de l'agent. Aquesta inestabilitat es veu agreujada pel fet que l'algoritme és molt senzill i no aprofita tècniques avançades d'aprenentatge, cosa que pot impedir trobar solucions òptimes per a problemes amb grans espais d'estats i accions.

Un altre problema important és l'escalabilitat. L'entorn Taxi-v3 té un espai d'estats i accions considerablement gran, la qual cosa requereix una quantitat substancial de trajectòries per obtenir estimacions precises. Aquesta necessitat de moltes dades comporta un cost computacional elevat en termes de temps i recursos, limitant l'aplicabilitat de l'estimació directa en entorns complexos. A més, l'algoritme corre el risc de sobreajustar-se a les trajectòries específiques utilitzades per a l'entrenament, cosa que pot afectar la capacitat de l'agent per generalitzar a noves situacions dins de l'entorn.

Els paràmetres d'entrenament, com el nombre d'episodis i el factor de descompte (`gamma`), també juguen un paper crucial en la qualitat de les estimacions. Un nombre insuficient d'episodis pot no proporcionar dades suficients per a una bona estimació dels valors esperats, mentre que un nombre excessiu pot resultar en un sobreajustament. Trobar un valor òptim per al factor de descompte és essencial per equilibrar la importància de les recompenses futures en comparació amb les immediates, cosa que afecta l'estabilitat i la qualitat de les prediccions.

En conclusió, tot i que l'estimació directa és fàcil d'implementar, la seva simplicitat i la inestabilitat inherent fan que sigui difícil obtenir resultats consistents i òptims en entorns complexos com Taxi-v3. Ajustar adequadament els paràmetres d'entrenament i considerar mètodes més avançats que puguin gestionar millor la complexitat de l'entorn pot ajudar a millorar el rendiment de l'agent, però el resultat no serà ni molt menys igual de bo que en altres algorismes.

3.5. Experimentació

Implementació i Disseny Experimental
Per a cada experiment, se seguiran els següents passos:

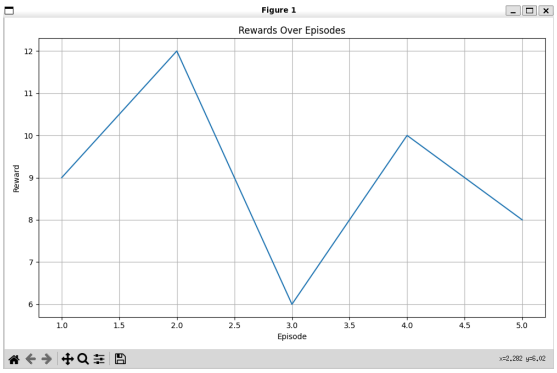
Definició del Paràmetre a Variar: Triar el paràmetre a variar (nombre d'episodis, γ, senyal de recompensa).
Configuració dels Valors de Prova: Definir un rang de valors per al paràmetre elegit.
Execució d'Experiments: Executar l'entrenament de l'agent diverses vegades per a cada configuració de paràmetre.
Recopilació de Dades: Registrar el rendiment de l'agent i
Anàlisi de Resultats: Comparar els resultats obtinguts per determinar l'impacte de cada paràmetre en el rendiment de l'agent.

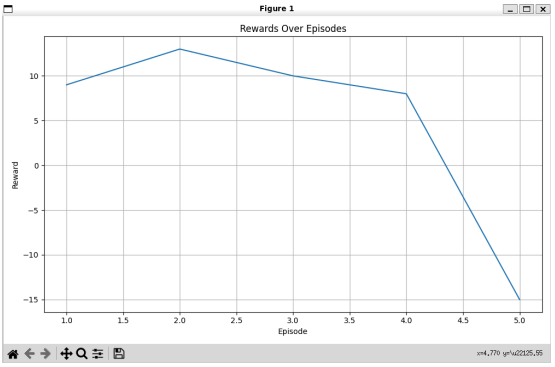
Experiments:

Variació del Nombre d'Episodis:

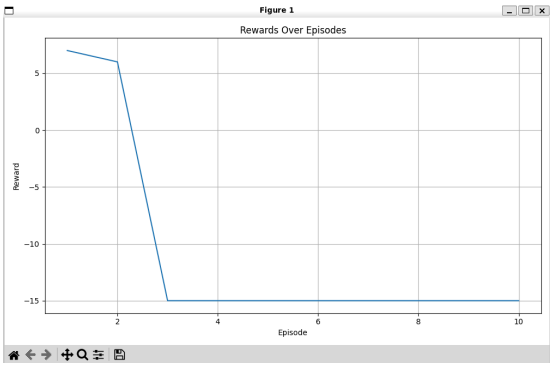
Objectiu: Avaluar com el rendiment de l'agent millora amb un major nombre d'episodis.
Hipòtesi: Augmentar el nombre d'episodis hauria de millorar la política resultant, però amb rendiments decreixents després d'un cert punt.

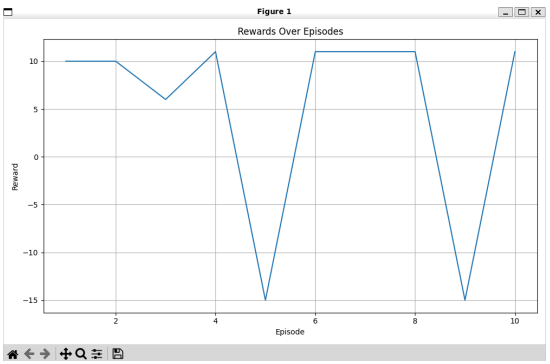
Farem execucions amb nombre d'episodis 5, 10 i 15 i veurem què passa:
Execucions amb nombre d'episodis = 5



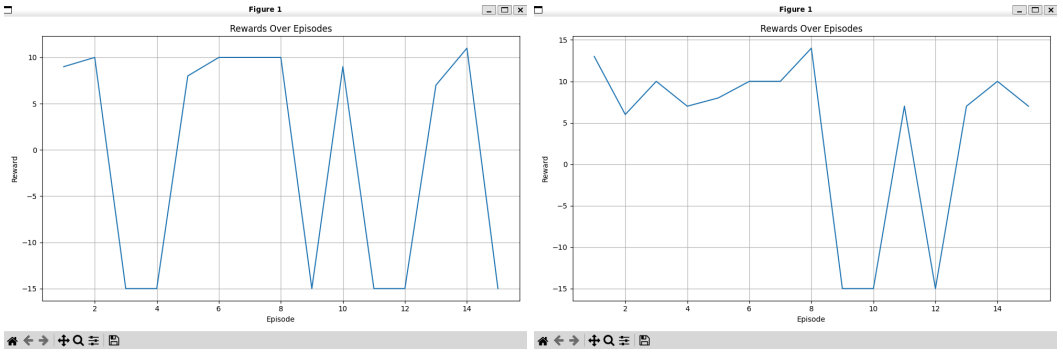


Execucions amb nombre d'episodis = 10





Execucions amb nombre d'episodis = 15



A continuació, analitzarem les recompenses obtingudes per episodi en les diferents execucions amb NUM_EPISODES establert a 5, 10 i 15 respectivament.

1. NUM_EPISODES = 5:
La gràfica mostra una variabilitat considerable en les recompenses per episodi.
Les recompenses fluctuen entre valors positius i negatius, indicant que l'agent no està aconseguint una política consistentment bona.

Amb només 5 episodis, l'agent no té suficient temps per aprendre una política estable i òptima.
La variabilitat indica que l'agent està explorant molt, però no ha convergit cap a una política que maximitzi les recompenses.

2. NUM_EPISODES = 10:
La gràfica continua mostrant una alta variabilitat, amb recompenses que oscil·len entre valors positius i negatius.
Tot i que hi ha alguns episodis amb recompenses positives consistents, no es veu una tendència clara cap a una millora constant.
Conclusió:

Incrementar el nombre d'episodis a 10 no ha estat suficient per eliminar la variabilitat en les recompenses.
L'agent segueix mostrant inestabilitat en la seva política, probablement degut a la insuficient quantitat de dades d'experiència per fer bones estimacions.

3. NUM_EPISODES = 15:
Encara que hi ha algunes recompenses positives consistents, la gràfica mostra variabilitat i pics negatius pronunciats.
La variabilitat persisteix, amb episodis que mostren recompenses negatives considerables.

Conclusió:

Amb 15 episodis, l'agent segueix sense aconseguir una política estable.
La inestabilitat persisteix, indicant que l'algoritme d'estimació directa no està aconseguint convergir cap a una política òptima amb el nombre d'episodis utilitzat.

El comportament observat en les gràfiques indica que l'agent entrenat amb estimació directa en l'entorn Taxi-v3 no està convergint cap a una política òptima, degut a la variabilitat alta i la manca d'una tendència clarament creixent en les recompenses.

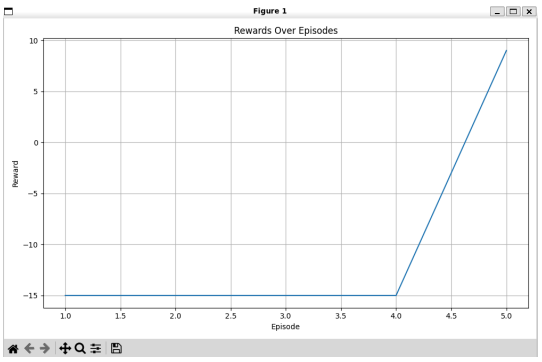
El comportament observat en les gràfiques indica que l'agent entrenat amb estimació directa en l'entorn Taxi-v3 no està convergint cap a una política òptima, degut a la variabilitat alta i la manca d'una tendència clarament creixent en les recompenses. Això té sentit degut a les capacitats de l'algorisme.

