ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ
DEMOCRITUS UNIVERSITY OF THRACE

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΝΕΥΡΟΜΟΡΦΙΚΗ ΕΝΙΣΧΥΤΙΚΗ ΜΑΘΗΣΗ ΓΙΑ ΑΥΤΟΝΟΜΟΥΣ ΡΟΜΠΟΤΙΚΟΥΣ ΧΕΙΡΙΣΤΕΣ

Διπλωματική Εργασία

Καψαχείλης Δημήτριος
Α.Ε.Μ :57085

Επιβλέπων Καθηγητής : Ηλίας Κοσματόπουλος, Καθηγητής Δ.Π.Θ.

Ξάνθη, Οκτώβριος 2023

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ
DEMOCRITUS UNIVERSITY OF THRACE

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΝΕΥΡΟΜΟΡΦΙΚΗ ΕΝΙΣΧΥΤΙΚΗ ΜΑΘΗΣΗ ΓΙΑ ΑΥΤΟΝΟΜΟΥΣ ΡΟΜΠΟΤΙΚΟΥΣ ΧΕΙΡΙΣΤΕΣ

Διπλωματική Εργασία

Καψαχείλης Δημήτριος
Α.Ε.Μ :57085

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ:

Επιβλέπων Καθηγητής : Ηλίας Κοσματόπουλος, Καθηγητής
2ο Μέλος : Ιωάννης Μπούταλης, Καθηγητής
3ο Μέλος : Αθανάσιος Καρλής, Αναπληρωτής Καθηγητής

Ξάνθη, Οκτώβριος 2023

DEMOCRITUS UNIVERSITY OF THRACE
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
SECTOR OF ELECTRONICS AND INFORMATION TECHNOLOGY SYSTEMS

NEUROMORPHIC REINFORCEMENT LEARNING FOR AUTONOMOUS ROBOTIC CONTROLLERS

Diploma Thesis

Kapsacheilis Dimitrios
Registration Number : 57085

Committee of Examiners:

Supervisor: Elias Kosmatopoulos, Professor
Member 2: Yiannis Boutalis, Professor
Member 3: Athanasios Karlis, Associate Professor

Xanthi, October 2023

Περίληψη

Στόχος της διπλωματικής είναι ένας πράκτορας να φτάσει σε συγκεκριμένες θέσεις στον τρισδιάστατο χώρο. Ο χώρος αυτός περιέχει ένα ρομπότ , τοίχους , έναν κώνο και έναν άνθρωπο. Ο πράκτορας είναι ένα ρομπότ το οποίο έχει δυνατότητα κίνησης στον τρισδιάστατο χώρο καθώς και αισθητήρες απόστασης ώστε να γνωρίζει πόσο κοντά βρίσκεται σε άλλα αντικείμενα. Το ρομπότ πρέπει να βρει τρόπο να φτάσει στην επιθυμητή τελική θέση αποφεύγοντας σύγκρουση με τους τοίχους που βρίσκονται στο επίπεδο. Οι τοίχοι έχουν τοποθετηθεί έτσι ώστε να προσδίδουν ένα συγκεκριμένο επίπεδο δυσκολίας. Υπάρχουν τέσσερα επίπεδα δυσκολίας με το ένα να είναι το πιο απλό. Πρώτος στόχος του πράκτορα είναι να βρει τον κώνο και έπειτα να φτάσει στην θέση του ανθρώπου χωρίς να ακουμπήσει κάποιον τοίχο ενδιάμεσα. Η προσομοίωση του χώρου γίνεται μέσω του Gazebo , ενώ η αποστολή εντολών στο ρομπότ γίνεται μέσω του ROS. Γίνεται χρήση διάφορων αλγορίθμων ενισχυτικής εκμάθησης μέσω του Spinning Up. Για να μπορέσει το ρομπότ στην προσομοίωση του gazebo να γίνει πράκτορας ενισχυτικής εκμάθησης στο Spinning Up πρέπει πρώτα το περιβάλλον να έρθει σε μια μορφή συμβατή με το OpenAI Gym. Η βιβλιοθήκη που τα ενώνει αυτά είναι η gym_gazebo_kinetic.
Στο δεύτερο κεφάλαιο θα γίνει αναφορά των εργαλείων και μεθόδων που χρησιμοποιήθηκαν κατά τη διάρκεια της διπλωματικής.
Στο κεφάλαιο 3 ορίζονται διάφορες έννοιες και η μαθηματική τους διατύπωση.
Στο κεφάλαιο 4 περιγράφονται αναλυτικότερα κάποια σημαντικά σημεία του κώδικα. Στο κεφάλαιο 5 δείχνονται τα αποτελέσματα των εκπαιδευμένων αλγορίθμων.
Στο κεφάλαιο 6 γίνεται συζήτηση για τα συμπεράσματα, τις δυσκολίες και τις πιθανές μελλοντικές προεκτάσεις της παρούσας δουλειάς.

Λέξεις-κλειδιά: Ενισχυτική εκμάθηση , Ρομποτική , Τεχνητή Νοημοσύνη

Abstract

The goal of this thesis is for an agent to reach specific positions in 3D space. This space contains a robot, walls, a cone and a human. The agent is a robot that has the ability to move in 3D space as well as distance sensors to know how close it is to other objects. The robot must find a way to reach the desired final position while avoiding collision with the walls on the plane. The walls are placed to give a certain level of difficulty. There are four difficulty levels with one being the easiest. The agent's first goal is to find the cone and then reach the human's location without touching any wall in between. The simulation of the space is done through Gazebo, while sending commands to the robot is done through ROS. Various reinforcement learning algorithms are used through Spinning Up. In order for the robot in the gazebo simulation to become a reinforcement learning agent in Spinning Up, the environment must first be in a format compatible with OpenAI Gym. The library that brings these together is gym_gazebo_kinetic.
In the second chapter, the tools and methods used during the thesis will be mentioned.
In chapter 3 various concepts and their mathematical formulation are defined.
In chapter 4 some important points of the code are described in more detail.
In chapter 5 the results of the trained algorithms are shown.
Chapter 6 discusses the conclusions, difficulties and possible future extensions of the present work.

Keywords : Reinforcement Learning , Robotics , Artificial Intelligence

Περιεχόμενα

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τα ακόλουθα άτομα για τη συμβολή τους σε αυτή τη διπλωματική:

Πρώτα και κύρια, είμαι ευγνώμων στον συνεργάτη μου, Μανώλη Ράπτη, για την καθοδήγηση και την υποστήριξή του σε όλη αυτή την έρευνα. Η τεχνογνωσία και τα σχόλιά του ήταν πολύτιμα στη διαμόρφωση της κατεύθυνσης αυτής της διπλωματικής.

Θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου , Ηλία Κοσματόπουλο, για τον χρόνο και την πολύτιμη συμβολή του. Τα σχόλιά του βοήθησαν στη βελτίωση της ποιότητας αυτής της εργασίας.

Είμαι ευγνώμων στο Δ.Π.Θ. για την παροχή των απαραίτητων πόρων και εγκαταστάσεων που υποστήριξαν αυτήν την έρευνα.

Θα ήθελα να ευχαριστήσω τους συμμετέχοντες που συνεισέφεραν γενναιόδωρα τον χρόνο και τις γνώσεις τους σε αυτή τη μελέτη. Η συμμετοχή τους υπήρξε καθοριστική για την επικύρωση των ευρημάτων.

Θα ήθελα επίσης να ευχαριστήσω τους ερευνητές και τους μελετητές στον τομέα της ενισχυτικής μάθησης των οποίων η εργασία έθεσε τα θεμέλια για αυτήν την έρευνα.

Τέλος, ευχαριστώ την οικογένεια και τους φίλους μου για την υποστήριξή τους και την κατανόησή τους σε όλη αυτή την ακαδημαϊκή διαδρομή.

Εκφράζω τις ειλικρινείς μου ευχαριστίες σε όλους όσους αναφέρθηκαν παραπάνω για τη συμβολή τους στην παρούσα διπλωματική εργασία.

Λεξιλόγιο

Αρκτικόλεξο	Πλήρες όνομα
AI	Artificial Intelligence
RL	Reinforcement Learning
ML	Machine Learning
MDP	Markov Decision Process
ROS	Robot Operating System
PPO	Proximal Policy Optimization
TRPO	Trust Region Policy Optimization
VPG	Vanilla Policy Gradient

1 Εισαγωγή

1.1 Artificial Intelligence - Τεχνητή Νοημοσύνη

Η Τεχνητή Νοημοσύνη (AI) είναι ένας διεπιστημονικός τομέας που εστιάζει στην ανάπτυξη ευφυών μηχανών ικανών να εκτελούν εργασίες που απαιτούν συνήθως ανθρώπινη νοημοσύνη. Η τεχνητή νοημοσύνη περιλαμβάνει ένα ευρύ φάσμα τεχνικών, αλγορίθμων και μεθοδολογιών που επιτρέπουν στους υπολογιστές να προσομοιώνουν γνωστικές λειτουργίες, όπως η αντίληψη, ο συλλογισμός, η μάθηση και η επίλυση προβλημάτων. Μέσω τεχνικών τεχνητής νοημοσύνης έχουμε καταφέρει να φτιάξουμε αλγορίθμους που παίζουν παιχνίδια καλύτερα από τον άνθρωπο. Μερικά από αυτά τα παιχνίδια είναι τα πιο περίπλοκα επιτραπέζια όπως το Go[1], το σκάκι και το Shogi[2] , καθώς και τα ακόμα πιο περίπλοκα ηλεκτρονικά παιχνίδια Starcraft 2[3] και Dota 2[4] . Ο τομέας της τεχνητής νοημοσύνης έχει γνωρίσει σημαντικές προόδους τα τελευταία χρόνια, οδηγώντας σε μετασχηματιστικές εφαρμογές σε διάφορους τομείς, συμπεριλαμβανομένης της υγειονομικής περίθαλψης, των οικονομικών, της ρομποτικής και της επεξεργασίας φυσικής γλώσσας. Στον πυρήνα της, η τεχνητή νοημοσύνη στοχεύει στη δημιουργία συστημάτων που μπορούν να αντιλαμβάνονται και να κατανοούν το περιβάλλον τους, να συλλογίζονται και να λαμβάνουν τεκμηριωμένες αποφάσεις, να μαθαίνουν από την εμπειρία και να επικοινωνούν αποτελεσματικά. Η τεχνητή νοημοσύνη περιλαμβάνει πολλά υποπεδία, συμπεριλαμβανομένης της μηχανικής εκμάθησης, της επεξεργασίας φυσικής γλώσσας, της όρασης υπολογιστών, των έμπειρων συστημάτων και της ρομποτικής. Αυτά τα υποπεδία λειτουργούν σε συνέργεια για την ανάπτυξη ευφυών συστημάτων που μπορούν να αναλύουν πολύπλοκα δεδομένα, να αναγνωρίζουν μοτίβα, να εξάγουν γνώση και να αλληλεπιδρούν με τους ανθρώπους ή το περιβάλλον τους. Η μηχανική μάθηση παίζει κρίσιμο ρόλο στην τεχνητή νοημοσύνη, επιτρέποντας στα συστήματα να μαθαίνουν από δεδομένα και να βελτιώνουν την απόδοσή τους με την πάροδο του χρόνου. Εκπαιδεύοντας μοντέλα σε μεγάλα σύνολα δεδομένων, οι αλγόριθμοι μηχανικής μάθησης μπορούν να αποκαλύψουν κρυφά μοτίβα, να κάνουν ακριβείς προβλέψεις και να αυτοματοποιήσουν πολύπλοκες εργασίες. Η βαθιά μάθηση, ένα υποπεδίο της μηχανικής μάθησης, αξιοποιεί τα νευρωνικά δίκτυα με πολλαπλά επίπεδα για την επεξεργασία τεράστιων ποσοτήτων δεδομένων και την επίτευξη αποτελεσμάτων αιχμής σε τομείς όπως η αναγνώριση εικόνας, η επεξεργασία ομιλίας και η κατανόηση φυσικής γλώσσας. Η Επεξεργασία Φυσικής Γλώσσας (NLP) εστιάζει στο να επιτρέπει στους υπολογιστές να κατανοούν και να δημιουργούν ανθρώπινη γλώσσα. Οι τεχνικές NLP επιτρέπουν στις μηχανές να επεξεργάζονται, να ερμηνεύουν και να ανταποκρίνονται σε κείμενο ή ομιλία, ανοίγοντας δρόμους για εφαρμογές όπως εικονικούς βοηθούς, chatbot και συστήματα μετάφρασης γλώσσας. Η όραση υπολογιστή, από την άλλη πλευρά, εξοπλίζει τις μηχανές με την ικανότητα ανάλυσης και κατανόησης οπτικών δεδομένων, επιτρέποντας εργασίες όπως η αναγνώριση αντικειμένων, η τμηματοποίηση εικόνων και η αυτόνομη πλοήγηση.Η έρευνα θα περιλαμβάνει την εφαρμογή και αξιολόγηση αλγορίθμων τεχνητής νοημοσύνης αιχμής, όπως μοντέλα βαθιάς μάθησης, τεχνικές επεξεργασίας φυσικής γλώσσας ή αλγόριθμους όρασης υπολογιστή. Επιπλέον, θα διερευνήσουμε

διεπιστημονικές προσεγγίσεις, συνδυάζοντας την τεχνητή νοημοσύνη με άλλους τομείς όπως η ρομποτική, η υγειονομική περίθαλψη ή τα οικονομικά, για την αντιμετώπιση σύνθετων προκλήσεων και την ανάπτυξη καινοτόμων λύσεων. Τα αποτελέσματα αυτής της έρευνας θα συμβάλουν στην πρόοδο της τεχνητής νοημοσύνης, ωθώντας τα όρια του τι είναι δυνατό με τα ευφυή συστήματα. Τελικά, ο στόχος είναι να ξεκλειδώσει τις δυνατότητες της τεχνητής νοημοσύνης για να βελτιώσει τον τρόπο που ζούμε, εργαζόμαστε και αλληλεπιδρούμε με την τεχνολογία.

1.2 Machine Learning - Μηχανική Εκμάθηση

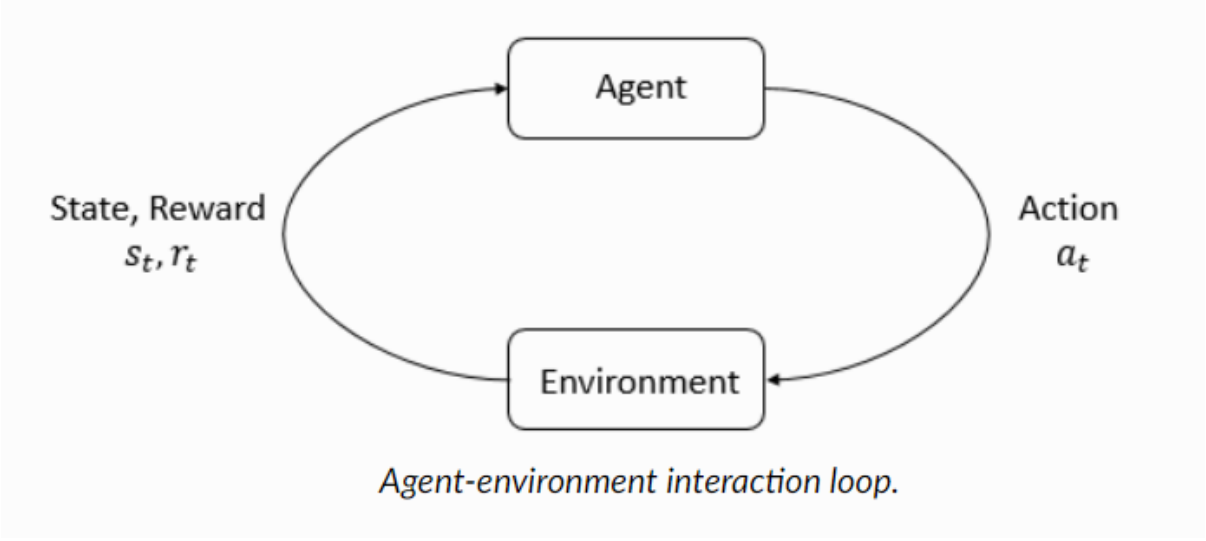
Η μηχανική μάθηση (ML) είναι ένας κλάδος της τεχνητής νοημοσύνης (AI) που εστιάζει στην ανάπτυξη αλγορίθμων και μοντέλων που επιτρέπουν στους υπολογιστές να μαθαίνουν και να κάνουν προβλέψεις ή αποφάσεις χωρίς να είναι ρητά προγραμματισμένοι. Οι αλγόριθμοι ML αξιοποιούν μεγάλες ποσότητες δεδομένων για να προσδιορίσουν μοτίβα, να εξάγουν σημαντικές πληροφορίες και να κάνουν ακριβείς προβλέψεις ή ταξινομήσεις. Την τελευταία δεκαετία, η μηχανική μάθηση έχει φέρει επανάσταση σε διάφορους κλάδους, συμπεριλαμβανομένης της υγειονομικής περίθαλψης, των οικονομικών, των μεταφορών και της επεξεργασίας φυσικής γλώσσας. Στον πυρήνα της, η μηχανική μάθηση περιλαμβάνει τη χρήση μαθηματικών και στατιστικών τεχνικών για την αυτόματη εξαγωγή προτύπων και σχέσεων από δεδομένα. Μαθαίνοντας από παραδείγματα, οι αλγόριθμοι μηχανικής μάθησης μπορούν να γενικεύουν και να λαμβάνουν προβλέψεις ή αποφάσεις για νέα, αόρατα δεδομένα. Οι αλγόριθμοι ML μπορούν να κατηγοριοποιηθούν ευρέως σε εποπτευόμενη μάθηση, μάθηση χωρίς επίβλεψη και ενισχυτική μάθηση. Η εποπτευόμενη μάθηση εστιάζει στη μάθηση από δεδομένα με ετικέτα, όπου κάθε σημείο δεδομένων συσχετίζεται με μια αντίστοιχη ετικέτα ή τιμή στόχο. Ο στόχος είναι να δημιουργηθεί ένα μοντέλο που να μπορεί να αντιστοιχίσει με ακρίβεια τα χαρακτηριστικά εισόδου στις αντίστοιχες εξόδους τους. Οι κοινοί αλγόριθμοι εποπτευόμενης μάθησης περιλαμβάνουν γραμμική παλινδρόμηση, δέντρα αποφάσεων, μηχανές διανυσμάτων υποστήριξης (SVM) και νευρωνικά δίκτυα. Η μη εποπτευόμενη μάθηση, από την άλλη πλευρά, ασχολείται με δεδομένα χωρίς ετικέτα, όπου η αποστολή είναι να αποκαλύψει μοτίβα, δομές ή σχέσεις μέσα στα δεδομένα. Οι αλγόριθμοι ομαδοποίησης, οι τεχνικές μείωσης διαστάσεων και τα παραγωγικά μοντέλα χρησιμοποιούνται συνήθως στην μάθηση χωρίς επίβλεψη για την ανακάλυψη κρυμμένων μοτίβων και την απόκτηση γνώσεων από τα δεδομένα. Η ενισχυτική μάθηση (RL) είναι ένας τύπος μηχανικής μάθησης που εστιάζει στην εκπαίδευση των πρακτόρων για τη λήψη διαδοχικών αποφάσεων σε ένα περιβάλλον για τη μεγιστοποίηση μιας σωρευτικής ανταμοιβής. Εμπνευσμένο από τη συμπεριφορική ψυχολογία, το RL περιλαμβάνει έναν πράκτορα που αλληλεπιδρά με ένα περιβάλλον, μαθαίνει από την ανατροφοδότηση με τη μορφή ανταμοιβών ή ποινών και βελτιστοποιεί τη στρατηγική λήψης αποφάσεων με την πάροδο του χρόνου. Η RL ήταν επιτυχημένη σε τομείς όπως η ρομποτική, η αναπαραγωγή παιχνιδιών και τα αυτόνομα συστήματα. Αξιοποιώντας τη δύναμη των τεχνικών μηχανικής μάθησης, στοχεύουμε στην εξαγωγή πολύτιμων πληροφοριών από πολύπλοκα και υψηλών