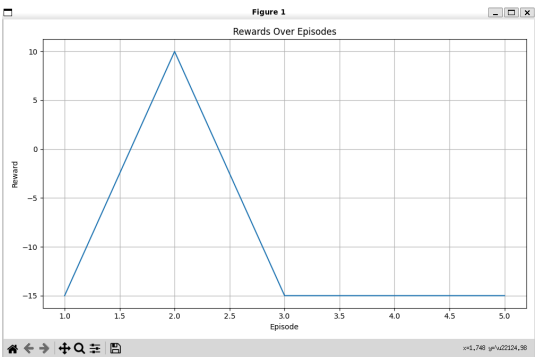
Variació del Factor de Descompte (γ):

Objectiu: Determinar l'efecte del factor de descompte en el rendiment de l'agent.
Hipòtesi: Valors intermedis de γ (per exemple, 0.7) podrien equilibrar millor les recompenses immediates i futures, resultant en una millor política.

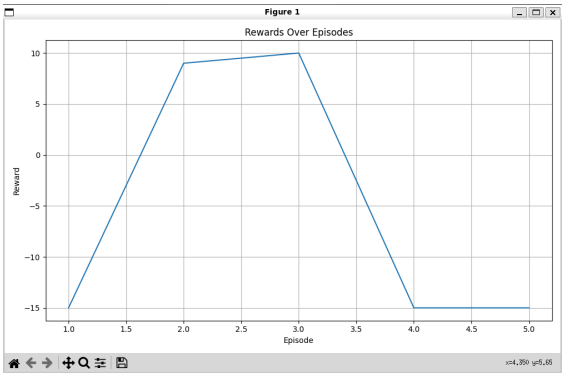
Farem execucions amb Gamma = 0.7, 0.8, 0.95.

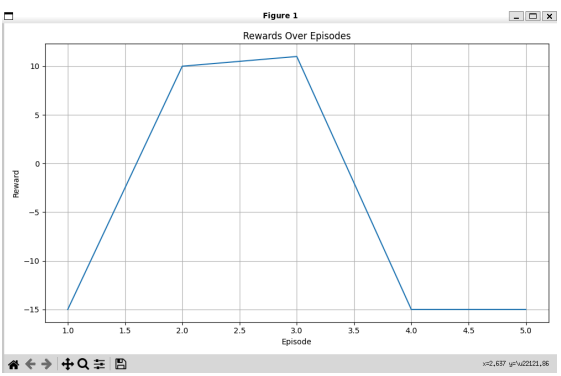
Execucions amb Gamma = 0.7



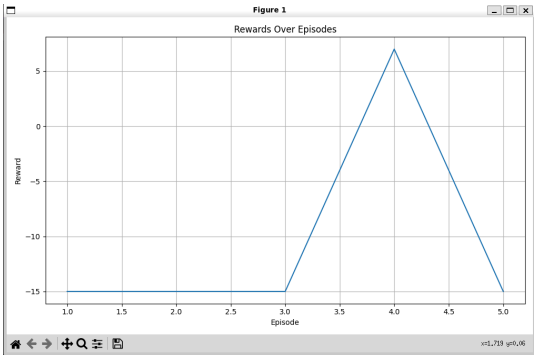


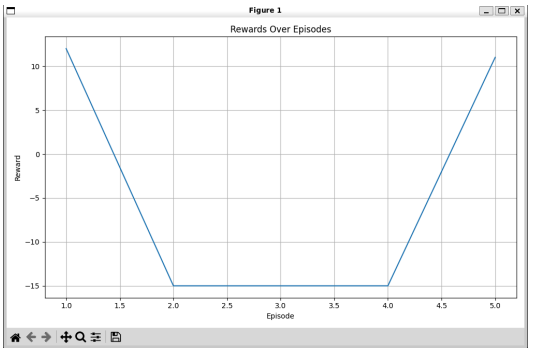
Execucions amb Gamma = 0.8





Execucions amb Gamma = 0.95





Ara, analitzarem les recompenses obtingudes per episodi en les diferents execucions amb GAMMA establert a 0.7, 0.8 i 0.95 respectivament.

1. GAMMA = 0.7:
La gràfica mostra una gran variabilitat en les recompenses per episodi, amb alguns episodis amb recompenses negatives i altres amb recompenses positives.
La tendència general no és clarament creixent, i es poden observar pics negatius seguits de pics positius.

Un valor de GAMMA = 0.7 posa més èmfasi en les recompenses immediates que en les futures, el que pot resultar en decisions que maximitzin les recompenses a curt termini però que no garanteixin una política òptima a llarg termini.
L'agent mostra inestabilitat en les seves decisions, probablement degut a la manca de consideració suficient per les recompenses futures.

2. GAMMA = 0.8:
La variabilitat continua sent alta, amb recompenses que oscil·len entre valors negatius i positius.
Tot i que hi ha alguns episodis amb recompenses positives constants, no es veu una tendència clara cap a una millora constant.

Amb GAMMA = 0.8, l'agent encara no està equilibrant adequadament les recompenses immediates i futures, cosa que resulta en una política inestable.
L'agent segueix mostrant dificultats per aprendre una política consistentment bona, indicant que encara no es té prou en compte les recompenses futures.

3. GAMMA = 0.95:
La variabilitat en les recompenses persisteix, amb episodis que mostren tant recompenses positives com negatives.

Encara que hi ha algunes recompenses positives consistents, la gràfica mostra una tendència inestable amb pics negatius pronunciats.

Amb GAMMA = 0.95, l'agent posa més èmfasi en les recompenses futures, però encara no aconsegueix una política estable.
L'agent mostra inestabilitat en les recompenses, probablement degut a la complexitat de l'entorn i a la insuficient quantitat de dades d'experiència per fer bones estimacions.

Les gràfiques per a diferents valors de GAMMA mostren diversos problemes inherents a l'algoritme d'estimació directa en l'entorn Taxi-v3:

Alta Variabilitat en les Recompenses:
Les recompenses per episodi mostren una alta variabilitat, indicant que l'agent no està aprenent de manera consistent.
Aquesta variabilitat pot ser deguda a la insuficient exploració de l'espai d'estats i accions, així com a les actualitzacions estocàstiques dels valors de les accions.

No Convergència:
Les gràfiques no mostren una tendència clara cap a una millora consistent en les recompenses, suggerint que l'agent no està convergint cap a una política òptima.
Això pot ser degut a la insuficient quantitat de trajectòries utilitzades per l'estimació directa.

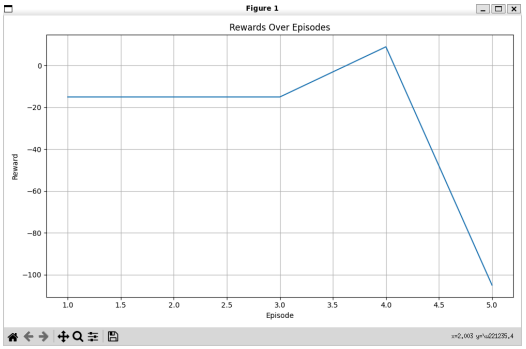
Dependència del Valor de GAMMA:
Canviar el valor de GAMMA afecta com l'agent valora les recompenses futures en comparació amb les immediates.
Valors baixos de GAMMA (com 0.7) fan que l'agent posi més èmfasi en les recompenses immediates, mentre que valors alts (com 0.95) fan que consideri més les recompenses futures.
Cap dels valors provats ha estat suficient per obtenir una política consistentment bona, la qual cosa remarca una vegada més la inestabilitat de l'algorisme i la dificultat d'obtenir bons resultats amb un algorisme tan senzill.

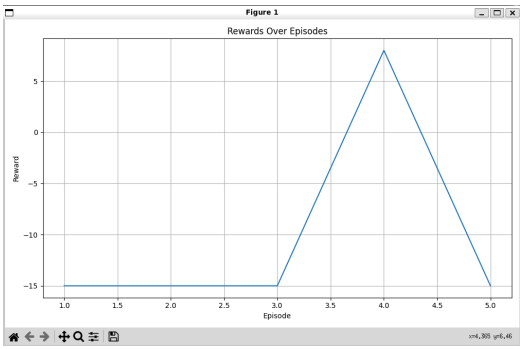
Modificació del Senyal de Recompensa:

Objectiu: Provar diferents senyals de recompensa i el seu impacte en la política de l'agent.
Hipòtesi: Una senyal de recompensa ben dissenyada pot accelerar l'aprenentatge de l'agent i millorar el seu rendiment final.