διαστάσεων σύνολα δεδομένων, τη βελτίωση της ακρίβειας πρόβλεψης και τη βελτίωση των διαδικασιών λήψης αποφάσεων. Η έρευνα θα περιλαμβάνει την υλοποίηση και αξιολόγηση αλγορίθμων ML τελευταίας τεχνολογίας, όπως αρχιτεκτονικές βαθιάς μάθησης, μεθόδους συνόλου ή πιθανολογικά μοντέλα. Επιπλέον, θα διερευνήσουμε προηγμένες τεχνικές, συμπεριλαμβανομένης της μάθησης μεταφοράς, της μετα-μάθησης και της εξηγήσιμης τεχνητής νοημοσύνης, για να αντιμετωπίσουμε τις προκλήσεις και να βελτιώσουμε την ερμηνευτικότητα και την ευρωστία των μοντέλων ML

1.3 Reinforcement Learning - Ενισχυτική εκμάθηση

Η ενισχυτική μάθηση (RL) είναι ένα υποπεδίο της μηχανικής μάθησης που εστιάζει στο να δίνει τη δυνατότητα σε έναν αυτόνομο παράγοντα να μαθαίνει και να λαμβάνει αποφάσεις μέσω της αλληλεπίδρασης με το περιβάλλον του. Είναι εμπνευσμένο από το πώς οι άνθρωποι και τα ζώα μαθαίνουν από τη δοκιμή και το λάθος για να μεγιστοποιήσουν τις ανταμοιβές και να επιτύχουν στόχους. Η ενισχυτική μάθηση έχει κερδίσει σημαντική προσοχή τα τελευταία χρόνια λόγω των δυνατοτήτων της στην αντιμετώπιση σύνθετων προβλημάτων λήψης αποφάσεων σε διάφορους τομείς, όπως η ρομποτική, το παιχνίδι, τα οικονομικά και η υγειονομική περίθαλψη.

Στον πυρήνα της, η ενισχυτική μάθηση περιλαμβάνει έναν παράγοντα που αλληλεπιδρά με ένα περιβάλλον. Ο πράκτορας μαθαίνει να αναλαμβάνει ενέργειες με βάση την τρέχουσα κατάσταση του περιβάλλοντος και λαμβάνει ανατροφοδότηση με τη μορφή ανταμοιβών ή ποινών. Ο στόχος του πράκτορα είναι να μάθει μια πολιτική, μια χαρτογράφηση από καταστάσεις σε ενέργειες, που μεγιστοποιεί τις σωρευτικές ανταμοιβές που λαμβάνονται με την πάροδο του χρόνου. Οι αλγόριθμοι RL χρησιμοποιούν διάφορες τεχνικές για να εξερευνήσουν και να εκμεταλλευτούν το περιβάλλον, να μάθουν από τις εμπειρίες και να βελτιώσουν την πολιτική επαναληπτικά.



Agent-environment interaction loop.

Ένα από τα βασικά πλεονεκτήματα της ενισχυτικής μάθησης είναι η ικανότητά της να χειρίζεται προβλήματα με αραιές ανταμοιβές, καθυστερημένες συνέπειες και πολύπλοκους χώρους δράσης κράτους. Οι αλγόριθμοι RL είναι ικανοί να μάθουν αποτελεσματικές στρατηγικές σε σενάρια όπου ενδέχεται να μην είναι διαθέσιμες ρητές γνώσεις ειδικών ή δεδομένα εκπαίδευσης με ετικέτα. Μέσω των αλληλεπιδράσεων με το περιβάλλον, ο πράκτορας μαθαίνει να γενικεύει από προηγούμενες εμπειρίες, να προσαρμόζεται στις μεταβαλλόμενες συνθήκες και να ανακαλύπτει βέλτιστες ή σχεδόν βέλτιστες πολιτικές.

Η παρούσα διπλωματική εργασία στοχεύει να διερευνήσει και να αναπτύξει τεχνικές ενισχυτικής μάθησης για τον έλεγχο ενός ρομπότ σε ένα δυναμικό και αβέβαιο περιβάλλον. Η εστίαση είναι στην εκπαίδευση ενός πράκτορα RL ώστε να πλοηγείται σε περιβάλλον που μοιάζει με λαβύρινθο, να αποφεύγει εμπόδια και να φτάνει αποτελεσματικά σε καθορισμένες θέσεις στόχου. Αξιοποιώντας τη δύναμη των αλγορίθμων RL, επιδιώκουμε να επιτρέψουμε στο ρομπότ να μαθαίνει αυτόνομα και να βελτιώνει τις ικανότητές του στη λήψη αποφάσεων με την πάροδο του χρόνου.

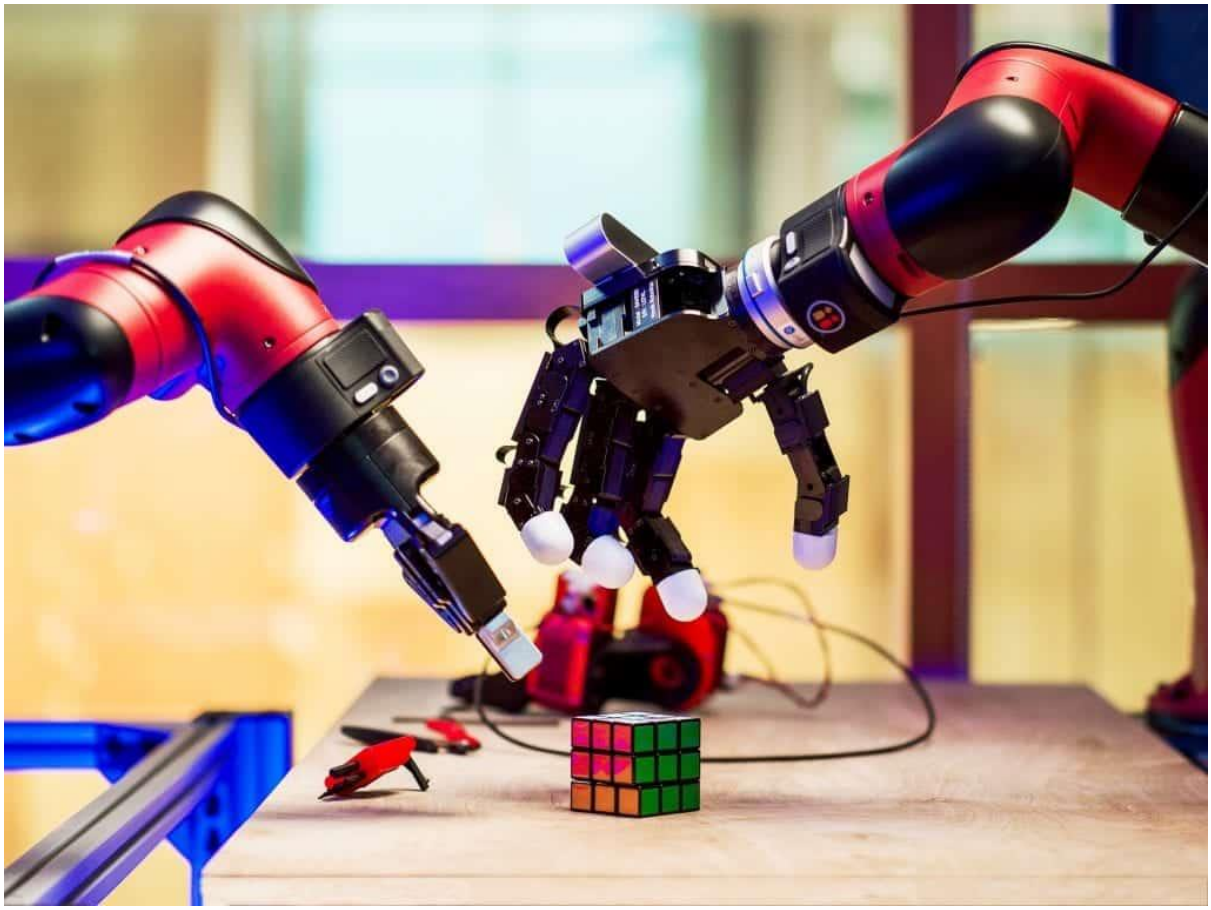
Η έρευνα θα περιλαμβάνει την υλοποίηση και αξιολόγηση αλγορίθμων RL τελευταίας τεχνολογίας, όπως το Proximal Policy Optimization (PPO) και το Deep Q-Networks (DQN), στο πλαίσιο του ελέγχου ρομπότ. Επιπλέον, θα διερευνήσουμε τον αντίκτυπο της διαμόρφωσης ανταμοιβής, των στρατηγικών εξερεύνησης και των τεχνικών εκμάθησης του προγράμματος σπουδών στη μαθησιακή απόδοση και τη σύγκλιση του πράκτορα.

Τα αποτελέσματα αυτής της έρευνας θα συμβάλουν στην πρόοδο του RL σε πραγματικές ρομποτικές εφαρμογές και θα παρέχουν πολύτιμες γνώσεις σχετικά με το σχεδιασμό και τη βελτιστοποίηση αλγορίθμων RL για σύνθετες εργασίες λήψης αποφάσεων. Τελικά, ο στόχος είναι να αναπτυχθούν ισχυρά και προσαρμοστικά συστήματα ελέγχου βασισμένα σε RL που μπορούν να ενισχύσουν την αυτονομία και τις δυνατότητες των ρομποτικών συστημάτων σε ένα ευρύ φάσμα τομέων

1.4 Robotics - Ρομποτική

Η ρομποτική είναι ένας ταχέως εξελισσόμενος τομέας στη διασταύρωση της μηχανικής, της επιστήμης των υπολογιστών και της τεχνητής νοημοσύνης που επικεντρώνεται στο σχεδιασμό, την ανάπτυξη και την ανάπτυξη ευφυών μηχανών ικανών να εκτελούν εργασίες αυτόνομα ή με ανθρώπινη βοήθεια. Τα ρομπότ έχουν τη δυνατότητα να φέρουν επανάσταση στις βιομηχανίες, να ενισχύσουν την ανθρώπινη παραγωγικότητα και να αντιμετωπίσουν τις κοινωνικές προκλήσεις πραγματοποιώντας σύνθετες λειτουργίες σε διάφορους τομείς, συμπεριλαμβανομένης της κατασκευής, της υγειονομικής περίθαλψης, της εξερεύνησης και των υπηρεσιών. Ο τομέας της ρομποτικής περιλαμβάνει ένα ευρύ φάσμα επιστημονικών κλάδων, συμπεριλαμβανομένων των μηχανολόγων μηχανικών, ηλεκτρολόγων μηχανικών, επιστήμης υπολογιστών, συστημάτων ελέγχου και τεχνητής νοημοσύνης. Περιλαμβάνει το σχεδιασμό φυσικών ρομποτικών συστημάτων με αισθητήρες και

ενεργοποιητές, την ανάπτυξη αλγορίθμων για την αντίληψη, τον προγραμματισμό και τον έλεγχο και την ενσωμάτωση στοιχείων λογισμικού και υλικού για να καταστεί δυνατή η έξυπνη συμπεριφορά. Τα ρομπότ έχουν σχεδιαστεί για να αλληλεπιδρούν με τον φυσικό κόσμο, να αντιλαμβάνονται το περιβάλλον τους χρησιμοποιώντας αισθητήρες όπως κάμερες, lidar ή αισθητήρες αφής και να λαμβάνουν τεκμηριωμένες αποφάσεις με βάση τα δεδομένα που συλλέγονται. Μπορούν να εκτελούν εργασίες όπως χειρισμό, πλοήγηση, αναγνώριση αντικειμένων, αλληλεπίδραση ανθρώπου-ρομπότ, ακόμη και αυτόνομη λήψη αποφάσεων σε πολύπλοκα και δυναμικά περιβάλλοντα.



Οι εξελίξεις στην τεχνητή νοημοσύνη, ιδιαίτερα στη μηχανική μάθηση και τη βαθιά μάθηση, έχουν επηρεάσει σημαντικά τη ρομποτική. Εκπαιδεύοντας ρομπότ σε μεγάλα σύνολα δεδομένων ή χρησιμοποιώντας τεχνικές ενισχυτικής μάθησης, μπορούν να μάθουν από την εμπειρία, να προσαρμοστούν σε μεταβαλλόμενα περιβάλλοντα και να βελτιώσουν την απόδοσή τους με την πάροδο του χρόνου. Αυτό δίνει τη δυνατότητα στα ρομπότ να χειρίζονται πολύπλοκες εργασίες, να αποκτούν νέες δεξιότητες και να αλληλεπιδρούν πιο φυσικά με τους ανθρώπους.. Αξιοποιώντας τις αρχές της ρομποτικής, της τεχνητής νοημοσύνης και των συστημάτων ελέγχου, επιδιώκουμε να δημιουργήσουμε ρομπότ που να μπορούν να λειτουργούν αυτόνομα, να αλληλεπιδρούν με ανθρώπους ή να συνεργάζονται με άλλα ρομπότ για να ολοκληρώσουν πολύπλοκες εργασίες. Η έρευνα θα περιλαμβάνει το σχεδιασμό και την ανάπτυξη ρομποτικού υλικού, την εφαρμογή προηγμένων αλγορίθμων αντίληψης, την ενοποίηση τεχνικών σχεδιασμού και ελέγχου και την εξερεύνηση μεθόδων

αλληλεπίδρασης ανθρώπου-ρομπότ. Επιπλέον, θα εξερευνήσουμε αναδυόμενες περιοχές στη ρομποτική, όπως η μαλακή ρομποτική, η ρομποτική σμήνος ή η ρομποτική εμπνευσμένη από βιοτεχνίες, για να αντιμετωπίσουμε μοναδικές προκλήσεις και να ξεπεράσουμε τα όρια των ρομποτικών δυνατοτήτων. Τα αποτελέσματα αυτής της έρευνας θα συμβάλλουν στην πρόοδο της ρομποτικής, φέρνοντάς μας πιο κοντά στην υλοποίηση ευφυών και ευέλικτων ρομποτικών συστημάτων. Δημιουργώντας ρομπότ που μπορούν να εκτελούν εργασίες αποτελεσματικά, με ασφάλεια και προσαρμοστικότητα, στοχεύουμε να ξεκλειδώσουμε τις δυνατότητές τους σε τομείς όπως η βοήθεια στην υγειονομική περίθαλψη, ο βιομηχανικός αυτοματισμός, η εξερεύνηση και η αντιμετώπιση καταστροφών. Μέσω αυτής της διατριβής, προσπαθούμε να γεφυρώσουμε το χάσμα μεταξύ θεωρίας και πράξης, συνδυάζοντας τη θεωρητική γνώση με την πρακτική εφαρμογή για την ανάπτυξη καινοτόμων λύσεων στη ρομποτική. Περνώντας τα όρια του τι μπορούν να επιτύχουν τα ρομπότ, φιλοδοξούμε να φέρουμε επανάσταση στις βιομηχανίες, να βελτιώσουμε την ποιότητα ζωής και να διαμορφώσουμε το μέλλον της αυτοματοποίησης και της συνεργασίας ανθρώπου-ρομπότ.