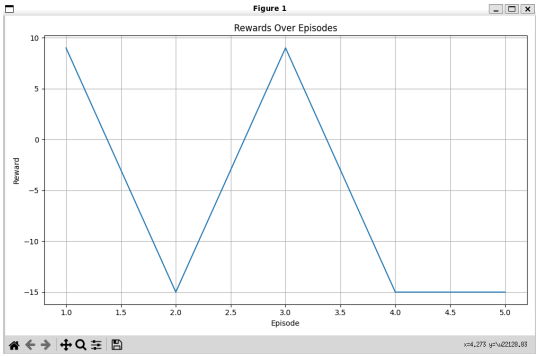
Farem execucions amb reward threshold 1, 5 i 10

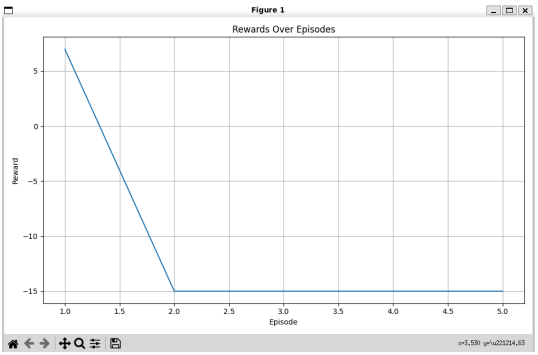
Execucions amb reward_threshold 1:



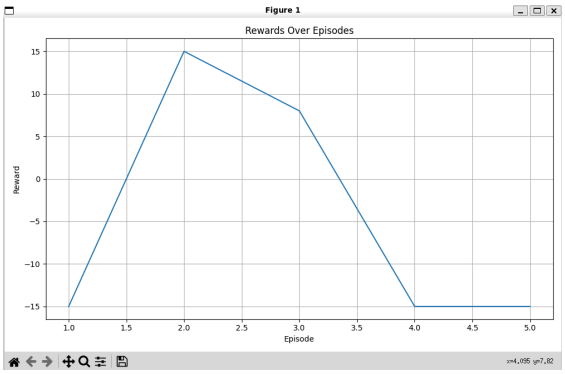


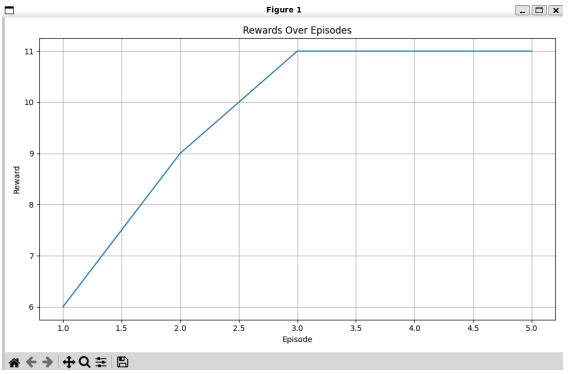
Execucions amb reward_threshold 5:





Execucions amb reward_threshold 10:





A continuació, analitzarem les recompenses obtingudes per episodi en les diferents execucions amb REWARD_THRESHOLD establert a 1, 5 i 10 respectivament.

1. REWARD_THRESHOLD = 1:
La gràfica mostra una alta variabilitat en les recompenses per episodi, amb valors negatius constants i un pic positiu al final.
No es veu una tendència clara cap a una millora constant.

Amb REWARD_THRESHOLD = 1, l'agent no està aconseguint millorar consistentment la seva política.

La variabilitat indica que l'agent està lluitant per aconseguir recompenses positives, i no sembla estar aprenent de manera estable.

2. REWARD_THRESHOLD = 5:
La variabilitat continua sent alta, amb recompenses que oscil·len entre valors negatius i positius.
Hi ha una tendència a aconseguir pics de recompensa positiva, però aquests no són constants ni sostinguts.

Amb REWARD_THRESHOLD = 5, l'agent mostra una mica més de capacitat per aconseguir recompenses positives, però encara hi ha molta variabilitat i inestabilitat.
L'agent encara té dificultats per mantenir una política consistentment bona.

3. REWARD_THRESHOLD = 10:
La variabilitat en les recompenses persisteix, amb episodis que mostren tant recompenses positives com negatives.
Hi ha una tendència més clara cap a l'obtenció de recompenses positives al final, però encara hi ha pics negatius pronunciats.

Amb REWARD_THRESHOLD = 10, l'agent sembla estar aconseguint una mica més de consistència en l'obtenció de recompenses positives, però encara mostra una variabilitat significativa.
La política de l'agent és una mica més estable, però encara no és òptima.

Les gràfiques per a diferents valors de REWARD_THRESHOLD mostren diversos problemes inherents a l'algoritme d'estimació directa en l'entorn Taxi-v3:

Alta Variabilitat en les Recompenses:
Les recompenses per episodi mostren una alta variabilitat, indicant que l'agent no està aprenent de manera consistent.
Aquesta variabilitat pot ser deguda a la insuficient exploració de l'espai d'estats i accions, així com a les actualitzacions estocàstiques dels valors de les accions.

No Convergència:
Les gràfiques no mostren una tendència clara cap a una millora consistent en les recompenses, suggerint que l'agent no està convergint cap a una política òptima.
Això pot ser degut a la insuficient quantitat de trajectòries utilitzades per l'estimació directa.

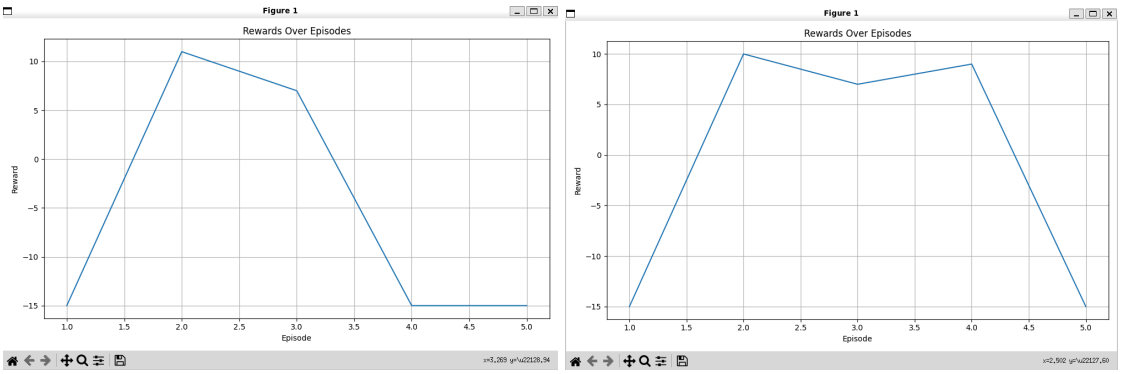
Dependència del Valor de REWARD_THRESHOLD:
Canviar el valor de REWARD_THRESHOLD afecta la meta que l'agent intenta assolir en termes de recompenses.
Valors baixos de REWARD_THRESHOLD (com 1) no semblen suficients per guiar l'agent cap a una política òptima, mentre que valors més alts (com 10) mostren una mica més de consistència, però encara no són suficients per una política estable. La millora obtinguda es

veu reflexada en sentit contrari a nivell de temps i iteracions, tenint amb un REWARD_THRESHOLD unes 8500 iteracions i en canvi amb 5 tenia 3500 i amb 1 en tenia 2000.

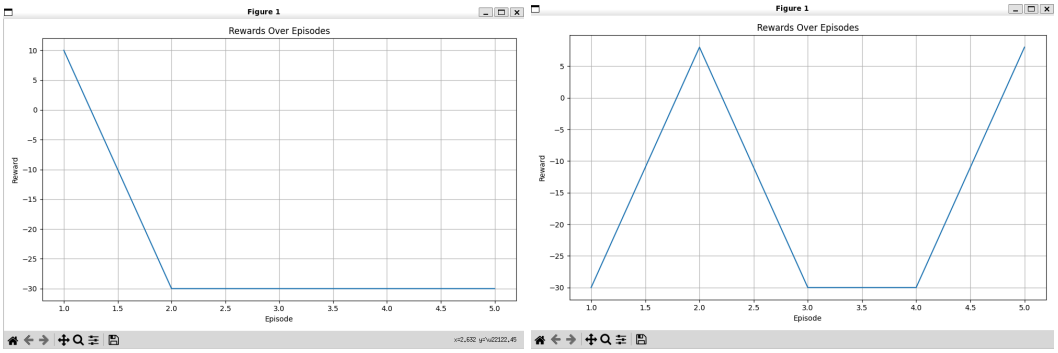
(Iteracions 1 : 1500 i 2500; Iteracions 5: 3500; Iteracions 10: 8500)

Variació del nombre de trajectòries:
Objectiu: Provar diferents nombres de trajectòries i avaluar el seu impacte en el rendiment de l'agent.
Hipòtesi: Un nombre adequat de trajectòries pot proporcionar una millor estimació dels valors esperats de les accions, millorant així l'aprenentatge de l'agent i el seu rendiment final.

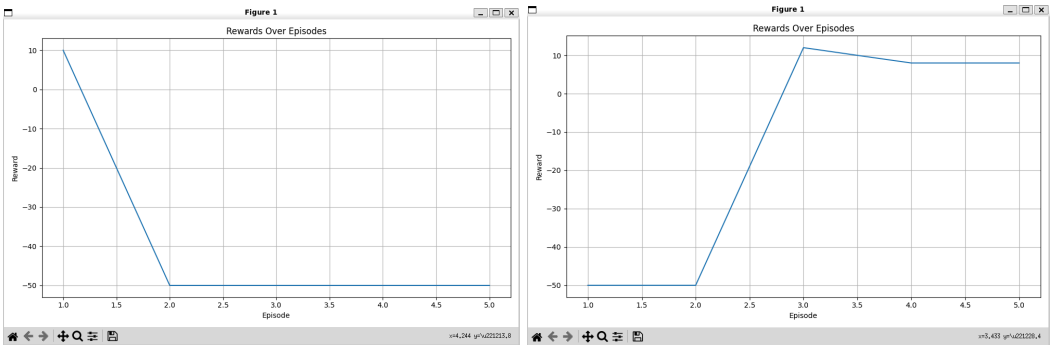
Experimentarem amb 15, 30, 50.
Execucions amb nombre de trajectories 15:



Execucions amb nombre de trajectories 30:



Execucions amb nombre de trajectories 50:



Per últim, analitzarem les recompenses obtingudes per episodi en les diferents execucions amb num_trajectories establert a 15, 30 i 50 respectivament.

1. num_trajectories = 15:
La gràfica mostra una alta variabilitat en les recompenses per episodi, amb valors que oscil·len entre recompenses positives i negatives.
No es veu una tendència clara cap a una millora constant.

Amb num_trajectories = 15, l'agent no està aconseguint millorar consistentment la seva política.
La variabilitat indica que l'agent està lluitant per obtenir recompenses positives, i no sembla estar aprenent de manera estable.