1.5 Περίληψη

Στόχος της διπλωματικής είναι ένας πράκτορας να φτάσει σε συγκεκριμένες θέσεις στον τρισδιάστατο χώρο. Ο χώρος αυτός περιέχει ένα ρομπότ , τοίχους , έναν κώνο και έναν άνθρωπο. Ο πράκτορας είναι ένα ρομπότ το οποίο έχει δυνατότητα κίνησης στον τρισδιάστατο χώρο καθώς και αισθητήρες απόστασης ώστε να γνωρίζει πόσο κοντά βρίσκεται σε άλλα αντικείμενα. Το ρομπότ πρέπει να βρει τρόπο να φτάσει στην επιθυμητή τελική θέση αποφεύγοντας σύγκρουση με τους τοίχους που βρίσκονται στο επίπεδο. Οι τοίχοι έχουν τοποθετηθεί έτσι ώστε να προσδίδουν ένα συγκεκριμένο επίπεδο δυσκολίας. Υπάρχουν τέσσερα επίπεδα δυσκολίας με το ένα να είναι το πιο απλό. Πρώτος στόχος του πράκτορα είναι να βρει τον κώνο και έπειτα να φτάσει στην θέση του ανθρώπου χωρίς να ακουμπήσει κάποιον τοίχο ενδιάμεσα. Η προσομοίωση του χώρου γίνεται μέσω του Gazebo[5] , ενώ η αποστολή εντολών στο ρομπότ γίνεται μέσω του ROS[6]. Γίνεται χρήση διάφορων αλγορίθμων ενισχυτικής εκμάθησης μέσω του Spinning Up[7]. Για να μπορέσει το ρομπότ στην προσομοίωση του gazebo να γίνει πράκτορας ενισχυτικής εκμάθησης στο Spinning Up πρέπει πρώτα το περιβάλλον να έρθει σε μια μορφή συμβατή με το OpenAI Gym[8] . Η βιβλιοθήκη που τα ενώνει αυτά είναι η gym_gazebo_kinetic[9].
Στο δεύτερο κεφάλαιο θα γίνει αναφορά των εργαλείων και μεθόδων που χρησιμοποιήθηκαν κατά τη διάρκεια της διπλωματικής.
Στο κεφάλαιο 3 ορίζονται διάφορες έννοιες και η μαθηματική τους διατύπωση.
Στο κεφάλαιο 4 περιγράφονται αναλυτικότερα κάποια σημαντικά σημεία του κώδικα.
Στο κεφάλαιο 5 δείχνονται τα αποτελέσματα των εκπαιδευμένων αλγορίθμων.
Στο κεφάλαιο 6 γίνεται συζήτηση για τα συμπεράσματα, τις δυσκολίες και τις πιθανές μελλοντικές προεκτάσεις της παρούσας δουλειάς.

2 Εργαλεία και μέθοδοι

2.1 OpenAI Gym

Το OpenAI Gym είναι μια δημοφιλής εργαλειοθήκη ανοιχτού κώδικα που αναπτύχθηκε από την OpenAI που παρέχει ένα τυποποιημένο και προσβάσιμο πλαίσιο για την ανάπτυξη, την αξιολόγηση και τη σύγκριση αλγορίθμων ενίσχυσης μάθησης. Η ενισχυτική μάθηση, ένα υποπεδίο της μηχανικής μάθησης, εστιάζει στη διδασκαλία των πρακτόρων πώς να λαμβάνουν διαδοχικές αποφάσεις σε ένα περιβάλλον για τη μεγιστοποίηση ενός σήματος ανταμοιβής. Το OpenAI Gym προσφέρει μια ποικιλόμορφη συλλογή προκατασκευασμένων περιβαλλόντων, που κυμαίνονται από κλασικές εργασίες ελέγχου έως πολύπλοκα προσομοιωμένα περιβάλλοντα, επιτρέποντας σε ερευνητές και προγραμματιστές να δοκιμάσουν και να αξιολογήσουν τους αλγόριθμους μάθησης ενίσχυσης με συνεπή και αναπαραγώγιμο τρόπο. Αυτά τα περιβάλλοντα περικλείουν ένα ευρύ φάσμα προβλημάτων, όπως έλεγχος ρομπότ, παιχνίδι, ρομποτική προσομοίωση, ακόμη και επεξεργασία φυσικής γλώσσας. Τα βασικά στοιχεία του OpenAI Gym περιλαμβάνουν τη διεπαφή περιβάλλοντος, η οποία ορίζει την αλληλεπίδραση μεταξύ του πράκτορα και του περιβάλλοντος, και τον πράκτορα, ο οποίος μαθαίνει να αναλαμβάνει ενέργειες με βάση τις παρατηρήσεις από το περιβάλλον και τις ανταμοιβές που λαμβάνει. Το περιβάλλον παρέχει παρατηρήσεις σχετικά με την κατάστασή του και ο πράκτορας μαθαίνει να χαρτογραφεί αυτές τις παρατηρήσεις σε κατάλληλες ενέργειες για να μεγιστοποιήσει τη σωρευτική ανταμοιβή με την πάροδο του χρόνου. Το OpenAI Gym παρέχει ένα ενοποιημένο API που επιτρέπει στους ερευνητές και τους προγραμματιστές να αλληλεπιδρούν εύκολα με διαφορετικά περιβάλλοντα και να εφαρμόζουν διάφορους αλγόριθμους μάθησης ενίσχυσης. Προσφέρει ένα σύνολο τυπικών λειτουργιών για τη λήψη ενεργειών, τη λήψη παρατηρήσεων, τον υπολογισμό ανταμοιβών και την παρακολούθηση της απόδοσης. Αυτή η τυποποίηση απλοποιεί τη διαδικασία ανάπτυξης και επιτρέπει τη σύγκριση διαφορετικών αλγορίθμων σε ίσους όρους ανταγωνισμού. Επιπλέον, το OpenAI Gym υποστηρίζει ένα ευρύ φάσμα αλγορίθμων ενισχυτικής μάθησης, συμπεριλαμβανομένου του Q-learning, των μεθόδων κλίσης πολιτικής, της μάθησης βαθιάς ενίσχυσης και πολλά άλλα. Οι ερευνητές μπορούν να αξιοποιήσουν αυτούς τους αλγόριθμους για να εκπαιδεύσουν τους πράκτορες να μαθαίνουν βέλτιστες πολιτικές σε διάφορα περιβάλλοντα βελτιστοποιώντας διαφορετικές αντικειμενικές συναρτήσεις, όπως μεγιστοποίηση σωρευτικών ανταμοιβών ή ελαχιστοποίηση του αναμενόμενου χρόνου για την επίτευξη μιας κατάστασης στόχου. Η διαθεσιμότητα του OpenAI Gym έχει διευκολύνει σημαντικά την έρευνα και την ανάπτυξη στον τομέα της ενισχυτικής μάθησης. Παρείχε μια κοινή πλατφόρμα για τους ερευνητές να μοιράζονται την εργασία τους, να συγκρίνουν τα αποτελέσματα και να συνεργάζονται για την προώθηση της τελευταίας τεχνολογίας. Επιπλέον, το OpenAI Gym έχει προωθήσει την ανάπτυξη της κοινότητας ενισχυτικής μάθησης προσελκύοντας ενθουσιώδεις και επαγγελματίες από διαφορετικά υπόβαθρα, επιταχύνοντας την πρόοδο σε αυτόν τον τομέα. Ο στόχος είναι να εφαρμοστούν και να αξιολογηθούν διάφοροι αλγόριθμοι σε διαφορετικά περιβάλλοντα OpenAI Gym, αναλύοντας τις επιδόσεις, την αποτελεσματικότητα και τις δυνατότητές τους γενίκευσης. Η έρευνα θα διερευνήσει

επίσης τεχνικές για τη βελτίωση της αποτελεσματικότητας και της σταθερότητας του δείγματος των αλγορίθμων ενισχυτικής μάθησης στο πλαίσιο του OpenAI Gym. Μέσω αυτής της διατριβής, στοχεύουμε να συμβάλουμε στην πρόοδο των τεχνικών ενισχυτικής μάθησης και να αναδείξουμε τις δυνατότητες του OpenAI Gym ως ένα ισχυρό εργαλείο για την επίλυση προβλημάτων του πραγματικού κόσμου. Αξιοποιώντας την πλούσια συλλογή περιβαλλόντων και αλγορίθμων που παρέχει το OpenAI Gym, προσπαθούμε να ξεπεράσουμε τα όρια της ενισχυτικής μάθησης και να ανοίξουμε το δρόμο για πρακτικές εφαρμογές σε τομείς όπως η ρομποτική, το παιχνίδι, τα αυτόνομα συστήματα και άλλα.

2.2 OpenAI Spinning Up

Το OpenAI Spinning Up είναι μια βιβλιοθήκη ανοιχτού κώδικα που αναπτύχθηκε από την OpenAI που στοχεύει να παρέχει έναν πρακτικό και προσβάσιμο πόρο για την εφαρμογή και τον πειραματισμό με υπερσύγχρονους αλγόριθμους ενίσχυσης μάθησης. Η ενισχυτική μάθηση, ένα υποπεδίο της μηχανικής μάθησης, εστιάζει στην εκπαίδευση των πρακτόρων για τη λήψη διαδοχικών αποφάσεων σε ένα περιβάλλον για τη μεγιστοποίηση ενός σωρευτικού σήματος ανταμοιβής. Το OpenAI Spinning Up προσφέρει μια ολοκληρωμένη συλλογή από καλά τεκμηριωμένες και εύχρηστες εφαρμογές κώδικα διαφόρων αλγορίθμων μάθησης ενίσχυσης. Αυτοί οι αλγόριθμοι κυμαίνονται από κλασικές προσεγγίσεις όπως τα βαθιά δίκτυα Q (DQN) και οι κλίσεις πολιτικής έως πιο προηγμένες τεχνικές όπως η εγγύς βελτιστοποίηση πολιτικής (PPO) και η soft actor-critic (SAC). Η βιβλιοθήκη περιλαμβάνει επίσης υποστήριξη τόσο για συνεχείς όσο και για διακριτούς χώρους δράσης, δίνοντας τη δυνατότητα σε ερευνητές και προγραμματιστές να αντιμετωπίσουν ένα ευρύ φάσμα προβλημάτων. Τα βασικά στοιχεία του OpenAI Spinning Up περιλαμβάνουν αρθρωτές υλοποιήσεις αλγορίθμων, διαισθητικά API για εκπαιδευτικούς και δοκιμαστικούς παράγοντες και εργαλεία για οπτικοποίηση και ανάλυση αποτελεσμάτων. Η βιβλιοθήκη παρέχει ένα σαφές και δομημένο πλαίσιο για τον πειραματισμό με διαφορετικούς αλγόριθμους και την προσαρμογή τους σε συγκεκριμένες εργασίες και περιβάλλοντα. Περιλαμβάνει επίσης λεπτομερή τεκμηρίωση, παραδείγματα κώδικα και σεμινάρια, καθιστώντας το προσβάσιμο τόσο σε νεοφερμένους όσο και σε έμπειρους επαγγελματίες στον τομέα. Το OpenAI Spinning Up είναι χτισμένο πάνω σε δημοφιλείς βιβλιοθήκες βαθιάς μάθησης, όπως το TensorFlow και το PyTorch, αξιοποιώντας την υπολογιστική τους απόδοση και ευελιξία. Εκμεταλλεύεται τις τεχνικές παράλληλων υπολογιστών και την κατανεμημένη εκπαίδευση για να επιταχύνει τη διαδικασία μάθησης, επιτρέποντας στους ερευνητές να κλιμακώσουν τα πειράματά τους και να εκπαιδεύσουν τους πράκτορες πιο αποτελεσματικά. Ένα από τα αξιοσημείωτα χαρακτηριστικά του OpenAI Spinning Up είναι η εστίασή του στην παροχή εκπαιδευτικών πόρων και βέλτιστων πρακτικών για ενισχυτική μάθηση. Η βιβλιοθήκη περιλαμβάνει ένα επιμελημένο σύνολο από σεμινάρια, οδηγούς και διδακτικό υλικό που εξηγούν τις θεμελιώδεις έννοιες και τους αλγόριθμους της ενισχυτικής μάθησης με σαφή και προσιτό τρόπο. Αυτή η εκπαιδευτική πτυχή κάνει το OpenAI Spinning Up ένα εξαιρετικό εργαλείο τόσο για ερευνητές όσο και για εκπαιδευτικούς που ενδιαφέρονται για την ενισχυτική μάθηση. Ο στόχος είναι να εφαρμοστούν και να αξιολογηθούν

διαφορετικοί αλγόριθμοι που παρέχονται από το OpenAI Spinning Up σε διάφορες εργασίες αναφοράς, αναλύοντας την απόδοσή τους, τις ιδιότητες σύγκλισης και την αποτελεσματικότητα του δείγματος. Επιπλέον, αυτή η έρευνα θα διερευνήσει τεχνικές για τη βελτίωση της σταθερότητας και της ευρωστίας των αλγορίθμων ενισχυτικής μάθησης στο πλαίσιο του OpenAI Spinning Up. Θα διερευνήσει προσεγγίσεις όπως η διαμόρφωση ανταμοιβής, η εκμάθηση προγράμματος σπουδών, η μεταφορά μάθησης και η βελτιστοποίηση υπερπαραμέτρων για τη βελτίωση της μαθησιακής διαδικασίας και την επίτευξη καλύτερης απόδοσης σε απαιτητικές εργασίες. Μέσω αυτής της διατριβής, στοχεύουμε να συμβάλουμε στην πρόοδο των τεχνικών ενισχυτικής μάθησης και να αναδείξουμε τις δυνατότητες του OpenAI Spinning Up ως ένα ισχυρό εργαλείο για την επίλυση προβλημάτων του πραγματικού κόσμου. Αξιοποιώντας την πλούσια συλλογή αλγορίθμων, εκπαιδευτικών πόρων και πρακτικών εργαλείων που παρέχονται από το OpenAI Spinning Up, προσπαθούμε να ξεπεράσουμε τα όρια της έρευνας για την ενίσχυση της μάθησης και να προωθήσουμε πρακτικές εφαρμογές σε τομείς όπως η ρομποτική, τα αυτόνομα συστήματα, το παιχνίδι και πολλά άλλα.

2.3 Robot Operating System - ROS

Το Robot Operating System (ROS) είναι ένα ευέλικτο και ισχυρό πλαίσιο για την ανάπτυξη ρομποτικών συστημάτων. Παρέχει μια συλλογή βιβλιοθηκών λογισμικού, εργαλείων και συμβάσεων που επιτρέπουν την απρόσκοπτη επικοινωνία, τον συντονισμό και τον έλεγχο διαφόρων στοιχείων μέσα σε ένα ρομπότ ή ένα δίκτυο ρομπότ. Το ROS έχει υιοθετηθεί ευρέως τόσο σε ακαδημαϊκό όσο και σε βιομηχανικό περιβάλλον, χρησιμεύοντας ως θεμελιώδης πλατφόρμα για την έρευνα, την ανάπτυξη και την ανάπτυξη της ρομποτικής. Το ROS προσφέρει μια κατανεμημένη αρχιτεκτονική που επιτρέπει τη σπονδυλωτή ανάπτυξη και ενσωμάτωση στοιχείων λογισμικού ρομπότ. Παρέχει ένα σύνολο τυποποιημένων μηχανισμών μετάδοσης μηνυμάτων, γνωστών ως θέματα ROS, που διευκολύνουν την απρόσκοπτη επικοινωνία μεταξύ διαφορετικών κόμβων σε ένα σύστημα ROS. Αυτή η υποδομή επικοινωνίας επιτρέπει στα ρομπότ να ανταλλάσσουν δεδομένα αισθητήρων, εντολές ενεργοποιητή και άλλες σχετικές πληροφορίες, επιτρέποντας συντονισμένες ενέργειες και συλλογικές συμπεριφορές. Ένα από τα βασικά πλεονεκτήματα του ROS είναι η εκτεταμένη βιβλιοθήκη με προκατασκευασμένα πακέτα και εργαλεία, που καλύπτει ένα ευρύ φάσμα λειτουργιών και εφαρμογών. Αυτά τα πακέτα περιλαμβάνουν έτοιμα προς χρήση προγράμματα οδήγησης για διάφορους αισθητήρες και ενεργοποιητές, αλγόριθμους για αντίληψη και έλεγχο, περιβάλλοντα προσομοίωσης, εργαλεία οπτικοποίησης και άλλα. Αυτό το τεράστιο οικοσύστημα πακέτων απλοποιεί τη διαδικασία ανάπτυξης παρέχοντας στους προγραμματιστές επαναχρησιμοποιήσιμα και καλά δοκιμασμένα εξαρτήματα που μπορούν εύκολα να ενσωματωθούν στα ρομποτικά τους συστήματα. Το ROS προσφέρει επίσης ισχυρές δυνατότητες προσομοίωσης μέσω της ενσωμάτωσης με προσομοιωτές όπως το Gazebo. Αυτό επιτρέπει στους προγραμματιστές να δοκιμάσουν και να επικυρώσουν τους αλγόριθμους και τις συμπεριφορές ρομπότ τους σε ένα προσομοιωμένο περιβάλλον προτού τα αναπτύξουν σε πραγματικό υλικό. Το περιβάλλον προσομοίωσης παρέχει ακριβή μοντελοποίηση φυσικής, προσομοίωση αισθητήρα και οπτικοποίηση,

επιτρέποντας ρεαλιστική και αποτελεσματική δοκιμή ρομποτικών συστημάτων. Ένα άλλο αξιοσημείωτο χαρακτηριστικό του ROS είναι η υποστήριξή του για ετερογενείς πλατφόρμες υλικού και λογισμικού. Το ROS μπορεί να χρησιμοποιηθεί με διάφορες πλατφόρμες ρομπότ, που κυμαίνονται από μικρά κινητά ρομπότ έως μεγάλους βιομηχανικούς χειριστές. Είναι επίσης συμβατό με διαφορετικά λειτουργικά συστήματα, συμπεριλαμβανομένων των Linux και macOS, παρέχοντας ευελιξία και διαλειτουργικότητα σε διάφορα υπολογιστικά περιβάλλοντα. Επιπλέον, το ROS ενθαρρύνει μια ζωντανή και συνεργατική κοινότητα ερευνητών, προγραμματιστών και ενθουσιωδών. Η φύση ανοιχτού κώδικα του ROS ενθαρρύνει την ανταλλαγή γνώσεων, τη συνεισφορά κώδικα και τη συνεργασία, με αποτέλεσμα ένα πλούσιο οικοσύστημα πόρων, σεμιναρίων και υποστήριξης με γνώμονα την κοινότητα. Αυτό το συνεργατικό περιβάλλον επιταχύνει την καινοτομία, προωθεί τις βέλτιστες πρακτικές και επιτρέπει την κοινή χρήση τεχνικών και αλγορίθμων αιχμής στην κοινότητα της ρομποτικής Επιπλέον, αυτή η διατριβή θα διερευνήσει τεχνικές για τη βελτιστοποίηση της απόδοσης, της αξιοπιστίας και της επεκτασιμότητας των ρομποτικών συστημάτων που βασίζονται σε ROS. Θα διερευνήσει θέματα όπως ο σχεδιασμός της αρχιτεκτονικής συστήματος, τα πρωτόκολλα επικοινωνίας, ο έλεγχος σε πραγματικό χρόνο, η σύντηξη αισθητήρων και η κατανεμημένη πληροφορική για τη βελτίωση των δυνατοτήτων των ROS και τη βελτίωση της συνολικής απόδοσης και αποτελεσματικότητας των ρομποτικών εφαρμογών. Μέσω αυτής της έρευνας, στοχεύουμε να συμβάλουμε στην πρόοδο της ρομποτικής αξιοποιώντας τη δύναμη και την ευελιξία των ROS. Αναπτύσσοντας καινοτόμες λύσεις, αξιοποιώντας το εκτεταμένο οικοσύστημα ROS και συνεργαζόμενοι με την κοινότητα των ROS, προσπαθούμε να ξεπεράσουμε τα όρια των ρομποτικών συστημάτων και να ανοίξουμε το δρόμο για πρακτικές και αποτελεσματικές εφαρμογές σε τομείς όπως [ειδικοί τομείς εφαρμογής].