2. num_trajectories = 30:
La variabilitat continua sent alta, amb recompenses que oscil·len entre valors negatius i positius.
Tot i que hi ha una tendència a aconseguir recompenses positives, els episodis mostren una inestabilitat significativa.

Amb num_trajectories = 30, l'agent mostra una mica més de capacitat per aconseguir recompenses positives, però encara hi ha molta variabilitat i inestabilitat.
L'agent encara té dificultats per mantenir una política consistentment bona.

3. num_trajectories = 50:
La variabilitat en les recompenses persisteix, amb episodis que mostren tant recompenses positives com negatives.
Hi ha una tendència més clara cap a l'obtenció de recompenses positives cap al final, però encara hi ha pics negatius pronunciats.

Amb num_trajectories = 50, l'agent sembla estar aconseguint una mica més de consistència en l'obtenció de recompenses positives, però encara mostra una variabilitat significativa.
La política de l'agent és una mica més estable, però encara no és òptima.

Les gràfiques per a diferents valors de num_trajectories mostren diversos problemes inherents a l'algoritme d'estimació directa en l'entorn Taxi-v3:

Alta Variabilitat en les Recompenses:
Les recompenses per episodi mostren una alta variabilitat, indicant que l'agent no està aprenent de manera consistent.
Aquesta variabilitat pot ser deguda a la insuficient exploració de l'espai d'estats i accions, així com a les actualitzacions estocàstiques dels valors de les accions.

No Convergència:
Les gràfiques no mostren una tendència clara cap a una millora consistent en les recompenses, suggerint que l'agent no està convergint cap a una política òptima.
Això pot ser degut a la insuficient quantitat de trajectòries utilitzades per l'estimació directa.

Dependència del Nombre de Trajectòries:
Canviar el nombre de trajectòries afecta la quantitat de dades d'experiència que l'agent pot utilitzar per aprendre.
Valors baixos de num_trajectories (com 15) no semblen suficients per guiar l'agent cap a una política òptima, mentre que valors més alts (com 50) mostren una mica més de consistència, però encara no són suficients per una política estable.

4. Q-Learning

4.1. Introducció

En aquesta tercera part, s'analitzarà el rendiment de *Q-Learning* en el mateix entorn de *Taxi-v3* de Gymnasium. Es raonarà sobre els hiperparàmetres més rellevants per aquest mètode, es detallaran els experiments proposats per avaluar el rendiment, i es discutirà com s'adapta *Q-Learning* a aquest context específic.

4.2. Paràmetres importants

1. T-MAX:

Representa el nombre màxim de passos que l'agent pot fer abans d'acabar l'episodi. Si el nombre de passos arriba al seu màxim, l'episodi acaba encara que no s'hagi arribat a un estat final. Un valor alt pot permetre a l'agent explorar més en cada episodi, però també pot augmentar el temps total d'entrenament.

2. NUM_EPISODES:

Indica el número d'episodis que l'agent realitzarà durant l'entrenament. Generalment, un nombre alt d'episodis pot permetre que l'agent aprengui millor les polítiques òptimes per diferents situacions de l'entorn.

3. Factor de descompte (GAMMA):

Determina quin pes s'atorga a les recompenses futures en relació amb les recompenses immediates. El rang de valors és entre 0 i 1. Contra més proper a 1 sigui el seu valor, més es valoraran les recompenses futures. En canvi, contra més proper a 0 sigui el seu valor, menys valor es donen a les recompenses futures i més valor a les immediates.

4. LEARNING_RATE:

La taxa d'aprenentatge és la taxa a la qual l'agent actualitza les seves estimacions de valors Q. El rang és entre 0 i 1. Un alt valor fa que l'agent sigui més volàtil en el seu aprenentatge, i podria potencialment crear inestabilitat, i un valor baix fa que l'aprenentatge sigui més estable però també més lent.

5. EPSILON:

Controla el balanç entre exploració i explotació. El seu rang de valors és entre 0 i 1. Un valor proper a 1 indica més exploració (prendre accions aleatòries), i un valor proper a 0 prioritza explotació (prendre les accions amb el valor Q més alt).

6. EPSILON_DECAY:

Després de cada episodi, el valor de epsilon es multiplica per *EPSILON_DECAY*. El seu rang de valors és entre 0 i 1, per tant un valor inferior a 1 causa que el valor d'epsilon decaigui gradualment després de cada episodi. D'aquesta forma, l'agent explora menys a mesura que avança l'entrenament.

7. LEARNING_RATE_DECAY:

Similar a *EPSILON_DECAY*, aquest paràmetre determina com decau la taxa d'aprenentatge al llarg del temps. Pot ser útil per ajustar la velocitat d'aprenentatge i estabilitzar-lo a mesura que l'agent acumula més experiència.

4.3. *Q-Learning*: Explicació general

Q-learning és un algoritme *off-policy* de *reinforcement learning* que s'utilitza per aprendre polítiques òptimes en entorns discrets, on un agent pren accions per maximitzar la seva recompensa a llarg termini. Utilitza una taula Q que mapeja pars estat-acció (s,a) a valors Q, que representen la utilitat esperada de prendre una acció '*a'* en un estat donat '*s'*.

A cada pas, l'agent selecciona una acció segons una política d'exploració i explotació. Això significa que a vegades escollirà les accions amb el Q-value més alt, i altres vegades n'escollirà d'aleatòries per explorar noves possibilitats (idealment, amb el temps el nombre d'accions aleatòries es va reduint).

Després de prendre una acció *'a'* en un estat donat *'s'*, s'observa la seva reva recompensa *'r'* i el pròxim estat *s'*, de tal forma que l'agent té coneixement de la tupla (*s,a,r,s'*). Amb això, l'agent actualitza la taula Q utilitzant la recompensa immediata i l'estimació de la recompensa futura prenen l'acció que la maximitza.

Tot el procediment es repeteix fins a convergir cap a una política òptima.

4.3.1. Avantatges

Model lliure: Q-learning és un algoritme de *reinforcement learning* no basat en model, el que significa que no requereix conèixer explícitament la funció de transició d'estats ni les rewards. Aquest fet el fa ser una opció adequada per problemes on el model de l'entorn és desconegut o difícil de modelar.

Convergència garantitzada: Si s'exploren totes les combinacions estat-acció un nombre infinit de cops, i sota certes altres condicions, es garantitza que Q-learning convergeix cap als valors Q òptims, i per tant cap a una política òptima.

4.3.2. Inconvenients

Exploració vs Explotació: Q-learning s'enfronta al desafiament d'equilibrar exploració i explotació. Una mala elecció d'aquest balanç pot ralentitzar o limitar l'aprenentatge.

Sensibilitat als hiperparàmetres: Q-learning depèn de diferents hiperparàmetres, per tant és vital un bon ajustament d'aquests per obtenir un bon rendiment de l'algoritme.

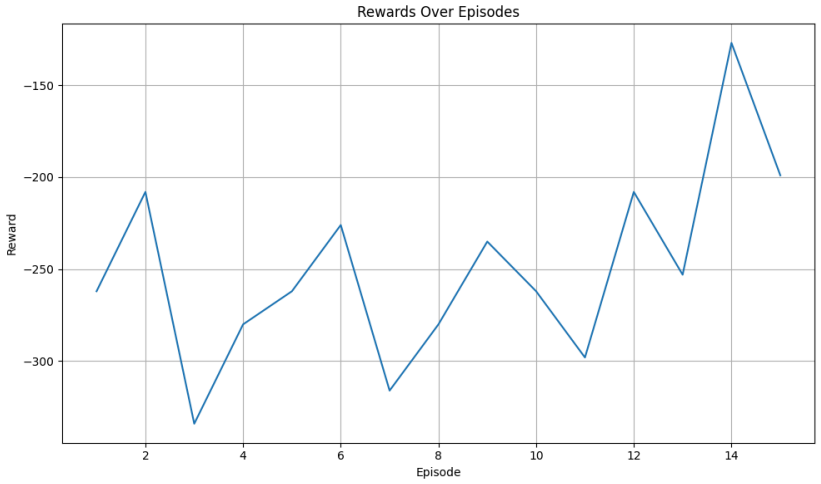
Escalabilitat: En entorns amb grans espais d'estats o accions, la representació de la taula Q pot tornar-se substancialment gran, el que dificulta el seu emmagatzematge i actualització.

4.4. *Q-Learning* aplicat a l'entorn Taxi-v3

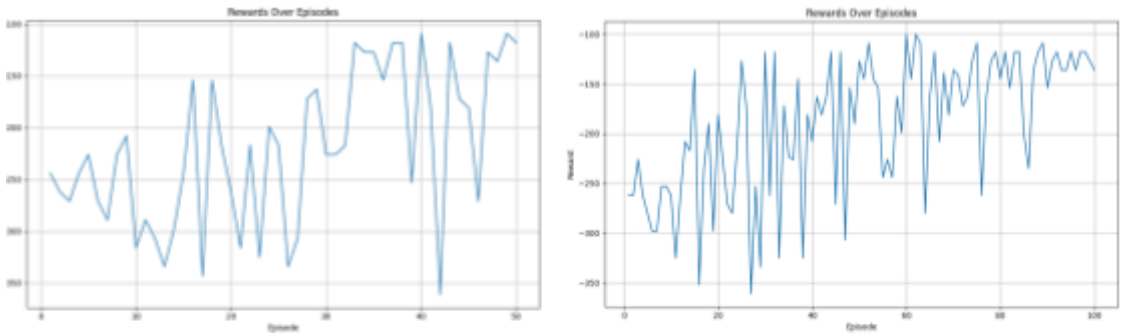
4.5. Experimentació

4.5.1. NUM_EPISODES

En aquest experiment analitzarem el nombre d'episodis necessaris per entrenar el model. Per començar, establim el nombre d'episodis a 15. El següent gràfic representa l'evolució de la reward obtinguda a mesura que avancen els episodis.

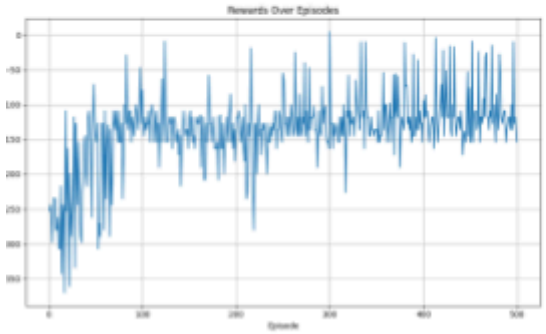


El que ens interessa és que la reward augmenti amb el pas dels episodis, perquè això demostraria que l'agent aprèn. En aquest cas, no es veu una clara tendència alcista, és molt divergent. L'hipòtesi que tenim en aquest cas és que 15 és un número molt petit d'episodis i no són suficients perquè l'agent aprengui bé. Necessitem més iteracions, per tant anem a augmentar el nombre d'episodis.

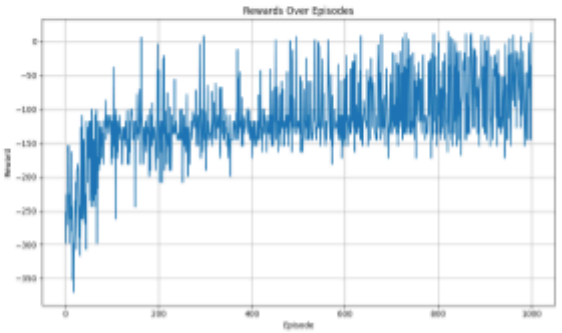


(a) 50 episodes

(b) 100 episodes



(c) 500 episodes

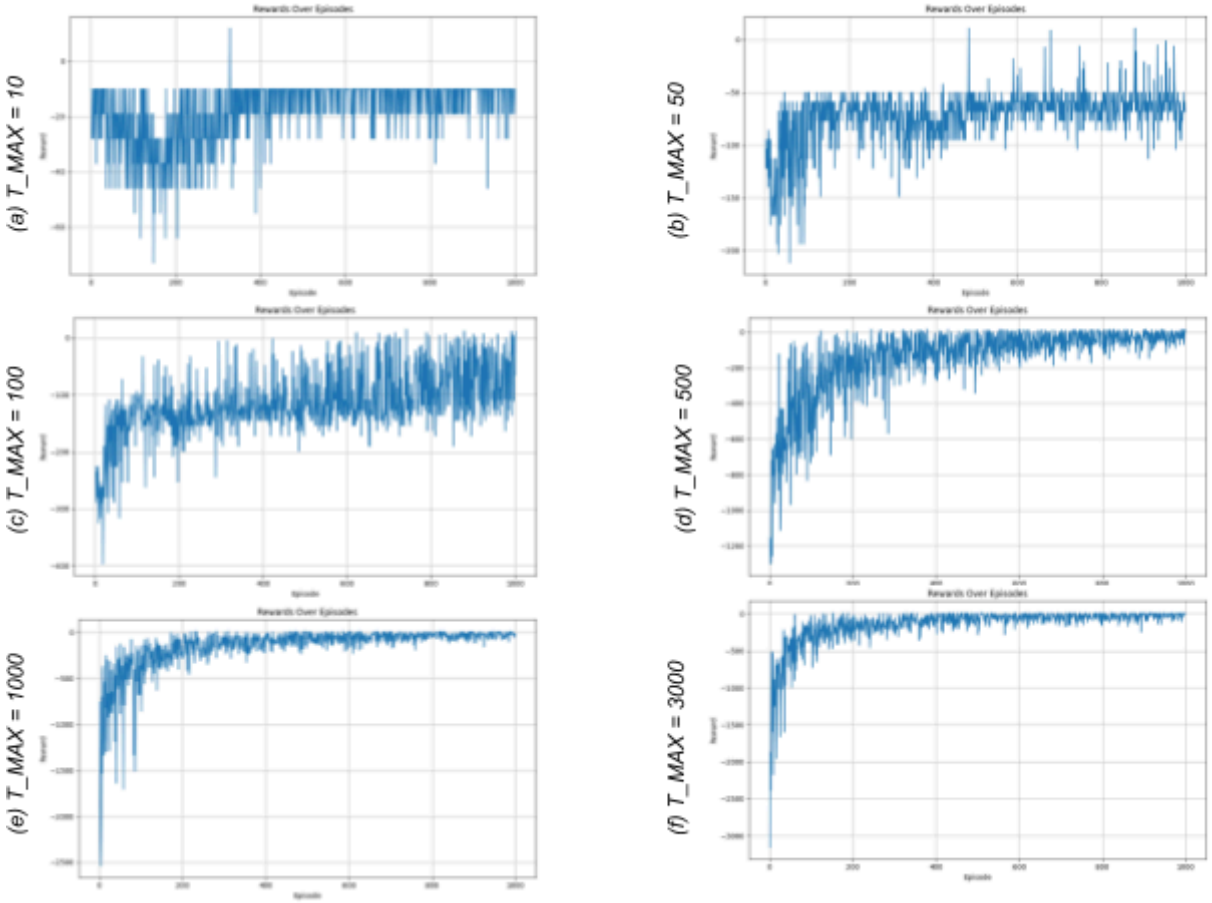


(d) 1000 episodes

Com podem veure, el gràfic (b) és el resultat de l'experiment amb 100 episodis, i en aquest cas es veu una tímida tendència alcista. Si ampliem a més episodis (500 i 1000), podem veure en ambdós casos com la reward augmenta respecte a l'inici i arriba un punt on s'estabilitza. Aproximadament, a partir dels 150-200 episodis la reward s'estanca, indicant que l'agent no és capaç d'aprendre més amb la configuració actual. Com veiem, augmentar de 500 a 1000 episodis no millora els resultats, perquè independentment del nombre d'episodis, l'agent ha arribat al seu "òptim" amb la configuració de paràmetres que té.
De moment, ens quedarem amb la configuració de 1000 episodis, tot i que aparentment amb 500 també ens valdria.

4.5.2. T_MAX

En aquest experiment analitzarem l'influència de t_max (el nombre màxim d'accions per episodi) amb l'evolució de la reward al llarg dels episodis.
La hipòtesi que tenim és que si el nombre t_max és molt baix, molts episodis del nostre agent acabaran truncats, i això impedeix que l'agent explori més trajectories que donin bons resultats. Si augmentem el t_max, l'agent tindrà més oportunitats per arribar a estats finals i per tant poder aprendre millors trajectòries, contràriament a si l'episodi acaba truncat. No obstant això, podria ser que un valor massa elevat de t_max augmenti el temps d'entrenament.



Encara que les gràfiques (e) i (f) semblin més adequades, precises i menys diverses, no ens hem de confondre que estan a escala diferent. Per tant, NO podem dir que augmentar t_max fa que el resultats siguin menys divergents. El que sí que podem veure és a mesura

que t_max augmenta, les rewards inicials són molt pitjors (grans nombres negatius), i aquest fet té sentit perquè com estem allargant la vida dels episodis, l'agent va realitzant moltes accions que van acumulant reward negativa. En canvi, quan t_max és més petita, a l'inici els episodis es tallen abans i per tant no s'arriba a una reward tant negativa. En tots els casos, es veu una augmentació de la reward amb el pas dels episodis.

La principal conclusió que podem extreure d'aquest experiment és que augmentant el t_max, permetem a l'agent investigar més, però si no s'arriba ràpidament a un estat final llavors s'acumula una reward molt negativa, i això es projecta clarament en els primers episodis, on l'agent encara no ha après res i per tant arriba a rewards molt negatives. A mesura que avancen els episodis, l'agent aprèn, i per tant no arriba a esgotar el nombre màxim d'accions, arriba abans a un estat final, i per tant l'episodi acaba satisfactòriament amb un reward més decent. A més, amb valors t_max alts, com li deixem més marge per realitzar accions sense "tallar" un episodi, l'agent descobreix i aprèn més trajectòries eficients i això es podria traduir en què les rewards obtingudes amb t_max més alts són més altes. Tot i així, en aquest cas no és una diferència enorme.

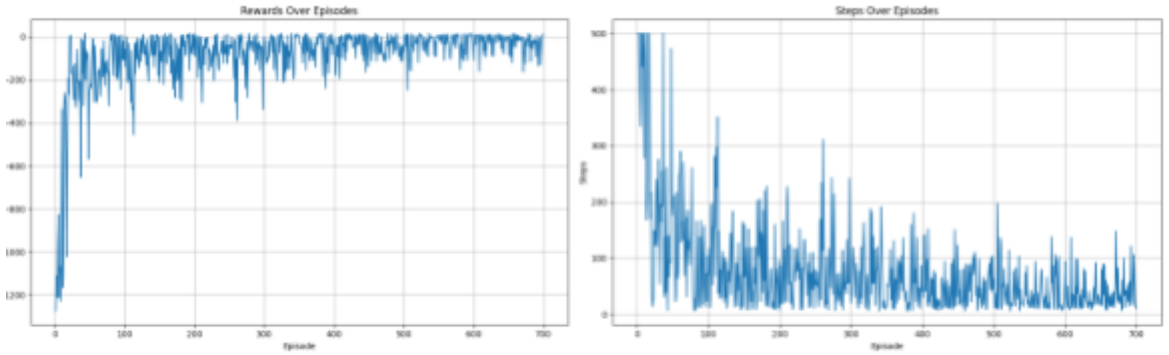
4.5.3. Learning rate decay

En aquest experiment, analitzarem en què afecta aplicar un learning_rate_decay al learning rate. A cada episodi, el learning rate disminueix multiplicant el seu valor pel learning_rate_decay (valor inferior a 1). Amb un decay adequat, s'aconsegueix estabilitzar l'aprenentatge al llarg del temps.