2.4 Gazebo

Το Gazebo είναι ένα ισχυρό και ευέλικτο περιβάλλον προσομοίωσης ρομποτικής ανοιχτού κώδικα που παρέχει μια ρεαλιστική και προσαρμόσιμη πλατφόρμα για την ανάπτυξη και τη δοκιμή ρομποτικών συστημάτων. Προσφέρει ένα ολοκληρωμένο σύνολο χαρακτηριστικών, όπως προσομοίωση φυσικής, μοντελοποίηση αισθητήρων, οπτικοποίηση και φιλικές προς τον χρήστη διεπαφές, καθιστώντας το μια δημοφιλή επιλογή μεταξύ ερευνητών, προγραμματιστών και εκπαιδευτικών στον τομέα της ρομποτικής. Η μηχανή φυσικής του Gazebo επιτρέπει την ακριβή προσομοίωση της δυναμικής του ρομπότ, επιτρέποντας στους προγραμματιστές να αναπαράγουν αλληλεπιδράσεις και συμπεριφορές στον πραγματικό κόσμο. Υποστηρίζει τη μοντελοποίηση διαφόρων ρομποτικών πλατφορμών, συμπεριλαμβανομένων κινητών ρομπότ, χειριστών και εναέριων οχημάτων, επιτρέποντας την προσομοίωση ενός ευρέος φάσματος διαμορφώσεων και περιβαλλόντων ρομπότ. Με την ακριβή προσομοίωση των φυσικών ιδιοτήτων και των περιορισμών των ρομπότ, το Gazebo διευκολύνει την αξιολόγηση και τη βελτίωση των αλγορίθμων ελέγχου, των στρατηγικών σχεδιασμού κίνησης και των τεχνικών αντίληψης που βασίζονται σε αισθητήρες. Ένα από τα βασικά πλεονεκτήματα του Gazebo είναι η εκτεταμένη βιβλιοθήκη μοντέλων αισθητήρων του, η οποία επιτρέπει την προσομοίωση ενός

ευρέος φάσματος αισθητήρων που χρησιμοποιούνται συνήθως στη ρομποτική, όπως κάμερες, lidar και IMU. Αυτά τα μοντέλα αισθητήρων δημιουργούν ρεαλιστικά δεδομένα που μπορούν να χρησιμοποιηθούν για τη δοκιμή και την αξιολόγηση αλγορίθμων αντίληψης, τεχνικών σύντηξης αισθητήρων και συστημάτων χαρτογράφησης και εντοπισμού. Το Gazebo παρέχει επίσης εργαλεία για την οπτικοποίηση δεδομένων αισθητήρων, επιτρέποντας στους προγραμματιστές να αποκτήσουν πληροφορίες για την απόδοση και τη συμπεριφορά των ρομποτικών συστημάτων τους. Οι δυνατότητες οπτικοποίησης του Gazebo επιτρέπουν τη ρεαλιστική απόδοση εικονικών περιβαλλόντων, συμπεριλαμβανομένων των εφέ φωτισμού, των υφών και των αλληλεπιδράσεων αντικειμένων. Αυτό επιτρέπει τη δημιουργία οπτικά συναρπαστικών και καθηλωτικών προσομοιώσεων που μοιάζουν πολύ με σενάρια του πραγματικού κόσμου. Τα εργαλεία οπτικοποίησης που παρέχονται από το Gazebo διευκολύνουν τον εντοπισμό σφαλμάτων και την απεικόνιση των συμπεριφορών του ρομπότ, καθώς και την ανάλυση των αποτελεσμάτων της προσομοίωσης και την αξιολόγηση της απόδοσης. Εκτός από το πλούσιο σύνολο δυνατοτήτων του, το Gazebo υποστηρίζει την ενοποίηση με άλλα ευρέως χρησιμοποιούμενα πλαίσια και βιβλιοθήκες ρομποτικής, όπως το ROS (Robot Operating System). Αυτό επιτρέπει στους προγραμματιστές να συνδυάσουν τις δυνατότητες του Gazebo με το εκτεταμένο οικοσύστημα ROS, αξιοποιώντας τα οφέλη και των δύο πλατφορμών. Η απρόσκοπτη ενοποίηση μεταξύ Gazebo και ROS επιτρέπει την ανάπτυξη πολύπλοκων και διασυνδεδεμένων ρομποτικών συστημάτων, όπου το Gazebo χρησιμεύει ως περιβάλλον προσομοίωσης για δοκιμές και πρωτότυπα, ενώ το ROS παρέχει το ενδιάμεσο λογισμικό για επικοινωνία, έλεγχο και λειτουργικότητα υψηλότερου επιπέδου. Επιπλέον, το Gazebo προωθεί μια κοινότητα συνεργασίας και ανοιχτού κώδικα, ενθαρρύνοντας την ανταλλαγή γνώσεων, τη συμβολή κώδικα και τη συνεργασία μεταξύ ερευνητών και προγραμματιστών. Η κοινότητα Gazebo παρέχει πληθώρα πόρων, συμπεριλαμβανομένων οδηγών, δειγμάτων μοντέλων και προσθηκών, καθώς και φόρουμ και λίστες αλληλογραφίας για την αναζήτηση βοήθειας και την ανταλλαγή πληροφοριών.Επιπλέον, αυτή η διατριβή θα διερευνήσει τεχνικές για τη βελτιστοποίηση της απόδοσης και της χρηστικότητας των προσομοιώσεων Gazebo. Θα διερευνήσει θέματα όπως η αποτελεσματική προσομοίωση φυσικής, η βελτιστοποίηση μοντέλων αισθητήρων, οι βελτιστοποιήσεις απόδοσης και η επεκτασιμότητα για τη βελτίωση των δυνατοτήτων του Gazebo και τη βελτίωση της συνολικής εμπειρίας προσομοίωσης. Περνώντας τα όρια των δυνατοτήτων του Gazebo και αξιοποιώντας τα εκτεταμένα χαρακτηριστικά του, στοχεύουμε να συμβάλουμε στην πρόοδο της έρευνας και ανάπτυξης της ρομποτικής.

2.5 Gym Gazebo Kinetic

Το gym_gazebo_kinetic είναι ένα ισχυρό ενισχυτικό εκπαιδευτικό περιβάλλον που συνδυάζει την ευελιξία του OpenAI Gym με τις ρεαλιστικές δυνατότητες προσομοίωσης του Gazebo στο οικοσύστημα ROS (Robot Operating System). Παρέχει μια απρόσκοπτη ενοποίηση μεταξύ του πλαισίου μάθησης ενίσχυσης Gym και του περιβάλλοντος προσομοίωσης Gazebo, επιτρέποντας σε ερευνητές και προγραμματιστές να εκπαιδεύσουν και να αξιολογήσουν αλγόριθμους ενίσχυσης

μάθησης για ρομποτική σε ένα ρεαλιστικό και προσαρμόσιμο περιβάλλον. Το περιβάλλον gym_gazebo_kinetic προσφέρει ένα ευρύ φάσμα σεναρίων ρομποτικής προσομοίωσης, επιτρέποντας στους ερευνητές να δοκιμάσουν και να επικυρώσουν τους αλγόριθμους ενίσχυσης μάθησης σε διάφορες εργασίες και περιβάλλοντα. Αυτά τα σενάρια μπορεί να περιλαμβάνουν πλοήγηση ρομπότ, χειρισμό αντικειμένων, έλεγχο βραχίονα ρομπότ και πολλά άλλα. Παρέχοντας μια τυποποιημένη διεπαφή μέσω του Gym, το gym_gazebo_kinetic απλοποιεί τη διαδικασία ανάπτυξης και αξιολόγησης παραγόντων ενίσχυσης μάθησης, καθιστώντας το προσβάσιμο σε ερευνητές και επαγγελματίες στον τομέα της ρομποτικής. Ένα από τα βασικά πλεονεκτήματα του gym_gazebo_kinetic είναι η ικανότητά του να μοντελοποιεί πολύπλοκες δυναμικές και φυσικές αλληλεπιδράσεις εντός του περιβάλλοντος προσομοίωσης Gazebo. Αξιοποιεί τη μηχανή φυσικής του Gazebo για να προσομοιώνει με ακρίβεια τις συμπεριφορές και τους περιορισμούς των ρομποτικών συστημάτων, συμπεριλαμβανομένων της κινηματικής, της δυναμικής και των αποκρίσεων αισθητήρων τους. Αυτό επιτρέπει στους ερευνητές να δημιουργήσουν ρεαλιστικά περιβάλλοντα εκπαίδευσης που μοιάζουν πολύ με σενάρια του πραγματικού κόσμου, επιτρέποντας την ανάπτυξη παραγόντων ενίσχυσης μάθησης που γενικεύονται καλά στα φυσικά ρομπότ. Ένα άλλο αξιοσημείωτο χαρακτηριστικό του gym_gazebo_kinetic είναι η συμβατότητά του με το οικοσύστημα ROS. Το ROS παρέχει ένα πλούσιο σύνολο εργαλείων και βιβλιοθηκών για ρομποτική ανάπτυξη, συμπεριλαμβανομένων ισχυρών πλαισίων αντίληψης, σχεδιασμού και ελέγχου. Με την ενσωμάτωση με το ROS, το gym_gazebo_kinetic επιτρέπει στους ερευνητές να αξιοποιήσουν τις δυνατότητες του ROS για αντίληψη, έλεγχο και επικοινωνία, ενισχύοντας περαιτέρω τον ρεαλισμό και τη λειτουργικότητα του περιβάλλοντος προσομοίωσης. Αυτή η ενοποίηση επιτρέπει την απρόσκοπτη μεταφορά εκπαιδευμένων μοντέλων μάθησης ενίσχυσης από την προσομοίωση σε πραγματικά ρομπότ, διευκολύνοντας την ανάπτυξη μαθησιακών πολιτικών σε φυσικά συστήματα ρομπότ. Επιπλέον, το gym_gazebo_kinetic επωφελείται από την ενεργή και ζωντανή κοινότητα ROS. Η κοινότητα παρέχει εκτεταμένη υποστήριξη, πόρους και τεκμηρίωση, διευκολύνοντας τους ερευνητές και τους προγραμματιστές να ξεκινήσουν με το περιβάλλον και να αντιμετωπίσουν διάφορες προκλήσεις της ρομποτικής. Η διαθεσιμότητα προκατασκευασμένων μοντέλων ρομπότ, σεναρίων προσομοίωσης και δειγμάτων κωδίκων επιταχύνει περαιτέρω τη διαδικασία ανάπτυξης, επιτρέποντας στους ερευνητές να επικεντρωθούν στους συγκεκριμένους ερευνητικούς στόχους τους και στους αλγόριθμους ενίσχυσης μάθησης. Επιπλέον, αυτή η διατριβή θα διερευνήσει τεχνικές για τη βελτιστοποίηση της απόδοσης και της επεκτασιμότητας των προσομοιώσεων gym_gazebo_kinetic. Θα διερευνήσει θέματα όπως η επιτάχυνση της προσομοίωσης, η παραλληλοποίηση, η διαχείριση πόρων και η ενοποίηση με πλατφόρμες υπολογιστικού νέφους για τη βελτίωση της αποτελεσματικότητας και της επεκτασιμότητας της εκπαιδευτικής διαδικασίας ενισχυτικής μάθησης. Περνώντας τα όρια του gym_gazebo_kinetic και αξιοποιώντας τις δυνατότητές του, στοχεύουμε να συμβάλουμε στην πρόοδο της έρευνας για την ενίσχυση της μάθησης και στην εφαρμογή της στη ρομποτική.

3 Έννοιες και Μαθηματική Διατύπωση

3.1 Μαρκοβιανές Διαδικασίες Αποφάσεων - Markov Decision Process

Οι Μαρκοβιανές Διαδικασίες Αποφάσεων (MDPs) παρέχουν ένα μαθηματικό πλαίσιο για τη μοντελοποίηση διαδοχικών προβλημάτων λήψης αποφάσεων στον τομέα της ενισχυτικής μάθησης. Τα MDP χρησιμοποιούνται ευρέως για την επισημοποίηση της αλληλεπίδρασης μεταξύ ενός πράκτορα και του περιβάλλοντος του, επιτρέποντας στον πράκτορα να μάθει βέλτιστες πολιτικές που μεγιστοποιούν τις μακροπρόθεσμες σωρευτικές ανταμοιβές. Τα MDP χαρακτηρίζονται από ένα σύνολο καταστάσεων, ενεργειών, πιθανοτήτων μετάβασης και ανταμοιβών. Σε κάθε χρονικό βήμα, ο πράκτορας παρατηρεί την τρέχουσα κατάσταση του περιβάλλοντος, επιλέγει μια ενέργεια και μεταβαίνει σε μια νέα κατάσταση με βάση την υποκείμενη δυναμική. Οι πιθανότητες μετάβασης περιγράφουν την πιθανότητα μετάβασης από τη μια κατάσταση στην άλλη όταν λαμβάνεται μια συγκεκριμένη ενέργεια. Επιπλέον, ο πράκτορας λαμβάνει ένα σήμα ανταμοιβής που ποσοτικοποιεί την άμεση επιθυμία του ζεύγους κατάστασης-δράσης. Η βασική έννοια στα MDP είναι η έννοια της ιδιότητας Markov, η οποία δηλώνει ότι η μελλοντική κατάσταση και η ανταμοιβή εξαρτώνται μόνο από την τρέχουσα κατάσταση και δράση, ανεξάρτητα από την προηγούμενη ιστορία. Αυτή η ιδιότητα επιτρέπει στον πράκτορα να λαμβάνει αποφάσεις με βάση την τρέχουσα κατάσταση χωρίς να απαιτείται γνώση του πλήρους ιστορικού προηγούμενων καταστάσεων και ενεργειών. Χρησιμοποιώντας αυτήν την ιδιότητα, τα MDPs διευκολύνουν αποτελεσματικούς και κλιμακωτούς αλγόριθμους για την επίλυση πολύπλοκων προβλημάτων λήψης αποφάσεων. Ο στόχος στα MDP είναι να βρεθεί μια βέλτιστη πολιτική που να μεγιστοποιεί τις αναμενόμενες σωρευτικές ανταμοιβές με την πάροδο του χρόνου. Η πολιτική καθορίζει τη συμπεριφορά του πράκτορα, αντιστοιχίζοντας καταστάσεις σε ενέργειες. Στόχος είναι να εντοπιστεί η πολιτική που οδηγεί στην υψηλότερη δυνατή μακροπρόθεσμη ανταμοιβή. Διάφοροι αλγόριθμοι, όπως επανάληψη τιμών, επανάληψη πολιτικής και αλγόριθμοι ενίσχυσης εκμάθησης, μπορούν να χρησιμοποιηθούν για την επίλυση MDP και τον καθορισμό της βέλτιστης πολιτικής. Τα MDP έχουν εφαρμογές σε ένα ευρύ φάσμα τομέων, όπως η ρομποτική, η αναπαραγωγή παιχνιδιών, τα αυτόνομα συστήματα και η διαχείριση πόρων. Παρέχουν ένα πλαίσιο αρχών για τη μοντελοποίηση προβλημάτων λήψης αποφάσεων και επιτρέπουν την ανάπτυξη ευφυών παραγόντων ικανών να μαθαίνουν και να προσαρμόζονται σε πολύπλοκα περιβάλλοντα. Σε αυτή τη διατριβή, στοχεύουμε να διερευνήσουμε τα χαρακτηριστικά και τις εφαρμογές των MDP σε διαφορετικούς τομείς. Θα διερευνήσουμε διάφορους αλγόριθμους για την επίλυση MDP, όπως ο δυναμικός προγραμματισμός, οι μέθοδοι Monte Carlo και η χρονική μάθηση διαφορών. Επιπλέον, θα διερευνήσουμε την επίδραση διαφορετικών παραγόντων, όπως ο συντελεστής έκπτωσης, η αναπαράσταση του χώρου κατάστασης και η δομή ανταμοιβής, στην απόδοση και τη σύγκλιση των αλγορίθμων MDP. Τα ευρήματα αυτής της έρευνας θα συμβάλουν στη βαθύτερη κατανόηση των MDP και της δυνατότητας εφαρμογής τους σε προβλήματα λήψης αποφάσεων σε πραγματικό κόσμο. Αναλύοντας την απόδοση και τη συμπεριφορά των αλγορίθμων MDP, μπορούμε να

αποκτήσουμε γνώσεις για τα δυνατά τους σημεία, τους περιορισμούς και τις ανταλλαγές τους. Αυτή η γνώση θα βοηθήσει στην προώθηση του τομέα της ενισχυτικής μάθησης και θα καθοδηγήσει την ανάπτυξη αποτελεσματικών συστημάτων λήψης αποφάσεων σε πρακτικές εφαρμογές. Συνολικά, τα MDP παρέχουν ένα επίσημο πλαίσιο για τη μοντελοποίηση διαδοχικών προβλημάτων λήψης αποφάσεων, επιτρέποντας στους πράκτορες να μάθουν βέλτιστες πολιτικές με βάση την ιδιότητα Markov. Διερευνώντας τα MDP και τους σχετικούς αλγορίθμους τους, μπορούμε να αποκαλύψουμε θεμελιώδεις αρχές λήψης αποφάσεων και να αναπτύξουμε έξυπνα συστήματα που υπερέχουν σε πολύπλοκα, δυναμικά περιβάλλοντα.