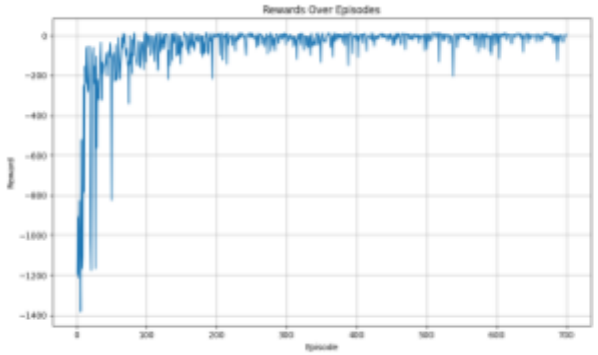
La nostra hipòtesi en aquest cas és que valors massa baixos de decay poden fer que l'agent adquireixi learning rates baixos massa ràpid, limitant l'aprenentatge, ralentitzant-lo massa. Al ser un entorn molt determinista, creiem que un learning rate alt podria donar millors resultats.

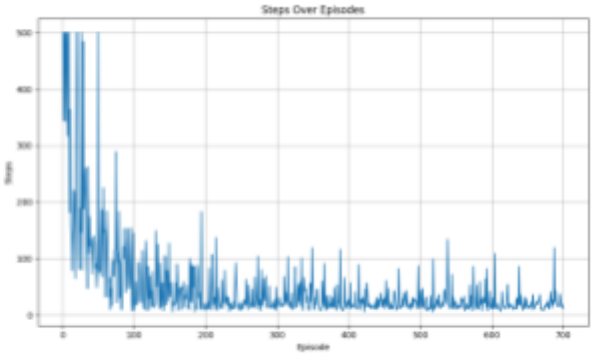
Per aquest experiment considerem que el valor mínim que pot tenir el learning rate és 0.05, i que al learning rate se li aplicarà el decay cada 5 iteracions. Comencem amb un learning rate d'1 que anirà decreixent en funció del decay que assignem en els experiments.
Considerarem 2 gràfiques: 1) La reward en funció dels episodis; i 2) el nombre de passos realitzats per l'agent en cada episodi.

Learning rate decay = **0.9**

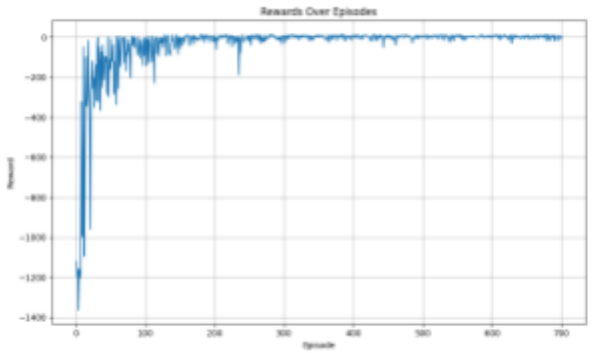


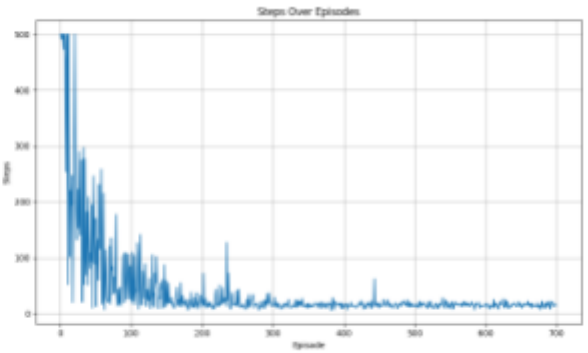
Learning rate decay = **0.95**





Learning rate decay = **0.99**





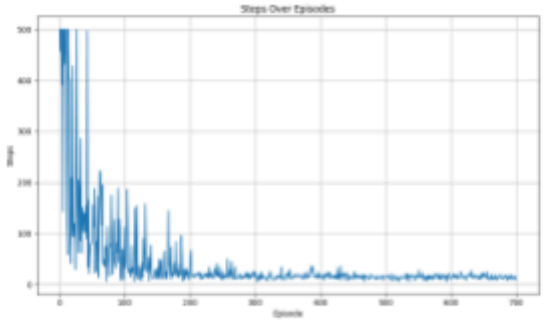
Com podem observar, el decay més baix (0.9) fa que les rewards siguin més divergents, que acaben oscil·lant entre valors de -100 i 0 aproximadament. Com podem veure, a mesura que augmentem el decay, i que per tant el learning rate decreix més lentament, els resultats són més concisos. Veiem que les rewards són menys variables, i que els passo de l'agent també acaben sent més estables. A més, és coherent que a mesura que la reward augmenta, el nombre de passos disminueix.

4.5.4. Epsilon decay

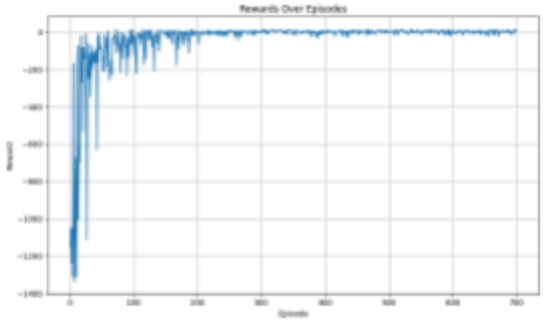
En aquest experiment veurem la influència del epsilon decay en el nostre agent. Com al principi l'agent no té cap coneixement, hem cregut convenient que sigui totalment aleatori, per tant el valor inicial d'epsilon és 1. La nostra hipòtesi és que si NO deixem epsilon amb un valor fixe, i l'anem decreixent, els resultats seran millors perquè al principi hi haurà molta exploració (és possible que trigui més en aprendre), podrà investigar noves trajectòries, però a mesura que passi el temps, anirà explotant més per garantir una convergència cap a una política òptima.

Compararem els resultats d'aplicar un valor fixe d'epsilon = 0.1, i també diferents decays sobre un valor de epsilon que començarà en 1. Valorarem l'evolució del reward i dels passos durant l'entrenament, i avaluarem quin percentatge d'episodis hauran acabat exitosament.

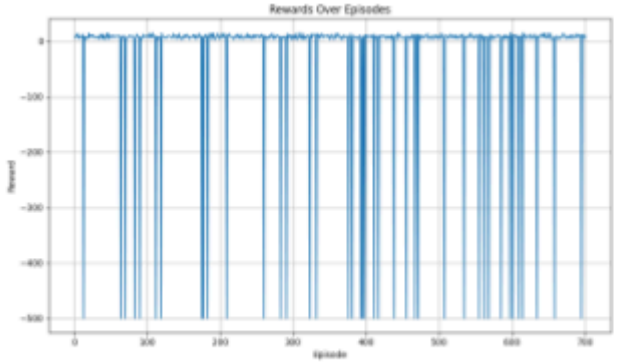
Epsilon fixe = **0.1 (SENSE DECAY)**



(a) Passos durant l'entrenament

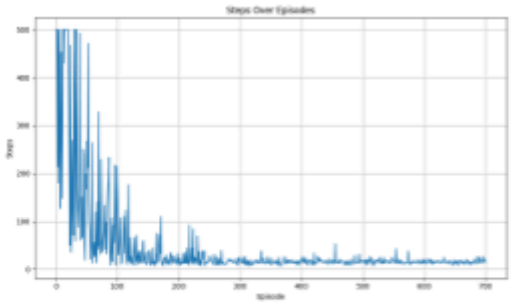


(b) Rewards durant l'entrenament

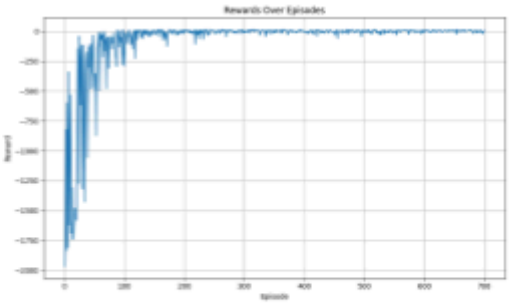


(c) Reward del model entrenat.
94% episodis acabats

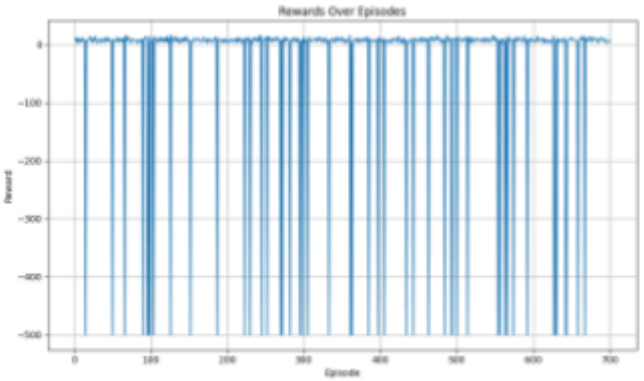
Epsilon inicial = 1; **Decay = 0.9**; Epsilon mínim = 0.1