3.2 Policy - Πολιτική

Μια πολιτική είναι ένα θεμελιώδες στοιχείο των αλγορίθμων ενισχυτικής μάθησης (RL), που διαδραματίζει κρίσιμο ρόλο στη διαδικασία λήψης αποφάσεων ενός πράκτορα RL. Καθορίζει τις ενέργειες που πρέπει να κάνει ο πράκτορας σε διαφορετικές καταστάσεις ενός περιβάλλοντος για να μεγιστοποιήσει τη μακροπρόθεσμη σωρευτική ανταμοιβή του. Η πολιτική περικλείει τη στρατηγική ή τη συμπεριφορά του πράκτορα, αντιστοιχίζοντας τις καταστάσεις σε ενέργειες που βασίζονται σε γνώσεις ή σε στρατηγικές εξερεύνησης. Ο σχεδιασμός και η επιλογή μιας κατάλληλης πολιτικής RL είναι μια κρίσιμη πτυχή της έρευνας και των εφαρμογών RL. Υπάρχουν διάφοροι τύποι πολιτικών, που κυμαίνονται από απλούς ντετερμινιστικούς κανόνες έως πολύπλοκα στοχαστικά μοντέλα. Η επιλογή της πολιτικής επηρεάζει σε μεγάλο βαθμό την ικανότητα του πράκτορα να εξερευνά το περιβάλλον, να εκμεταλλεύεται τη γνώση που έχει μάθει, να χειρίζεται την αβεβαιότητα και να γενικεύει τη συμπεριφορά του σε διαφορετικές καταστάσεις. Ένας κοινός τύπος πολιτικής είναι η ντετερμινιστική πολιτική, η οποία αντιστοιχίζει άμεσα τα κράτη σε συγκεκριμένες ενέργειες. Οι ντετερμινιστικές πολιτικές χρησιμοποιούνται συχνά σε εργασίες όπου η βέλτιστη δράση είναι σχετικά απλή και ντετερμινιστική, όπως ο έλεγχος απλών ρομποτικών συστημάτων ή παραγόντων που παίζουν παιχνίδι. Αυτές οι πολιτικές μπορούν να αναπαρασταθούν από προσεγγιστές συναρτήσεων, όπως τα νευρωνικά δίκτυα, που λαμβάνουν καταστάσεις ως είσοδο και εξάγουν τις αντίστοιχες ενέργειες. Οι στοχαστικές πολιτικές, από την άλλη πλευρά, εισάγουν ένα πιθανό στοιχείο στη διαδικασία λήψης αποφάσεων. Αντί να επιλέγουν ντετερμινιστικά μια ενέργεια, οι στοχαστικές πολιτικές εκχωρούν πιθανότητες σε διαφορετικές ενέργειες με βάση την παρατηρούμενη κατάσταση. Αυτή η πιθανοτική φύση επιτρέπει την εξερεύνηση και επιτρέπει στον πράκτορα να χειριστεί την αβεβαιότητα στο περιβάλλον. Οι στοχαστικές πολιτικές μπορούν να αναπαρασταθούν από κατανομές πιθανότητας, όπως συναρτήσεις softmax ή κατανομές Gaussian, όπου οι πιθανότητες διαφορετικών ενεργειών δειγματοληπτούνται ή υπολογίζονται με βάση τις παραμέτρους πολιτικής και τις παρατηρούμενες καταστάσεις. Η επιλογή μεταξύ ντετερμινιστικών και στοχαστικών πολιτικών εξαρτάται από τα χαρακτηριστικά του προβλήματος RL. Οι ντετερμινιστικές πολιτικές συχνά ευνοούνται σε εργασίες όπου η βέλτιστη δράση μπορεί να προσδιοριστεί με ακρίβεια και η δυναμική του περιβάλλοντος είναι καλά κατανοητή. Οι στοχαστικές πολιτικές, από την άλλη πλευρά, είναι πιο κατάλληλες για εργασίες με αβέβαιες ή μερικώς παρατηρήσιμες καταστάσεις, που απαιτούν

εξερεύνηση και προσαρμοστικότητα. Επιπλέον, οι πολιτικές RL μπορούν να κατηγοριοποιηθούν σε πολιτικές εντός και εκτός πολιτικής. Οι πολιτικές εντός πολιτικής ενημερώνουν τις παραμέτρους τους με βάση τα δεδομένα που συλλέγονται κατά την τρέχουσα αλληλεπίδραση με το περιβάλλον. Συχνά υποφέρουν από αργή σύγκλιση και περιορισμένη απόδοση δειγμάτων, αλλά μπορούν να χειριστούν αποτελεσματικά σενάρια διαδικτυακής μάθησης. Οι πολιτικές εκτός πολιτικής, από την άλλη πλευρά, μαθαίνουν από δεδομένα που συλλέγονται από διαφορετικές πολιτικές, επιτρέποντας την αποτελεσματικότερη χρήση της προηγούμενης εμπειρίας και την καλύτερη αποτελεσματικότητα του δείγματος. Χρησιμοποιούνται συνήθως σε εργασίες όπου χρησιμοποιείται μια ξεχωριστή πολιτική εξερεύνησης για τη συλλογή δεδομένων κατά την ενημέρωση της πολιτικής στόχου. Αυτή η έρευνα στοχεύει στον εντοπισμό της καταλληλότερης προσέγγισης αναπαράστασης και μάθησης πολιτικής για το δεδομένο πρόβλημα, προωθώντας την κατανόηση και την εφαρμογή των πολιτικών RL σε πρακτικά σενάρια. Επιπλέον, αυτή η διατριβή θα διερευνήσει τεχνικές για τη βελτιστοποίηση και τη βελτίωση της πολιτικής, όπως οι κλίσεις πολιτικής, οι μέθοδοι περιοχής εμπιστοσύνης ή οι προσεγγίσεις που βασίζονται στην αξία. Ο στόχος είναι η βελτίωση της απόδοσης, της σταθερότητας και των δυνατοτήτων γενίκευσης των πολιτικών RL, δίνοντάς τους τη δυνατότητα να χειρίζονται πολύπλοκα και υψηλών διαστάσεων περιβάλλοντα. Αυτή η έρευνα θα συμβάλει στην πρόοδο των τεχνικών σχεδιασμού και βελτιστοποίησης πολιτικής RL, οδηγώντας τελικά σε πιο αποτελεσματικούς και αποτελεσματικούς πράκτορες RL σε διάφορους τομείς.
Μια πολιτική μπορεί να είναι ντετερμινιστική οπότε και συμβολίζεται με μ:

$$a_t = \mu(s_t)$$

Αλλιώς μπορεί να είναι στοχαστική οπότε συμβολίζεται με π :

$$a_t \sim \pi(\cdot|s_t)$$

3.3 Εκτιμώμενη Επιστροφή - Expected Return

Στον τομέα της ενισχυτικής μάθησης, η έννοια της αναμενόμενης απόδοσης παίζει θεμελιώδη ρόλο στην αξιολόγηση της σκοπιμότητας των ενεργειών και των πολιτικών σε διαδοχικά προβλήματα λήψης αποφάσεων. Η αναμενόμενη απόδοση, γνωστή και ως αναμενόμενη σωρευτική ανταμοιβή ή αναμενόμενη συνολική ανταμοιβή, ποσοτικοποιεί τη μακροπρόθεσμη χρησιμότητα ή αξία μιας συγκεκριμένης απόφασης ή πολιτικής. Η αναμενόμενη απόδοση ορίζεται ως το άθροισμα των ανταμοιβών που αναμένει να συγκεντρώσει ένας πράκτορας με την πάροδο του χρόνου όταν ακολουθεί μια συγκεκριμένη πολιτική ή κάνει μια σειρά ενεργειών. Αποτυπώνει την έννοια της σωρευτικής ανταμοιβής, λαμβάνοντας υπόψη τόσο τις άμεσες ανταμοιβές όσο και τις μελλοντικές ανταμοιβές που μπορεί να είναι καθυστερημένες ή αβέβαιες. Λαμβάνοντας υπόψη την αναμενόμενη απόδοση, οι πράκτορες μπορούν να αξιολογήσουν τα πιθανά αποτελέσματα και τις ανταλλαγές που σχετίζονται με διαφορετικές ενέργειες και πολιτικές. Ο υπολογισμός της αναμενόμενης απόδοσης βασίζεται στην πιθανολογική φύση του περιβάλλοντος. Σε μια Διαδικασία Απόφασης Markov (MDP), για παράδειγμα, η αναμενόμενη απόδοση σε μια δεδομένη κατάσταση εξαρτάται από την τρέχουσα κατάσταση, την επιλεγμένη ενέργεια και τις μελλοντικές καταστάσεις και τις ανταμοιβές που μπορεί να συναντηθούν. Αυτός ο υπολογισμός συνήθως περιλαμβάνει την εκτίμηση της αναμενόμενης αξίας των μελλοντικών ανταμοιβών

μέσω μεθόδων όπως ο δυναμικός προγραμματισμός, η προσομοίωση Monte Carlo ή η χρονική εκμάθηση διαφορών. Η έννοια της αναμενόμενης απόδοσης χρησιμεύει ως μέτρο απόδοσης για την αξιολόγηση της αποτελεσματικότητας διαφορετικών πολιτικών και στρατηγικών λήψης αποφάσεων. Οι πράκτορες στοχεύουν στη μεγιστοποίηση της αναμενόμενης απόδοσης επιλέγοντας ενέργειες ή πολιτικές που οδηγούν στις υψηλότερες δυνατές σωρευτικές ανταμοιβές. Αυτός ο στόχος επιτυγχάνεται συχνά μέσω επαναληπτικών αλγορίθμων βελτιστοποίησης, όπως η επανάληψη τιμών, η επανάληψη πολιτικής ή οι μέθοδοι ενίσχυσης μάθησης. Η αναμενόμενη απόδοση έχει σημαντικές επιπτώσεις σε διάφορους τομείς και εφαρμογές. Επιτρέπει στους πράκτορες να λαμβάνουν τεκμηριωμένες αποφάσεις σε δυναμικά περιβάλλοντα, που κυμαίνονται από τη ρομποτική και τα αυτόνομα συστήματα μέχρι τη χρηματοδότηση και το παιχνίδι. Βελτιστοποιώντας την αναμενόμενη απόδοση, οι πράκτορες μπορούν να προσαρμόσουν τη συμπεριφορά τους στις μεταβαλλόμενες συνθήκες, να μάθουν από την εμπειρία και να ανακαλύψουν αποτελεσματικές πολιτικές που εξισορροπούν τις βραχυπρόθεσμες ανταμοιβές και τους μακροπρόθεσμους στόχους. Σε αυτή τη διατριβή, διερευνούμε την έννοια της αναμενόμενης απόδοσης στο πλαίσιο της ενισχυτικής μάθησης. Εξερευνούμε τα θεωρητικά θεμέλια και τις μαθηματικές διατυπώσεις της αναμενόμενης απόδοσης, λαμβάνοντας υπόψη τον ρόλο της στη λήψη αποφάσεων και στη βελτιστοποίηση της πολιτικής. Επιπλέον, εξετάζουμε διαφορετικούς αλγόριθμους και τεχνικές για την εκτίμηση και τη μεγιστοποίηση της αναμενόμενης απόδοσης, αξιολογώντας την υπολογιστική τους απόδοση, τις ιδιότητες σύγκλισης και την εφαρμογή τους σε προβλήματα του πραγματικού κόσμου. Μελετώντας την αναμενόμενη απόδοση, στοχεύουμε να βελτιώσουμε την κατανόησή μας για τη βέλτιστη λήψη αποφάσεων και το σχεδιασμό έξυπνων συστημάτων. Αναλύουμε τον αντίκτυπο διαφόρων παραγόντων, όπως παράγοντες έκπτωσης, δομές ανταμοιβής και συμβιβασμούς εξερεύνησης-εκμετάλλευσης, στην αναμενόμενη απόδοση και στη συμπεριφορά σύγκλισής της. Μέσω εμπειρικών πειραμάτων και αναλύσεων, επιδιώκουμε να εντοπίσουμε στρατηγικές που εξισορροπούν αποτελεσματικά τις άμεσες ανταμοιβές και τους μακροπρόθεσμους στόχους, οδηγώντας σε βελτιωμένες δυνατότητες λήψης αποφάσεων. Τα αποτελέσματα αυτής της έρευνας θα συμβάλουν στην πρόοδο της ενισχυτικής μάθησης και θα παρέχουν πολύτιμες γνώσεις για τη βελτιστοποίηση της αναμενόμενης απόδοσης. Εξετάζοντας τις θεωρητικές πτυχές και τις πρακτικές επιπτώσεις της αναμενόμενης απόδοσης, μπορούμε να αναπτύξουμε πιο αποδοτικούς και αποτελεσματικούς αλγόριθμους για τη λήψη αποφάσεων και τη βελτιστοποίηση πολιτικής σε δυναμικά και αβέβαια περιβάλλοντα. Συμπερασματικά, η αναμενόμενη απόδοση χρησιμεύει ως κρίσιμη μέτρηση για την αξιολόγηση της σκοπιμότητας των ενεργειών και των πολιτικών στην ενισχυτική μάθηση. Με την ποσοτικοποίηση των σωρευτικών ανταμοιβών που μπορεί να αναμένει να επιτύχει ένας πράκτορας, καθοδηγεί τη λήψη αποφάσεων και τη βελτιστοποίηση πολιτικής. Μέσω αυτής της διατριβής, στοχεύουμε να εμβαθύνουμε την κατανόησή μας για την αναμενόμενη απόδοση, να διερευνήσουμε τις υπολογιστικές πτυχές της και να ενισχύσουμε την εφαρμογή της σε πρακτικά ενισχυτικά μαθησιακά προβλήματα.

Η συνάρτηση ανταμοιβής R είναι εξαιρετικά σημαντική στην ενισχυτική μάθηση. Εξαρτάται από την τρέχουσα κατάσταση του κόσμου, τη δράση που μόλις έγινε και την επόμενη κατάσταση του κόσμου:

$$r_t = R(s_t, a_t, s_{t+1})$$

Ένα είδος επιστροφής είναι η μη προεξοφλημένη επιστροφή πεπερασμένου ορίζοντα, η οποία είναι απλώς το άθροισμα των ανταμοιβών που λαμβάνονται σε ένα σταθερό παράθυρο βημάτων:

$$R(\tau) = \sum_{t=0}^{T} r_t$$

Ένα άλλο είδος επιστροφής είναι η προεξοφλημένη επιστροφή άπειρου ορίζοντα, η οποία είναι το άθροισμα όλων των ανταμοιβών που έλαβε ποτέ ο πράκτορας, αλλά με έκπτωση από το πόσο μακριά θα αποκτηθούν στο μέλλον. Αυτή η διατύπωση ανταμοιβής περιλαμβάνει έναν συντελεστή έκπτωσης $\gamma \in (0,1)$:

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t$$

3.4 Value Functions

Είναι συχνά χρήσιμο να γνωρίζουμε την αξία μιας κατάστασης ή ενός ζεύγους κατάστασης-ενέργειας. Με τον όρο αξία, εννοούμε την αναμενόμενη επιστροφή εάν ξεκινήσουμε σε αυτήν την κατάσταση ή το ζεύγος κατάστασης-ενέργειας και στη συνέχεια ενεργήσουμε σύμφωνα με μια συγκεκριμένη πολιτική για πάντα. Οι value functions (συναρτήσεις αξίας) χρησιμοποιούνται, με τον ένα ή τον άλλο τρόπο, σχεδόν σε κάθε αλγόριθμο RL. Οι 4 κυριότερες value functions είναι :

1.Η **On-Policy Value Function** (συνάρτηση αξίας εντός πολιτικής) που δίνει την εκτιμώμενη επιστροφή όταν αρχίζεις από μια κατάσταση s και δρας πάντα με βάση μιας πολιτικής p:

$$V^{\pi}(s) = E_{\tau \sim \pi}[R(\tau)|s_0 = s]$$

Στη θεωρία πιθανοτήτων, το E ονομάζεται εκτιμώμενη αξία (expected value) ή αλλιώς προσδοκία (expectation) και δείχνει τη μέση τιμή για πολλά πειράματα μιας τυχαίας μεταβλητής.