(a) Passos durant l'entrenament

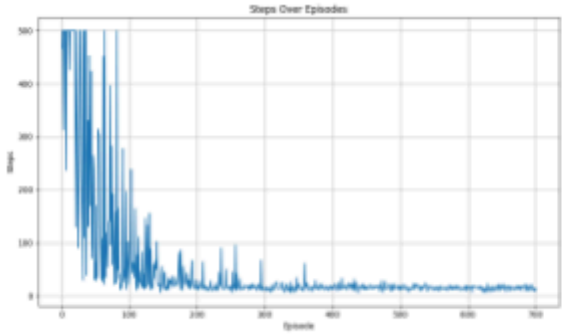


(b) Rewards durant l'entrenament

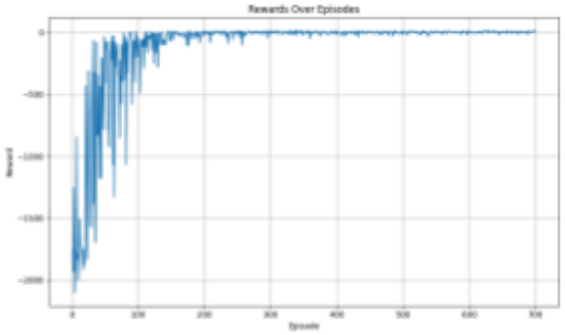


(c) Reward del model entrenat.
93% episodis acabats

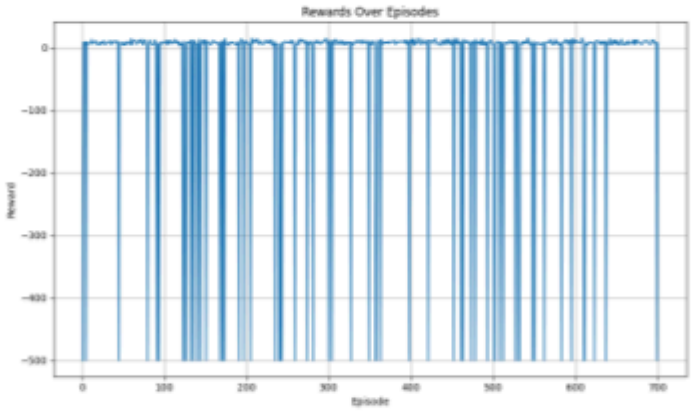
Epsilon inicial = 1; **Decay = 0.95**; Epsilon mínim = 0.1



(a) Passos durant l'entrenament

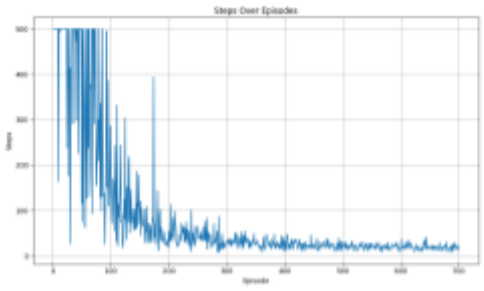


(b) Rewards durant l'entrenament

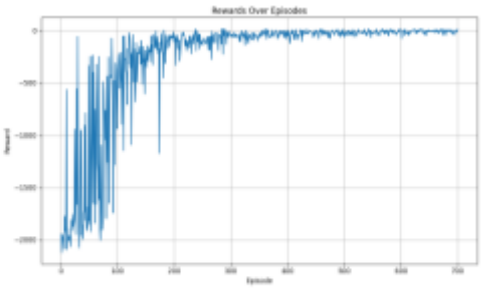


(c) Reward del model entrenat.
92% episodis acabats

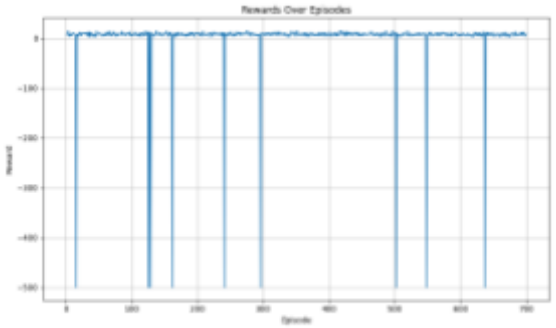
Epsilon inicial = 1; **Decay = 0.99**; Epsilon mínim = 0.1



(a) Passos durant l'entrenament



(b) Rewards durant l'entrenament

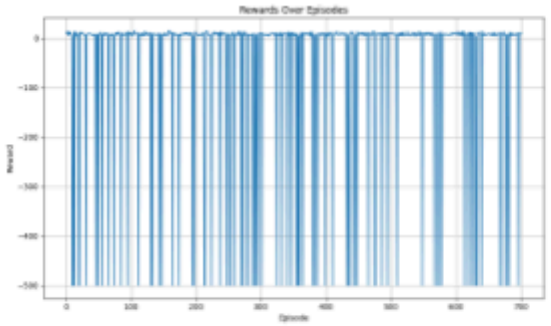


(c) Reward del model entrenat.
99% episodis acabats

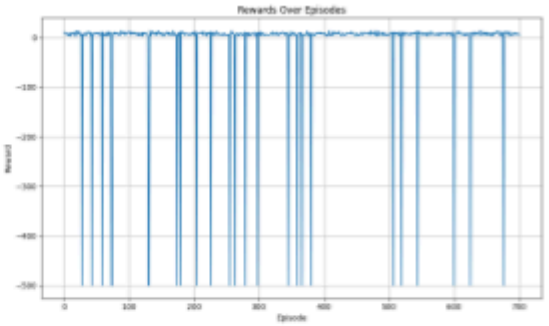
Com podem veure, la corba de reward amb decay = 0.9 ens mostra que l'agent amb aquella configuració aprèn més ràpid perquè les rewards arriben al seu màxim i s'estabilitzen abans. En canvi, utilitzant decay=0.99, les rewards inicials són molt més divergents degut a que mantenim més aleatorietat a l'inici de l'entrenament. No obstant això, si ens fixem en els gràfics (c), on s'avalua el model, podem veure que la opció de decay=0.99 acaba el 99% dels episodis, essent una xifra superior a la de la resta. Per tant, encara que el model amb decay=0.99 sigui més divergent a l'inici, l'exploració de noves trajectòries aleatòries li permet a la llarga obtenir un aprenentatge de millor qualitat i amb millors resultats que la resta.

4.5.5. Factor de descompte (gamma)

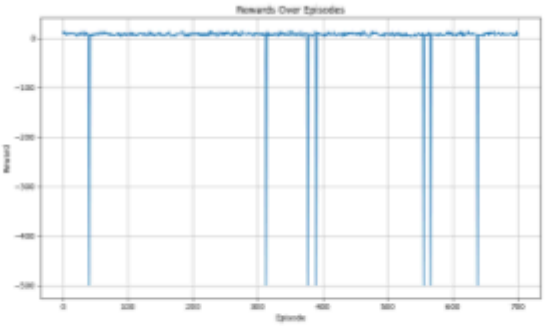
En aquest experiment variarem els valors del factor de descompte i veurem la seva influència sobre el model.
En aquest problema del *Taxi-v3*, les recompenses depenen molt dels pròxims estats en el futur als quals volem arribar. Per tant, la nostra hipòtesi és que contra més importància li donem als etats futurs, la qual cosa es tradueix amb una gamma més alta, millor aprendrà el nostre model.



(a) gamma = 0.7
88% episodis acabats



(b) gamma = 0.9
96% episodis acabats



(c) gamma = 0.99
99% episodis acabats

Com podem veure en els resultats, els rewards són similars independentment de la gamma utilitzada, però el que sí que podem constatar és que, a mesura que el factor de descompte

augmenta (es prenen més en compte els estats futurs), el nombre d'episodis que acaben satisfactòriament augmenta.

Conclusions Q-Learning

Q-Learning és un algoritme de reinforcement learning no basat en model, que pot ser molt útil si no tenim la formalització completa d'un MDP. En aquest cas, només ens fixem en les rewards i estats que ens retorna l'entorn de *Taxi-v3*, sense necessitat de saber com es simulen internament. Q-Learning necessita calcular el Qvalue de cada acció possible sobre cada estat, la qual cosa comporta avantatges i inconvenients. Un model ben parametritzat pot convergir cap a una política bastant òptima, però si l'espai d'observacions i d'accions són molt grans, podem tenir problemas d'escalabilitat (entrenaments llargs i costosos). En el cas de l'entorn de *Taxi-v3*, el tamany d'aquests espais no és excessivament gran per Q-Learning, perquè hem pogut entrenar amb molts episodis en un temps considerable, per tant Q-Learning és una bona opció per aquest entorn. Amb una experimentació més extensiva, utilitzant per exemple grid search o altres mètodes, es podrien trobar hiperparàmetres que millorin encara més els resultats trobats.

5. Reinforce

5.1. Introducció

En aquesta secció, s'analitzarà el rendiment de l'algoritme REINFORCE en l'entorn Taxi-v3 de Gymnasium. Es detallaran els paràmetres més importants per a aquest mètode, es proposaran experiments per avaluar el seu rendiment, i es discutiran els avantatges i desavantatges del mètode en general i en el context específic de l'entorn Taxi-v3.

5.2. Paràmetres importants

1. Nombre d'Episodis (TRAINING_EPISODES):

Aquest paràmetre determina quantes vegades s'executarà l'agent en l'entorn per recopilar dades i millorar la seva política. Un major nombre d'episodis permet a l'agent explorar més l'entorn i obtenir una millor estimació dels valors esperats de les accions en cada estat. En aquest estudi, es consideren diversos valors per avaluar el rendiment inicial de l'agent.

2. Factor de Descompte (GAMMA):

Aquest paràmetre (gamma) determina la importància de les recompenses futures en comparació amb les recompenses immediates. Un valor proper a 1 significa que es consideren fortament les recompenses futures, mentre que un valor proper a 0 fa que l'agent es concentri més en les recompenses immediates.

3. Coeficient d'Exploració (EPSILON):

Aquest paràmetre determina la probabilitat que l'agent explori noves accions en lloc d'explotar la seva política actual. Un valor alt d'epsilon promou més exploració, mentre que un valor baix promou més explotació.