2.Η **On-Policy Action-Value Function** (συνάρτηση δράσης-αξίας εντός πολιτικής) που δίνει την αναμενόμενη επιστροφή εάν ξεκινήσετε σε κατάσταση s , προβείτε σε αυθαίρετη δράση a (η οποία μπορεί να μην προήλθε από την πολιτική) και, στη συνέχεια, για πάντα μετά ενεργήστε σύμφωνα με την πολιτική p:

$$Q^{\pi}(s, a) = E_{\tau \sim \pi}[R(\tau)|s_0 = s, a_0 = a]$$

3.Η **Optimal Value Function** (βέλτιστη συνάρτηση αξίας) που δίνει την εκτιμώμενη επιστροφή όταν αρχίζεις από μια κατάσταση s και δρας πάντα με βάση της βέλτιστης πολιτικής του περιβάλλοντος :

$$V^*(s) = max_\pi E_{\tau\sim\pi}[R(\tau)|s_0 = s]$$

4.Η **Optimal Action-Value Function** (βέλτιστη συνάρτηση δράσης-αξίας) που δίνει την αναμενόμενη επιστροφή εάν ξεκινήσετε σε κατάσταση s , προβείτε σε αυθαίρετη δράση a (η οποία μπορεί να μην προήλθε από την πολιτική) και, στη συνέχεια, για πάντα μετά ενεργήστε σύμφωνα με την βέλτιστη πολιτική :

$$Q^*(s,a) = max_\pi E_{\tau\sim\pi}[R(\tau)|s_0 = s, a_0 = a]$$

3.5 Εξίσωση Bellman

Στον τομέα της ενισχυτικής μάθησης, η εξίσωση Bellman είναι μια θεμελιώδης έννοια που αποτελεί τη βάση της διαδικασίας μάθησης και λήψης αποφάσεων σε διαδοχικά περιβάλλοντα. Παρέχει μια αναδρομική σχέση που χαρακτηρίζει τη βέλτιστη συνάρτηση τιμής και την πολιτική στις Διαδικασίες Αποφάσεων Markov (MDPs). Η εξίσωση Bellman πήρε το όνομά της από τον Richard Bellman, ο οποίος συνέβαλε σημαντικά στον τομέα του δυναμικού προγραμματισμού και της εφαρμογής του στον έλεγχο προβλημάτων. Η εξίσωση εκφράζει την τιμή ενός ζεύγους κατάστασης ή κατάστασης-δράσης ως προς την άμεση ανταμοιβή και την αναμενόμενη τιμή των επόμενων καταστάσεων που μπορεί να επιτευχθεί με τη λήψη συγκεκριμένων ενεργειών. Η εξίσωση Bellman συλλαμβάνει την αρχή της βέλτιστης, δηλώνοντας ότι η αξία μιας κατάστασης είναι ίση με την άμεση ανταμοιβή που λαμβάνεται σε αυτήν την κατάσταση συν την προεξοφλημένη αξία των μελλοντικών καταστάσεων. Ο παράγοντας έκπτωσης αντιπροσωπεύει τη σημασία που αποδίδεται στις άμεσες ανταμοιβές σε σύγκριση με τις μελλοντικές ανταμοιβές και καθορίζει την αντιστάθμιση μεταξύ των άμεσων κερδών και των μακροπρόθεσμων στόχων. Η εξίσωση Bellman παρέχει τη βάση για αλγόριθμους επανάληψης τιμών και επανάληψης πολιτικών, οι οποίοι χρησιμοποιούνται ευρέως για τον υπολογισμό των βέλτιστων συναρτήσεων και πολιτικών τιμών σε MDP. Αυτοί οι αλγόριθμοι αξιοποιούν την αναδρομική φύση της εξίσωσης Bellman για να ενημερώνουν επαναληπτικά τις εκτιμήσεις τιμών μέχρι τη σύγκλιση, οδηγώντας τελικά στον προσδιορισμό των βέλτιστων πολιτικών. Επιλύοντας την εξίσωση Bellman, οι πράκτορες ενίσχυσης μάθησης μπορούν να καθορίσουν τη βέλτιστη ενέργεια που πρέπει να λάβουν σε κάθε κατάσταση για να μεγιστοποιήσετε την αναμενόμενη αθροιστική ανταμοιβή. Η εξίσωση επιτρέπει στους πράκτορες να αξιολογήσουν τα πιθανά αποτελέσματα διαφορετικών ενεργειών και να λαμβάνουν τεκμηριωμένες αποφάσεις με βάση τις εκτιμώμενες τιμές των καταστάσεων ή των ζευγών κατάστασης-δράσεων. Η εξίσωση Bellman έπαιξε καθοριστικό ρόλο στην ανάπτυξη διαφόρων αλγορίθμων και τεχνικών ενισχυτικής μάθησης, συμπεριλαμβανομένων των μεθόδων Q-learning, SARSA και μεθόδων κλίσης πολιτικής. Αυτοί οι αλγόριθμοι αξιοποιούν τις πληροφορίες που παρέχονται από την εξίσωση Bellman για να μάθουν συναρτήσεις αξίας και πολιτικές που βελτιστοποιούν τις μακροπρόθεσμες ανταμοιβές. Σε αυτή τη διατριβή, διερευνούμε την εξίσωση Bellman και τις εφαρμογές της στην ενισχυτική μάθηση. Εμβαθύνουμε στα θεωρητικά θεμέλια και τις μαθηματικές διατυπώσεις της εξίσωσης Bellman, λαμβάνοντας υπόψη τον ρόλο της στην εκτίμηση της αξίας, τη βελτιστοποίηση πολιτικής και τη λήψη

αποφάσεων. Επιπλέον, διερευνούμε διαφορετικούς αλγόριθμους και προσεγγίσεις για την επίλυση της εξίσωσης Bellman, αναλύοντας την υπολογιστική τους απόδοση, τις ιδιότητες σύγκλισης και την εφαρμογή τους σε προβλήματα του πραγματικού κόσμου. Μελετώντας την εξίσωση Bellman, στοχεύουμε να βελτιώσουμε την κατανόησή μας για τη βέλτιστη λήψη αποφάσεων και το σχεδιασμό ευφυών συστημάτων. Εξετάζουμε τον αντίκτυπο διάφορων παραγόντων, όπως παράγοντες έκπτωσης, δομές ανταμοιβής και συμβιβασμούς εξερεύνησης-εκμετάλλευσης, στη σύγκλιση και τη σταθερότητα των λύσεων των εξισώσεων Bellman. Μέσω εμπειρικών πειραμάτων και αναλύσεων, επιδιώκουμε να εντοπίσουμε στρατηγικές που βελτιώνουν την αποδοτικότητα και την αποτελεσματικότητα της επίλυσης της εξίσωσης Bellman, οδηγώντας σε βελτιωμένες δυνατότητες λήψης αποφάσεων. Τα αποτελέσματα αυτής της έρευνας θα συμβάλουν στην πρόοδο της ενισχυτικής μάθησης και θα παρέχουν πολύτιμες γνώσεις για τη βελτιστοποίηση των συναρτήσεων και των πολιτικών αξίας. Διερευνώντας τις θεωρητικές πτυχές και τις πρακτικές επιπτώσεις της εξίσωσης Bellman, μπορούμε να αναπτύξουμε πιο αποδοτικούς και αποτελεσματικούς αλγόριθμους για τη λήψη αποφάσεων και τη βελτιστοποίηση πολιτικής σε δυναμικά και αβέβαια περιβάλλοντα. Συμπερασματικά, η εξίσωση Bellman χρησιμεύει ως βασική έννοια στην ενισχυτική μάθηση, παρέχοντας μια αναδρομική σχέση που χαρακτηρίζει τη βέλτιστη συνάρτηση τιμής και την πολιτική. Επιλύοντας την εξίσωση Bellman, οι πράκτορες μπορούν να καθορίσουν τη βέλτιστη ενέργεια που πρέπει να λάβουν σε κάθε κατάσταση και να επιτύχουν μακροπρόθεσμους στόχους. Μέσω αυτής της διατριβής, στοχεύουμε να εμβαθύνουμε την κατανόησή μας για την εξίσωση Bellman, να διερευνήσουμε τις υπολογιστικές πτυχές της και να ενισχύσουμε την εφαρμογή της σε πρακτικά ενισχυτικά μαθησιακά προβλήματα.
Οι εξισώσεις Bellman για τις on policy value functions είναι :

$$V^{\pi}(s) = E_{\alpha\sim\pi, s'\sim P}[r(s,a) + \gamma V^{\pi}(s')]$$
$$Q^{\pi}(s,a) = E_{s'\sim P}\left[r(s,a) + \gamma E_{\alpha'\sim\pi}[Q^{\pi}(s',a')]\right]$$

Οι εξισώσεις Bellman για τις optimal value functions είναι :

$$V^{*}(s) = max_{a} E_{s'\sim P}[r(s,a) + \gamma V^{*}(s')]$$
$$Q^{*}(s,a) = E_{s'\sim P}[r(s,a) + \gamma max_{a} Q^{*}(s',a')]$$

3.6 Εξερεύνηση - Εκμετάλλευση

Στον τομέα της ενισχυτικής μάθησης, η αντιστάθμιση εξερεύνησης και εκμετάλλευσης είναι μια κρίσιμη έννοια που αντιμετωπίζει την πρόκληση της εξισορρόπησης μεταξύ της απόκτησης νέας γνώσης (εξερεύνηση) και της εκμετάλλευσης της υπάρχουσας γνώσης (εκμετάλλευση) για τη μεγιστοποίηση των σωρευτικών ανταμοιβών σε διαδοχικές εργασίες λήψης αποφάσεων. Οι πράκτορες ενίσχυσης μάθησης στοχεύουν να μάθουν τις βέλτιστες πολιτικές αλληλεπιδρώντας με το περιβάλλον τους, λαμβάνοντας ανατροφοδότηση με τη μορφή ανταμοιβών ή κυρώσεων. Η αντιστάθμιση

εξερεύνησης και εκμετάλλευσης προκύπτει από την εγγενή αβεβαιότητα και τη μερική παρατηρησιμότητα του περιβάλλοντος. Οι πράκτορες πρέπει να αποφασίσουν εάν θα προβούν σε ενέργειες που είναι ήδη γνωστό ότι αποφέρουν ευνοϊκά αποτελέσματα (εκμετάλλευση) ή θα εξερευνήσουν νέες ενέργειες με την ελπίδα να ανακαλύψουν ακόμη καλύτερες επιλογές (εξερεύνηση). Η εξερεύνηση επιτρέπει στους πράκτορες να συλλέξουν περισσότερες πληροφορίες για το περιβάλλον, αποκαλύπτοντας δυνητικά πιο ικανοποιητικές ενέργειες που ήταν προηγουμένως άγνωστες. Από την άλλη πλευρά, η εκμετάλλευση εστιάζει στην εκμετάλλευση της γνωστής γνώσης για τη μεγιστοποίηση των άμεσων ανταμοιβών. Η επίτευξη ισορροπίας μεταξύ εξερεύνησης και εκμετάλλευσης είναι ζωτικής σημασίας για να αποφύγετε να κολλήσετε σε μη βέλτιστες πολιτικές ή να χάσετε ανεξερεύνητες ενέργειες υψηλής ανταμοιβής. Μια δημοφιλής προσέγγιση για την εξερεύνηση έναντι της εκμετάλλευσης είναι η στρατηγική epsilon-greedy, η οποία περιλαμβάνει την επιλογή της ενέργειας με την υψηλότερη εκτιμώμενη αξία τις περισσότερες φορές (εκμετάλλευση), αλλά περιστασιακά τη λήψη τυχαίων ενεργειών με μια ορισμένη πιθανότητα (εξερεύνηση). Αυτή η προσέγγιση επιτρέπει στους πράκτορες να εξερευνούν διαφορετικές ενέργειες, ενώ παράλληλα προτιμούν ενέργειες με υψηλότερες εκτιμώμενες τιμές. Μια άλλη ευρέως χρησιμοποιούμενη τεχνική είναι το Upper Confidence Bound (UCB), το οποίο χρησιμοποιεί ένα διάστημα εμπιστοσύνης για να εξισορροπήσει την εξερεύνηση και την εκμετάλλευση. Η UCB αποδίδει υψηλότερη προτεραιότητα σε ενέργειες με υψηλότερη αβεβαιότητα, καθώς μπορεί να έχουν υψηλότερες πιθανές ανταμοιβές. Μειώνοντας σταδιακά την αβεβαιότητα μέσω της εξερεύνησης, ο πράκτορας μπορεί να βελτιώσει τις εκτιμήσεις του και να λάβει πιο τεκμηριωμένες αποφάσεις. Οι αλγόριθμοι ληστών πολλαπλών όπλων, όπως ο epsilon-greedy και ο UCB, παρέχουν πρακτικές λύσεις στην αντιστάθμιση εξερεύνησης και εκμετάλλευσης. Αυτοί οι αλγόριθμοι επιτρέπουν στους πράκτορες να εξερευνούν και να εκμεταλλεύονται προσαρμοστικά με βάση τις γνώσεις τους και τις εκτιμήσεις αβεβαιότητας. Επιπλέον, πιο προηγμένες τεχνικές, όπως η δειγματοληψία Thompson και η αναζήτηση δέντρων στο Monte Carlo[10], προσφέρουν πιθανολογικές προσεγγίσεις στην εξερεύνηση που προσαρμόζουν δυναμικά τους ρυθμούς εξερεύνησης με βάση τα παρατηρούμενα αποτελέσματα. Η αντιστάθμιση εξερεύνησης και εκμετάλλευσης είναι ένα κρίσιμο ζήτημα στην ενισχυτική μάθηση, καθώς επηρεάζει άμεσα την ικανότητα του πράκτορα να μαθαίνει βέλτιστες πολιτικές και να λαμβάνει τεκμηριωμένες αποφάσεις. Η επιλογή της στρατηγικής εξερεύνησης μπορεί να επηρεάσει σημαντικά την ταχύτητα μάθησης, τη σύγκλιση σε βέλτιστες πολιτικές και τη συνολική απόδοση του πράκτορα. Σε αυτή τη διατριβή, εμβαθύνουμε στην αντιστάθμιση εξερεύνησης και εκμετάλλευσης στην ενισχυτική μάθηση. Διερευνούμε διαφορετικές στρατηγικές εξερεύνησης, αναλύουμε τα δυνατά και τους περιορισμούς τους και διερευνούμε τον αντίκτυπό τους στη μαθησιακή διαδικασία και την απόδοση. Επιπλέον, διερευνούμε τεχνικές για προσαρμοστική εξερεύνηση που προσαρμόζουν δυναμικά τους ρυθμούς εξερεύνησης με βάση τη μαθησιακή πρόοδο του πράκτορα και τα περιβαλλοντικά χαρακτηριστικά. Μέσα από εμπειρικά πειράματα και αναλύσεις, στοχεύουμε να εντοπίσουμε αποτελεσματικές στρατηγικές για την επίτευξη ισορροπίας μεταξύ εξερεύνησης και εκμετάλλευσης σε διάφορα σενάρια ενισχυτικής μάθησης. Αξιολογούμε την απόδοση διαφορετικών τεχνικών εξερεύνησης και συγκρίνουμε την αποτελεσματικότητά τους όσον αφορά την αποδοτικότητα μάθησης, την ταχύτητα σύγκλισης και τις συνολικές σωρευτικές ανταμοιβές. Τα αποτελέσματα αυτής της έρευνας θα συμβάλουν στην πρόοδο των στρατηγικών εξερεύνησης στην ενισχυτική μάθηση και θα παρέχουν

πολύτιμες γνώσεις για τη βελτιστοποίηση της αντιστάθμισης εξερεύνησης και εκμετάλλευσης. Μελετώντας τις θεωρητικές πτυχές και τις εμπειρικές αξιολογήσεις των τεχνικών εξερεύνησης, στοχεύουμε να ενισχύσουμε την αποδοτικότητα και την αποτελεσματικότητα των αλγορίθμων ενισχυτικής μάθησης σε διάφορους τομείς και εφαρμογές πραγματικού κόσμου. Συμπερασματικά, η αντιστάθμιση εξερεύνησης και εκμετάλλευσης είναι μια κρίσιμη εξέταση στην ενισχυτική μάθηση, εξισορροπώντας την απόκτηση νέας γνώσης με την εκμετάλλευση της υπάρχουσας γνώσης για τη μεγιστοποίηση των σωρευτικών ανταμοιβών. Επιλέγοντας προσεκτικά και προσαρμόζοντας στρατηγικές εξερεύνησης, οι παράγοντες ενίσχυσης μάθησης μπορούν να εξερευνήσουν αποτελεσματικά το περιβάλλον, να ανακαλύψουν βέλτιστες πολιτικές και να λάβουν τεκμηριωμένες αποφάσεις. Μέσω αυτής της διατριβής, στοχεύουμε να εμβαθύνουμε την κατανόησή μας για τις τεχνικές εξερεύνησης, να διερευνήσουμε τα χαρακτηριστικά απόδοσης τους και να συμβάλουμε στην ανάπτυξη πιο αποτελεσματικών και αποτελεσματικών στρατηγικών εξερεύνησης στην ενισχυτική μάθηση.