4. Tasa d'Aprenentatge (ALPHA):

Aquest paràmetre determina la velocitat a la qual l'agent actualitza els seus valors esperats basats en noves experiències. Un valor alt d'alpha fa que l'agent s'adapti ràpidament a noves experiències, mentre que un valor baix fa que l'agent aprengui de manera més gradual.

5.3. REINFORCE: Explicació general

L'algoritme REINFORCE és un mètode de gradient de política que actualitza directament els paràmetres de la política de l'agent per maximitzar la recompensa total esperada. En lloc d'aprendre una funció de valor, REINFORCE aprèn una política que mapeja directament els estats a accions. Aquest mètode utilitza el gradient ascendent per ajustar els paràmetres de la política en funció de les recompenses observades.

5.3.1. Avantatges

- Simplicitat: És fàcil d'implementar i entendre.
- Aprenentatge Directe de la Política: Actualitza directament la política de l'agent, el que pot resultar en una convergència més ràpida en alguns casos.
- Eficient en Espais Continus: Pot ser més efectiu en espais d'accions continus comparat amb mètodes basats en valors.

5.3.2. Inconvenients

- Variància Alta: Les actualitzacions basades en el gradient poden tenir alta variància, el que pot fer que l'aprenentatge sigui inestable.
- Dependència del Factor de Descompte: La qualitat de les polítiques apreses pot dependre fortament del valor del factor de descompte.
- Requereix Exploració Eficient: Per a un bon rendiment, l'agent ha d'explorar de manera eficient el seu espai d'estats i accions.

5.4. Reinforce aplicat a l'entorn Taxi-v3

En l'entorn Taxi-v3, l'algoritme REINFORCE es pot aplicar recopilant trajectòries on l'agent executa diferents accions en diversos estats del joc, i després utilitzant aquestes trajectòries per actualitzar directament la política de l'agent. A diferència dels mètodes basats en valors, REINFORCE aprèn a mapejar directament els estats a les accions mitjançant l'ús de gradient ascendent per maximitzar la recompensa total esperada. Això es fa ajustant els paràmetres de la política en funció de les recompenses observades.

L'entorn Taxi-v3 té un espai d'estats i accions considerablement gran, el que pot requerir una exploració eficient i una gran quantitat de dades per obtenir bones estimacions. REINFORCE pot ser particularment avantatjós en aquest entorn perquè permet actualitzar directament la política, el que pot resultar en una convergència més ràpida en alguns casos. No obstant això, la variància alta en les actualitzacions de gradient pot introduir inestabilitat en l'aprenentatge, requerint ajustaments en els paràmetres d'entrenament com el nombre d'episodis, el factor de descompte, el coeficient d'exploració i la tasa d'aprenentatge per millorar la consistència i la qualitat de les prediccions.

La complexitat de l'entorn i la variabilitat en les trajectòries poden afectar les estimacions, per la qual cosa és essencial ajustar aquests paràmetres de manera adequada per evitar el sobreajustament a les trajectòries específiques utilitzades per a l'entrenament, i assegurar que la política apresa pugui generalitzar-se a noves situacions dins de l'entorn.

5.5. Experimentació

Variació del Nombre d'Episodis:

Objectiu: Avaluar com el rendiment de l'agent millora amb un major nombre d'episodis.
Hipòtesi: Augmentar el nombre d'episodis hauria de millorar la política resultant, però amb rendiments decreixents després d'un cert punt.

Variació del Factor de Descompte (γ):

Objectiu: Determinar l'efecte del factor de descompte en el rendiment de l'agent.
Hipòtesi: Valors intermedis de γ (per exemple, 0.9) podrien equilibrar millor les recompenses immediates i futures, resultant en una millor política.

Variació del Coeficient d'Exploració (EPSILON):

Objectiu: Determinar l'efecte del coeficient d'exploració en l'aprenentatge de l'agent.
Hipòtesi: Valors intermedis d'epsilon podrien equilibrar millor l'exploració i l'explotació, resultant en una millor política.

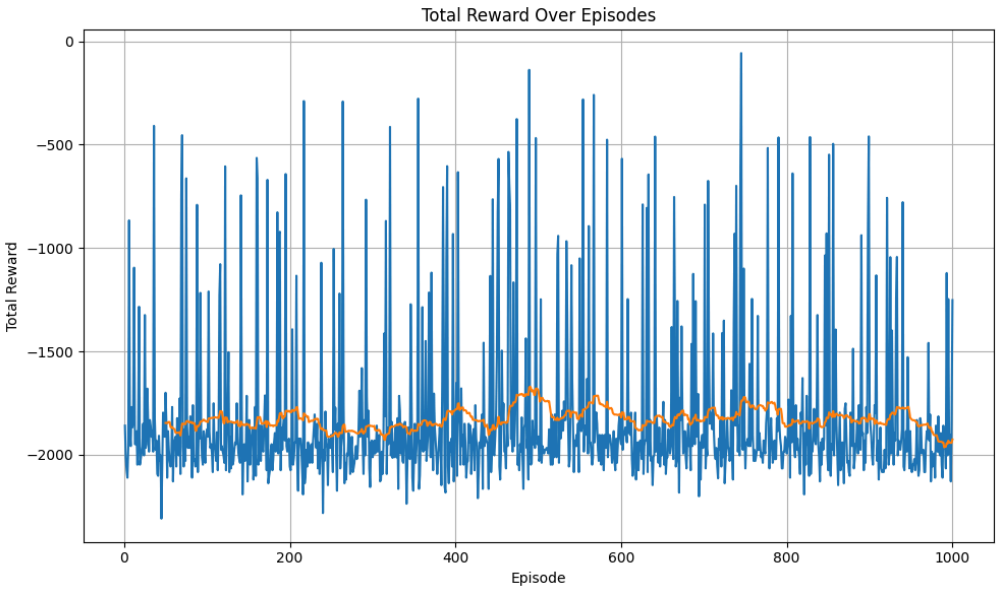
Variació de la Tasa d'Aprenentatge (ALPHA):

Objectiu: Determinar l'efecte de la tasa d'aprenentatge en l'adaptació de l'agent.
Hipòtesi: Valors intermedis d'alpha podrien equilibrar millor l'adaptació ràpida i l'aprenentatge gradual, resultant en una millor política.

Experimentació

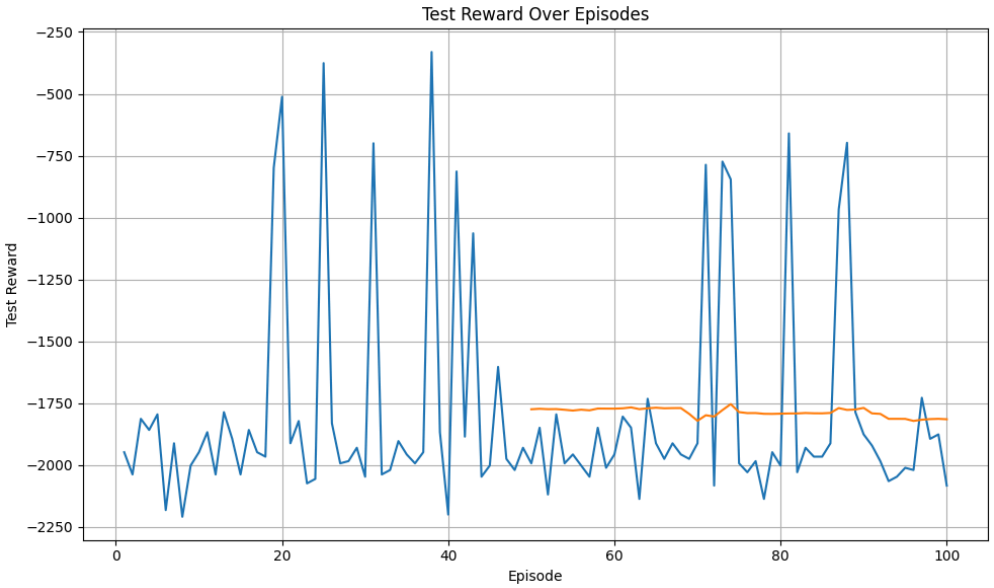
A la pràctica, no hem tingut bons resultats amb REINFORCE, l'agent no ha sigut capaç d'aprendre al llarg dels episodis. Un dels experiments que podem fer és ampliar el nombre d'episodis actual (1000) per veure si necessita més episodis per aprendre, però el temps d'entrenament del model és bastant lent i no ens ha sigut viable ampliar el nombre d'episodis.

De totes formes, compartim les gràfiques resultants de les recompenses que hem obtingut.
La següent gràfica mostra la reward durant l'entrenament. Com es pot observar, no es detecta una tendència alcista, pel que podem concloure que, almenys amb 1000 episodis (no podem investigar-ne amb més), l'agent no és capaç d'aprendre.



Com hem pogut veure en la gràfica anterior, l'agent no ha sigut capaç d'aprendre, per tant, si ara l'avaluem, és d'esperar que les rewards no seran bones.

Avaluem el model amb 100 iteracions. La següent gràfica mostra el resultat d'aquesta avaluació. Podem veure que la recompensa més alta està entre -500 i -250, la qual no és bona, i la majoria de rewards obtingudes durant els 100 episodis d'avaluació oscil·len entre els valors -2000 i -1750. Aquestes rewards són molt dolentes, però eren el que ens esperàvem després de veure en la primera gràfica que l'agent no ha sigut capaç d'aprendre.



REFERÈNCIES

[1]: FrozenLake: https://gymnasium.farama.org/environments/toy_text/frozen_lake/
[2]: Taxi: https://gymnasium.farama.org/environments/toy_text/taxi/