3.7 Αλγόριθμοι Ενισχυτικής Εκμάθησης

Οι αλγόριθμοι Reinforcement Learning (RL) διαδραματίζουν κεντρικό ρόλο δίνοντας τη δυνατότητα στους αυτόνομους πράκτορες να μαθαίνουν και να λαμβάνουν έξυπνες αποφάσεις σε πολύπλοκα περιβάλλοντα. Οι αλγόριθμοι RL έχουν σχεδιαστεί για να αντιμετωπίζουν διαδοχικά προβλήματα λήψης αποφάσεων, όπου ένας πράκτορας αλληλεπιδρά με ένα περιβάλλον, αναλαμβάνει ενέργειες και λαμβάνει ανατροφοδότηση με τη μορφή ανταμοιβών. Ο στόχος του RL είναι να μάθει μια βέλτιστη πολιτική που μεγιστοποιεί τη σωρευτική ανταμοιβή με την πάροδο του χρόνου. Ένα ευρύ φάσμα αλγορίθμων RL έχει αναπτυχθεί για την αντιμετώπιση διαφόρων προκλήσεων και σεναρίων. Αυτοί οι αλγόριθμοι μπορούν να κατηγοριοποιηθούν ευρέως σε μεθόδους που βασίζονται στην αξία, σε μεθόδους που βασίζονται σε πολιτικές και σε μεθόδους κριτικής. Οι αλγόριθμοι RL που βασίζονται σε τιμές στοχεύουν στην εκτίμηση της συνάρτησης τιμής, η οποία αντιπροσωπεύει την αναμενόμενη αθροιστική ανταμοιβή του να είσαι σε μια συγκεκριμένη κατάσταση ή να κάνεις μια συγκεκριμένη ενέργεια. Αυτοί οι αλγόριθμοι, όπως το Q-Learning[10] , το SARSA[11] και το Deep Q-Networks[12] (DQN), μαθαίνουν τη βέλτιστη συνάρτηση τιμής ενέργειας ενημερώνοντας επαναληπτικά τις εκτιμήσεις με βάση τις παρατηρούμενες ανταμοιβές και τις μεταβάσεις καταστάσεων. Οι μέθοδοι που βασίζονται σε τιμές είναι ιδιαίτερα αποτελεσματικές σε εργασίες με διακριτούς χώρους δράσης ή όταν ο χώρος κατάστασης είναι υψηλών διαστάσεων. Οι αλγόριθμοι RL που βασίζονται σε πολιτικές, από την άλλη πλευρά, βελτιστοποιούν άμεσα τη συνάρτηση πολιτικής, η οποία αντιστοιχίζει καταστάσεις σε ενέργειες. Αυτοί οι αλγόριθμοι, συμπεριλαμβανομένου του Policy Gradient και του Proximal Policy Optimization[13] (PPO), μαθαίνουν την πολιτική προσαρμόζοντας επαναληπτικά τις παραμέτρους της για να μεγιστοποιήσουν την αναμενόμενη αθροιστική ανταμοιβή. Οι μέθοδοι που βασίζονται σε πολιτικές είναι κατάλληλες για εργασίες με χώρους συνεχούς δράσης ή όταν η βέλτιστη πολιτική είναι περίπλοκη και απαιτεί στοχαστική αναπαράσταση. Οι μέθοδοι που ασκούν κριτική

συνδυάζουν στοιχεία τόσο των προσεγγίσεων που βασίζονται στην αξία όσο και των προσεγγίσεων που βασίζονται σε πολιτικές. Διατηρούν δύο ξεχωριστά μοντέλα: έναν ηθοποιό που μαθαίνει την πολιτική και έναν κριτικό που εκτιμά τη συνάρτηση αξίας. Ο ηθοποιός δημιουργεί ενέργειες με βάση την πολιτική και ο κριτικός παρέχει ανατροφοδότηση για την ποιότητα αυτών των ενεργειών. Στη συνέχεια, αυτή η ανατροφοδότηση χρησιμοποιείται για την επαναληπτική ενημέρωση της συνάρτησης πολιτικής και τιμής. Οι αλγόριθμοι που ασκούν κριτική, όπως το Advantage Actor-Critic[14] (A2C) και το Trust Region Policy Optimization[15] (TRPO), επιτυγχάνουν μια ισορροπία μεταξύ εξερεύνησης και εκμετάλλευσης, καθιστώντας τους αποτελεσματικούς σε ένα ευρύ φάσμα εργασιών RL. Επιπλέον, οι αλγόριθμοι RL μπορούν να βελτιωθούν περαιτέρω μέσω της ενσωμάτωσης πρόσθετων τεχνικών, όπως στρατηγικές εξερεύνησης, επανάληψη εμπειρίας, διαμόρφωση ανταμοιβής και μεταφορά μάθησης. Οι στρατηγικές εξερεύνησης, όπως το epsilon-greedy[16] και το Thomson sampling[17],προωθούν την ανακάλυψη νέων και δυνητικά καλύτερων ενεργειών, επιτρέποντας στον πράκτορα να εξερευνήσει αποτελεσματικά το περιβάλλον. Η επανάληψη της εμπειρίας επιτρέπει στον πράκτορα RL να αποθηκεύει και να επαναχρησιμοποιεί προηγούμενες εμπειρίες, επιτρέποντας πιο αποτελεσματική εκμάθηση και βελτιωμένη απόδοση δειγμάτων. Η διαμόρφωση ανταμοιβής περιλαμβάνει την παροχή πρόσθετων ανταμοιβών ή κυρώσεων για την καθοδήγηση της μαθησιακής διαδικασίας, την επιτάχυνση της σύγκλισης και την ενίσχυση της συμπεριφοράς του πράκτορα. Η μεταφορά μάθησης αξιοποιεί τη γνώση που αποκτήθηκε σε μια εργασία ή τομέα για να επιταχύνει τη μάθηση ή να βελτιώσει την απόδοση σε μια διαφορετική αλλά σχετική εργασία ή τομέα. Ο στόχος είναι να συγκριθούν και να αναλυθούν τα χαρακτηριστικά απόδοσης και σύγκλισης των μεθόδων που βασίζονται σε αξία, βασισμένες σε πολιτικές και μεθόδων κριτικής, ρίχνοντας φως στην καταλληλότητά τους για διαφορετικούς τύπους προβλημάτων RL. Αυτή η έρευνα θα διερευνήσει επίσης την ενσωμάτωση πρόσθετων τεχνικών για τη βελτίωση των μαθησιακών δυνατοτήτων και της αποτελεσματικότητας των αλγορίθμων RL. Μέσα από εκτεταμένους πειραματισμούς και αναλύσεις, θα αποκτήσουμε γνώσεις σχετικά με τις ανταλλαγές και τις εκτιμήσεις κατά την επιλογή και τη λεπτομερή ρύθμιση αλγορίθμων RL για διάφορες εφαρμογές του πραγματικού κόσμου. Τα ευρήματα αυτής της έρευνας θα συμβάλουν στην πρόοδο του σχεδιασμού και της κατανόησης του αλγορίθμου RL, ανοίγοντας το δρόμο για πιο αποτελεσματικούς και αποδοτικούς αυτόνομους παράγοντες μάθησης σε ένα ευρύ φάσμα τομέων.

3.7.1 Vanilla Policy Gradient - VPG

Το Vanilla Policy Gradient (VPG), γνωστό και ως REINFORCE[18], είναι ένας θεμελιώδης αλγόριθμος μάθησης ενίσχυσης που έχει χρησιμοποιηθεί ευρέως σε διάφορους τομείς. Το VPG είναι μια μέθοδος βασισμένη σε πολιτικές που στοχεύει στην εκμάθηση μιας βέλτιστης πολιτικής βελτιστοποιώντας απευθείας τις παραμέτρους πολιτικής με βάση τις παρατηρούμενες ανταμοιβές. Το VPG λειτουργεί συλλέγοντας επαναληπτικές τροχιές αλληλεπιδρώντας με το περιβάλλον χρησιμοποιώντας την τρέχουσα πολιτική και ενημερώνοντας τις παραμέτρους πολιτικής με βάση τα δεδομένα που συλλέγονται. Σε αντίθεση με τις μεθόδους που

βασίζονται σε αξία που εκτιμούν τη συνάρτηση αξίας, το VPG εστιάζει στην άμεση βελτιστοποίηση των παραμέτρων της πολιτικής για τη μεγιστοποίηση των αναμενόμενων σωρευτικών ανταμοιβών. Η βασική ιδέα πίσω από το VPG είναι να υπολογιστεί μια αμερόληπτη εκτίμηση της κλίσης πολιτικής, η οποία υποδεικνύει την κατεύθυνση της βελτίωσης της πολιτικής. Αυτό επιτυγχάνεται χρησιμοποιώντας το τέχνασμα αναλογίας πιθανότητας για την εξαγωγή του εκτιμητή κλίσης. Με τη δειγματοληψία ενεργειών από την πολιτική και τον υπολογισμό της κλίσης των αναμενόμενων ανταμοιβών σε σχέση με τις παραμέτρους πολιτικής, το VPG ενημερώνει τις παραμέτρους πολιτικής για να μεγιστοποιήσει τις αναμενόμενες ανταμοιβές. Ένα από τα βασικά πλεονεκτήματα του VPG είναι η απλότητα και η ευκολία εφαρμογής του. Ο αλγόριθμος είναι εννοιολογικά απλός και δεν απαιτεί σύνθετες προσεγγίσεις συναρτήσεων τιμών ή bootstrapping. Αυτή η απλότητα καθιστά το VPG ελκυστική επιλογή, ιδιαίτερα σε σενάρια όπου η δυναμική του περιβάλλοντος είναι πολύπλοκη ή ο χώρος κατάστασης είναι υψηλών διαστάσεων. Το VPG είναι επίσης γνωστό για την ικανότητά του να χειρίζεται αποτελεσματικά χώρους συνεχούς δράσης. Με την άμεση βελτιστοποίηση των παραμέτρων πολιτικής, το VPG φιλοξενεί φυσικά χώρους συνεχούς δράσης χωρίς την ανάγκη διακριτοποίησης ή πρόσθετων τεχνικών προσέγγισης. Αυτή η ιδιότητα καθιστά το VPG κατάλληλο για εργασίες που περιλαμβάνουν λεπτό έλεγχο ή απαιτούν ακριβείς επιλογές ενεργειών. Ωστόσο, το VPG υποφέρει από υψηλή διακύμανση στις εκτιμήσεις της κλίσης, η οποία μπορεί να οδηγήσει σε αργή σύγκλιση και ασταθή εκπαίδευση. Για να αντιμετωπιστεί αυτό το ζήτημα, έχουν προταθεί διάφορες τεχνικές όπως η βασική εκτίμηση, η κανονικοποίηση ανταμοιβής και η κανονικοποίηση της εντροπίας για τη μείωση της διακύμανσης και τη βελτίωση της ταχύτητας σύγκλισης του VPG. Σε αυτή τη διατριβή, στοχεύουμε να διερευνήσουμε την απόδοση και τα χαρακτηριστικά του VPG σε διαφορετικές εργασίες και περιβάλλοντα ενισχυτικής μάθησης. Θα διερευνήσουμε την επίδραση διαφόρων τεχνικών για τη μείωση της διακύμανσης, όπως οι μέθοδοι βασικής γραμμής και η κανονικοποίηση της εντροπίας, στη σύγκλιση και τη σταθερότητα του VPG. Επιπλέον, θα συγκρίνουμε το VPG με άλλους υπερσύγχρονους αλγόριθμους RL για να αξιολογήσουμε τα πλεονεκτήματα και τους περιορισμούς του. Τα ευρήματα αυτής της έρευνας θα συμβάλουν στη βαθύτερη κατανόηση του VPG και της εφαρμογής του σε διαφορετικά σενάρια RL. Αναλύοντας την απόδοση, την ταχύτητα σύγκλισης και την αποτελεσματικότητα του δείγματος του VPG, μπορούμε να αποκτήσουμε πληροφορίες για την αποτελεσματικότητά του και να εντοπίσουμε στρατηγικές για τη βελτίωση της απόδοσής του σε τομείς προκλήσεων. Συνολικά, το VPG προσφέρει μια απλή αλλά ισχυρή προσέγγιση για την ενίσχυση της μάθησης βελτιστοποιώντας άμεσα τις παραμέτρους πολιτικής. Με την ικανότητά του να χειρίζεται χώρους συνεχούς δράσης και την απλότητά του στην υλοποίηση, το VPG έχει χρησιμοποιηθεί και μελετηθεί ευρέως στην κοινότητα RL. Με την περαιτέρω διερεύνηση και αξιολόγηση του VPG, μπορούμε να προωθήσουμε την κατανόησή μας για αυτόν τον αλγόριθμο και να αποκαλύψουμε ευκαιρίες για βελτίωση και εφαρμογή του σε προβλήματα του πραγματικού κόσμου.Ο VPG είναι on policy αλγόριθμος και μπορεί να χρησιμοποιηθεί και σε συνεχή και σε διακριτά πεδία δράσης.

3.7.2 Trust Region Policy Optimization - TRPO

Το Trust Region Policy Optimization (TRPO) είναι ένας υπερσύγχρονος αλγόριθμος ενίσχυσης εκμάθησης που έχει κερδίσει σημαντική προσοχή λόγω της ικανότητάς του να παρέχει σταθερές και αποτελεσματικές ενημερώσεις πολιτικής. Το TRPO ανήκει στην οικογένεια των αλγορίθμων βελτιστοποίησης πολιτικής και στοχεύει να μάθει μια βέλτιστη πολιτική βελτιώνοντας επαναληπτικά τις παραμέτρους πολιτικής με βάση τις παρατηρούμενες ανταμοιβές. Η TRPO λειτουργεί συλλέγοντας τροχιές μέσω της αλληλεπίδρασης με το περιβάλλον χρησιμοποιώντας την τρέχουσα πολιτική. Στη συνέχεια χρησιμοποιεί αυτές τις τροχιές για να εκτιμήσει τα πλεονεκτήματα διαφορετικών ενεργειών και ενημερώνει τις παραμέτρους πολιτικής ανάλογα. Ένα από τα βασικά χαρακτηριστικά του TRPO είναι η έμφαση που δίνει στη διατήρηση μιας περιοχής εμπιστοσύνης, η οποία περιορίζει την έκταση των ενημερώσεων πολιτικής για να διασφαλίσει τη σταθερότητα και να αποτρέψει καταστροφικές καταρρεύσεις πολιτικής. Το κύριο πλεονέκτημα του TRPO είναι η ικανότητά του να παρέχει μονοτονική βελτίωση πολιτικής με εγγυημένα όρια απόδοσης. Περιορίζοντας τις ενημερώσεις πολιτικής σε μια περιοχή εμπιστοσύνης, το TRPO διασφαλίζει ότι η ενημερωμένη πολιτική δεν αποκλίνει πολύ από την προηγούμενη πολιτική. Αυτός ο περιορισμός συμβάλλει στη διατήρηση της σταθερότητας κατά τη διάρκεια της μαθησιακής διαδικασίας και αποφεύγει δραστικές αλλαγές πολιτικής που θα μπορούσαν να οδηγήσουν σε μη βέλτιστη ή μη ασφαλή συμπεριφορά. Το TRPO το επιτυγχάνει βελτιστοποιώντας μια συνάρτηση υποκατάστατου στόχου που προσεγγίζει τη βελτίωση της πολιτικής, λαμβάνοντας παράλληλα υπόψη τους περιορισμούς της περιοχής εμπιστοσύνης. Ο αλγόριθμος επιλύει ένα περιορισμένο πρόβλημα βελτιστοποίησης για την εύρεση της ενημέρωσης πολιτικής που μεγιστοποιεί τη συνάρτηση στόχου εντός της περιοχής αξιοπιστίας. Αυτή η προσέγγιση παρέχει ισχυρές και αξιόπιστες ενημερώσεις πολιτικής, ακόμη και σε περιβάλλοντα υψηλών διαστάσεων ή πολύπλοκα. Μια άλλη αξιοσημείωτη πτυχή του TRPO είναι η συμβατότητά του τόσο με διακριτούς όσο και με συνεχείς χώρους δράσης. Μπορεί να χειριστεί ένα ευρύ φάσμα εργασιών ενισχυτικής μάθησης, καθιστώντας το εφαρμόσιμο σε διάφορους τομείς και ρυθμίσεις προβλημάτων. Αυτή η ευελιξία καθιστά το TRPO μια δημοφιλή επιλογή για εργασίες που περιλαμβάνουν διακριτή λήψη αποφάσεων ή απαιτούν συνεχή έλεγχο. Ωστόσο, το TRPO μπορεί να είναι υπολογιστικά ακριβό λόγω της ανάγκης επίλυσης ενός περιορισμένου προβλήματος βελτιστοποίησης σε κάθε επανάληψη. Αυτή η υπολογιστική πολυπλοκότητα μπορεί να περιορίσει την επεκτασιμότητα του TRPO σε ορισμένα σενάρια, ειδικά όταν πρόκειται για περιβάλλοντα μεγάλης κλίμακας ή περίπλοκες πολιτικές. Ως εκ τούτου, οι ερευνητές έχουν προτείνει διάφορες τεχνικές και βελτιώσεις για την ενίσχυση της αποτελεσματικότητας και της επεκτασιμότητας του TRPO, όπως μεθόδους παραλληλοποίησης και προσέγγισης. Σε αυτή τη διατριβή, στοχεύουμε να διερευνήσουμε την απόδοση και τα χαρακτηριστικά του TRPO σε διαφορετικούς τομείς ενισχυτικής μάθησης και να αξιολογήσουμε τα δυνατά και τους περιορισμούς του. Θα διερευνήσουμε την επίδραση των περιορισμών της περιοχής εμπιστοσύνης στη σταθερότητα, την αποτελεσματικότητα του δείγματος και τις ιδιότητες σύγκλισης του TRPO. Επιπλέον, θα συγκρίνουμε το TRPO με άλλους προηγμένους αλγόριθμους RL για να αξιολογήσουμε την απόδοσή του και να εντοπίσουμε πιθανούς τομείς για βελτίωση. Τα ευρήματα αυτής της έρευνας θα συμβάλουν σε μια βαθύτερη κατανόηση του TRPO και της δυνατότητας εφαρμογής του σε διάφορα σενάρια RL. Αναλύοντας την απόδοση, τη σταθερότητα και την επεκτασιμότητα του αλγορίθμου, μπορούμε να αποκτήσουμε γνώσεις για την αποτελεσματικότητά του και να εντοπίσουμε

στρατηγικές για τη βελτιστοποίηση της απόδοσής του σε δύσκολα και πολύπλοκα περιβάλλοντα. Συνολικά, το TRPO προσφέρει μια βασική προσέγγιση για τη βελτιστοποίηση της πολιτικής, διατηρώντας μια περιοχή εμπιστοσύνης και παρέχοντας εγγυήσεις απόδοσης. Με τη σταθερότητα, την ευρωστία και τη συμβατότητά του με διαφορετικούς χώρους δράσης, ο TRPO έχει αναδειχθεί ως ένας εξέχων αλγόριθμος στον τομέα της ενισχυτικής μάθησης. Με την περαιτέρω διερεύνηση και αξιολόγηση του TRPO, μπορούμε να προωθήσουμε την κατανόησή μας για αυτόν τον αλγόριθμο και να διερευνήσουμε τις οδούς για τη βελτίωση και την εφαρμογή του σε προβλήματα του πραγματικού κόσμου.

3.7.3 Proximal Policy Optimization - PPO

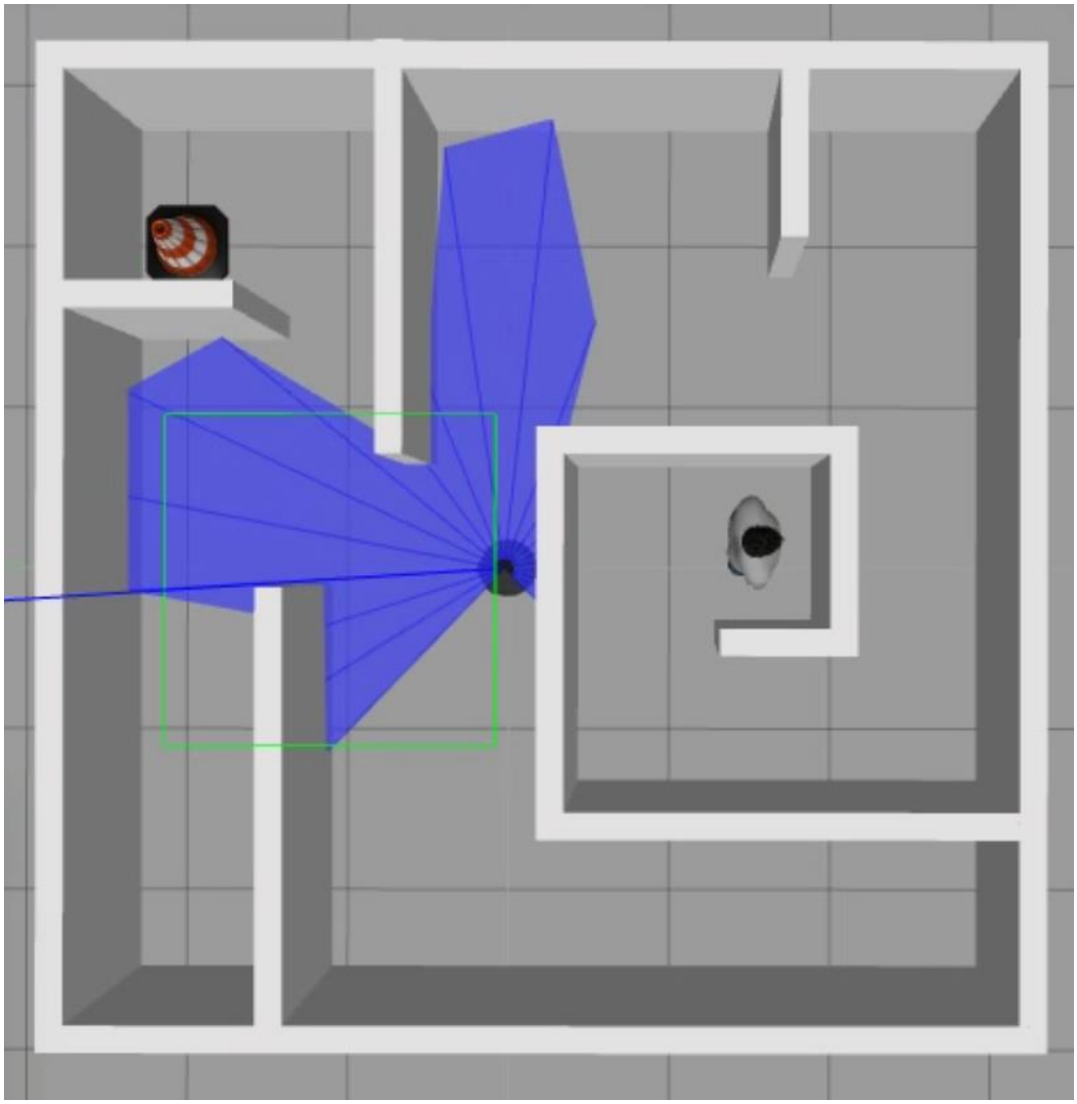
Ο Proximal Policy Optimization (PPO) είναι ένας υπερσύγχρονος αλγόριθμος ενισχυτικής μάθησης που έχει κερδίσει σημαντική προσοχή και δημοτικότητα τα τελευταία χρόνια. Ο PPO είναι μια μέθοδος βασισμένη σε πολιτικές που στοχεύει στην εκμάθηση μιας βέλτιστης πολιτικής βελτιστοποιώντας επαναληπτικά τις παραμέτρους πολιτικής με βάση τις παρατηρούμενες ανταμοιβές και τις μεταβάσεις κατάστασης. Ο PPO έχει σχεδιαστεί για να αντιμετωπίζει τις προκλήσεις της βελτιστοποίησης των πολιτικών στην ενισχυτική μάθηση, συμπεριλαμβανομένης της αντιστάθμισης μεταξύ εξερεύνησης και εκμετάλλευσης, της αποτελεσματικότητας του δείγματος και της σταθερότητας. Ο PPO αντιμετωπίζει αυτές τις προκλήσεις αξιοποιώντας μια νέα αντικειμενική συνάρτηση και έναν περιορισμό περιοχής εμπιστοσύνης. Η βασική ιδέα πίσω από τον PPO είναι να διασφαλιστεί ότι η ενημέρωση πολιτικής παραμένει σε μια περιοχή εμπιστοσύνης γύρω από την τρέχουσα πολιτική, αποτρέποντας δραστικές αλλαγές πολιτικής που μπορεί να οδηγήσουν σε αστάθεια ή καταστροφική λήθη. Αυτό επιτυγχάνεται μέσω της χρήσης μιας συνάρτησης υποκατάστατου στόχου που προσεγγίζει τη βελτίωση της πολιτικής, ενώ παραμένει κοντά στην προηγούμενη πολιτική. Ο PPO χρησιμοποιεί μια διαδικασία δύο βημάτων κατά τη διάρκεια κάθε επανάληψης. Πρώτον, συλλέγει μια παρτίδα δεδομένων αλληλεπιδρώντας με το περιβάλλον χρησιμοποιώντας την τρέχουσα πολιτική. Στη συνέχεια, εκτελεί πολλαπλές εποχές βελτιστοποίησης σε αυτά τα δεδομένα που συλλέγονται για να ενημερώσει τις παραμέτρους πολιτικής. Η διαδικασία βελτιστοποίησης μεγιστοποιεί τη συνάρτηση υποκατάστατου στόχου, ενώ ικανοποιεί τον περιορισμό της περιοχής εμπιστοσύνης, εξισορροπώντας αποτελεσματικά την εξερεύνηση και την εκμετάλλευση. Ένα από τα βασικά πλεονεκτήματα του PPO είναι η αποτελεσματικότητα του δείγματος. Με την επαναχρησιμοποίηση των συλλεγόμενων δεδομένων σε πολλές εποχές βελτιστοποίησης, ο PPO επιτυγχάνει υψηλότερη απόδοση δεδομένων σε σύγκριση με άλλους αλγόριθμους, μειώνοντας τον αριθμό των αλληλεπιδράσεων που απαιτούνται με το περιβάλλον. Αυτή η ιδιότητα καθιστά τον PPO ιδιαίτερα χρήσιμο σε σενάρια όπου η συλλογή δεδομένων είναι χρονοβόρα ή δαπανηρή. Επιπλέον, ο PPO προσφέρει καλή σταθερότητα κατά τη διάρκεια της εκπαίδευσης. Ο περιορισμός περιοχής εμπιστοσύνης διασφαλίζει ότι η ενημέρωση πολιτικής είναι ελεγχόμενη και σταδιακή, αποφεύγοντας μεγάλες αποκλίσεις πολιτικής που μπορεί να οδηγήσουν σε αστάθεια ή κακή σύγκλιση. Αυτή η σταθερότητα είναι ζωτικής σημασίας για προπόνηση μεγάλης διάρκειας ή όταν αντιμετωπίζετε σύνθετα περιβάλλοντα με αραιές ανταμοιβές. Ο PPO ενσωματώνει επίσης έναν μηχανισμό για

τον έλεγχο της ανταλλαγής μεταξύ εξερεύνησης και εκμετάλλευσης. Μέσω της χρήσης μιας υπερπαραμέτρου που ονομάζεται αναλογία κλιπ, ο PPO περιορίζει την ενημέρωση πολιτικής σε ένα συγκεκριμένο εύρος, αποτρέποντας υπερβολικά επιθετικές αλλαγές πολιτικής. Αυτό επιτρέπει την ελεγχόμενη εξερεύνηση και διασφαλίζει την ομαλή μετάβαση μεταξύ διαφορετικών επαναλήψεων πολιτικής. Σε αυτή τη διατριβή, στοχεύουμε να διερευνήσουμε και να αναλύσουμε την απόδοση και τα χαρακτηριστικά του PPO σε διάφορες εργασίες και τομείς ενισχυτικής μάθησης. Θα διερευνήσουμε την επίδραση διαφορετικών υπερπαραμέτρων, όπως η αναλογία κλιπ, ο ρυθμός εκμάθησης και η αρχιτεκτονική δικτύου, στη σύγκλιση και την απόδοση του PPO. Επιπλέον, θα συγκρίνουμε τον PPO με άλλους προηγμένους αλγόριθμους RL για να αξιολογήσουμε τα πλεονεκτήματα και τους περιορισμούς του. Τα ευρήματα αυτής της έρευνας θα συμβάλουν στη βαθύτερη κατανόηση του PPO και της δυνατότητας εφαρμογής του σε διαφορετικά σενάρια RL. Αναλύοντας την απόδοση, την ταχύτητα σύγκλισης και την αποτελεσματικότητα του δείγματος του PPO, μπορούμε να αποκτήσουμε πληροφορίες για τους παράγοντες που επηρεάζουν την αποτελεσματικότητά του και να προσδιορίσουμε κατευθυντήριες γραμμές για τη λεπτομέρεια και τη βελτιστοποίηση του PPO για συγκεκριμένους τομείς προβλημάτων. Συνολικά, ο PPO προσφέρει μια πολλά υποσχόμενη προσέγγιση για την ενισχυτική μάθηση με την αποτελεσματικότητα του δείγματος, τη σταθερότητα και την ελεγχόμενη εξερεύνηση. Με την περαιτέρω διερεύνηση και αξιολόγηση του PPO, μπορούμε να προωθήσουμε την κατανόησή μας για αυτόν τον αλγόριθμο και να ανοίξουμε το δρόμο για την αποτελεσματική εφαρμογή του σε σενάρια πραγματικού κόσμου.

4 Πειράματα

4.1 Το περιβάλλον μου

Στο περιβάλλον που έχω δημιουργήσει υπάρχει ένα ρομπότ το οποίο μπορεί να κινηθεί στον τρισδιάστατο χώρο και γνωρίζει την απόστασή του από εμπόδια και τοίχους μέσω αισθητήρων απόστασης. Το κάθε επίπεδο αποτελείται από το ρομπότ, τους τοίχους, έναν κώνο και τέλος έναν άνθρωπο. Το ρομπότ είναι ο πράκτοράς μας και έχει τελικό στόχο να φτάσει στον άνθρωπο χωρίς να χτυπήσει σε κάποιον τοίχο. Για να φτάσει στον άνθρωπο πρέπει πρώτα να φτάσει στον κώνο. Όταν βρει τον κώνο ή τον άνθρωπο ο πράκτορας παίρνει θετικό βραβείο, ενώ παίρνει αρνητικό όταν χτυπήσει σε τοίχο.



Το περιβάλλον έχει διακριτό χώρο ενεργειών. Συγκεκριμένα έχει 3 ενέργειες : Μπροστά, Αριστερά, Δεξιά. Έχει συνεχή χώρο παρατηρήσεων. Οι παρατηρήσεις έχουν 4 διαστάσεις : 2 από τις τιμές του laser απόστασης, 1 η απόσταση από τον στόχο και 1 ο προσανατολισμός. Οι τιμές που δέχεται είναι από 0.0599 ως 20.

Πρώτα θέτουμε αρχικές τιμές σε μερικές μεταβλητές. Κάποιες boolean μεταβλητές όπως το αν έχει βρει τον στόχο τις θέτουμε σε false. Μετά θέτουμε τις συντεταγμένες του πρώτου στόχου και κάποιες τιμές που βοηθάνε στο να ξέρουμε πόσες φορές βρίσκει τον πρώτο και τον δεύτερο στόχο.

```
29      def __init__(self,  observation_size=0,min_range = 0.13,max_range = 3.5,max_env_size=3,got_key=F
30          # Launch the simulation with the given launchfile name
31          gazebo_env.GazeboEnv.__init__(self, "GazeboRoundTurtlebotLidar_v0.launch")
32          self.vel_pub = rospy.Publisher('/mobile_base/commands/velocity', Twist, queue_size=5)
33          self.unpause = rospy.ServiceProxy('/gazebo/unpause_physics', Empty)
34          self.pause = rospy.ServiceProxy('/gazebo/pause_physics', Empty)
35          self.reset_proxy = rospy.ServiceProxy('/gazebo/reset_simulation', Empty)
36
37          self.action_space = spaces.Discrete(3) #F,L,R
38
39          self.got_key = got_key
40          self.got_key_reward = got_key_reward
41          self.opened_door = opened_door
42          self.reached_human = reached_human
43
44
45          self.goal_x=goal_x
46          self.goal_y=goal_y
47          self.heading = heading
48          self.steps = 0
49          self.current_steps = 0
50          self.iterations = 1
51          self.num_keys = 0
52          self.successes = 0
53
54          self.observation_size = observation_size
55          self.min_range = min_range
56          self.max_range = max_range
57          self.max_env_size = max_env_size
58          self.observation_space = spaces.Box(low=0.0599, high=20.0, shape=(4, ), dtype=np.float64)
```

Η συνάρτηση discretize_observation παίρνει τις συνεχείς τιμές του laser και επιστρέφει μια λίστα από διακριτές τιμές.

```
79      def discretize_observation(self,data,new_ranges):
80          discretized_ranges = []
81          min_range = 0.2
82          done = False
83          mod = len(data.ranges)/new_ranges
84          for i, item in enumerate(data.ranges):
85              if (i%mod==0):
86                  if data.ranges[i] == float ('Inf'):
87                      discretized_ranges.append(6)
88                  elif np.isnan(data.ranges[i]):
89                      discretized_ranges.append(0)
90                  else:
91                      discretized_ranges.append(int(data.ranges[i]))
92              if (min_range > data.ranges[i] > 0):
93                  done = True
94
95
96          return discretized_ranges,done
```

Το πιο σημαντικό κομμάτι ενός αλγορίθμου ενισχυτικής εκμάθησης με τη χρήση του OpenAI Gym είναι η συνάρτηση step. Σε αυτό το σημείο ο αλγόριθμος επιλέγει ποια δράση θα πάρει και αλλάζει η κατάσταση του περιβάλλοντος. Αρχικά βάζουμε τις αρχικές συντεταγμένες του τοίχου.

```python
    def step(self, action):
        if not self.got_key:
            set_model_state = rospy.ServiceProxy('/gazebo/set_model_state', SetModelState)

            # Create a ModelState message
            model_state_msg = ModelState()

            # Set the name of the model to be changed
            model_state_msg.model_name = 'grey_wall'

            # Set the new pose of the model
            new_pose = Pose()
            # new_pose.position = Point(x=-20, y=-10, z=0.0) # Level 4
            # new_pose.position = Point(x=-2, y=-1, z=0.0) # Level 3
            new_pose.position = Point(x=4, y=1, z=0.0) # Level 2
            # new_pose.position = Point(x=2.85, y=-2.6, z=0.0)  # Level 1
            new_pose.orientation = Quaternion(x=0.0, y=0.0, z=0, w=1.0)
            # new_pose.orientation = Quaternion(x=0.0, y=0.0, z=1.57, w=1.0) # Level 1
            model_state_msg.pose = new_pose
            resp = set_model_state(model_state_msg)
        rospy.wait_for_service('/gazebo/unpause_physics')
        try:
            self.unpause()
        except (rospy.ServiceException) as e:
            print ("/gazebo/unpause_physics service call failed")
```

Παρακάτω είναι οι 3 δράσεις που μπορεί να πάρει το ρομπότ.

```python
            if action == 0: #FORWARD
                vel_cmd = Twist()
                vel_cmd.linear.x = 0.3  #0.3
                vel_cmd.angular.z = 0.0
                self.vel_pub.publish(vel_cmd)
            elif action == 1: #LEFT
                vel_cmd = Twist()
                vel_cmd.linear.x = 0.0
                vel_cmd.angular.z = 0.3 # 0.3
                self.vel_pub.publish(vel_cmd)
            elif action == 2: #RIGHT
                vel_cmd = Twist()
                vel_cmd.linear.x = 0.0
                vel_cmd.angular.z = -0.3  # -0.3
                self.vel_pub.publish(vel_cmd)
```

Εδώ ο κώδικας παίρνει δεδομένα από τους αισθητήρες που είναι ενεργοί στο Gazebo μέσω του ROS. Το data είναι τα δεδομένα του αισθητήρα laser του turtlebot που μετράει την απόσταση από αντικείμενα. Έπειτα είναι ο αισθητήρας που έβαλα στον κώνο για να ανιχνεύει την επαφή με το ρομπότ. Τέλος είναι ο αντίστοιχος αισθητήρας επαφής του ανθρώπου.

```
147             data = None
148             while data is None:
149                 try:
150                     data = rospy.wait_for_message('/scan', LaserScan, timeout=5)
151                 except:
152                     pass
153 
154 
155             #HANDLE
156 
157             handle_contact =None
158             if not self.got_key :
159                 while handle_contact is None:
160                     try:
161                         handle_contact =rospy.wait_for_message('/my_contact_handle', ContactsState, timeout=5)
162                     except:
163                         pass
164 
165 
166             #HANDLE
167 
168             person_contact =None
169             if self.got_key :
170                while person_contact is None:
171                     try:
172                         person_contact = rospy.wait_for_message('/my_contact_person', ContactsState, timeout=5)
173                     except:
174                         pass
```

Με το που πάρουμε σήμα από τον αισθητήρα στον κώνο ελέγχουμε αν ο κώνος έχει έρθει σε επαφή με κάτι άλλο εκτός από το έδαφος. Αν ισχύει αυτό τότε το ρομπότ έχει ακουμπήσει τον κώνο (δεν μπορεί να είναι άλλο αντικείμενο) . Με το που γίνει αυτό αλλάζουμε θέση τον κώνο αλλά και τον τοίχο και τα βάζουμε κάπου εκτός πίστας. Ταυτόχρονα αλλάζουμε την μεταβλητή self.got_key απο false σε true. Όταν γίνει αυτό αλλάζει ο στόχος από τον κώνο στον άνθρωπο. Αυτό γίνεται βάζοντας τις μεταβλητές self.goal_x , self.goal_y τις συντεταγμένες του turtlebot. Τέλος ελέγχουμε αν το ρομπότ έχει έρθει σε επαφή με τον άνθρωπο και βάζουμε την μεταβλητή self.reached_human σε true.

```
            state = state + [ goal_distance , heading ]


            self.steps += 1
            self.current_steps += 1


              #FOUND HANDLE!!!
            if not self.got_key :
                if handle_contact !=None:
                    if handle_contact.states[0].collision2_name != "ground_plane::link::collision":
                        self.got_key=True
                        self.num_keys += 1

                        reward = 10000

                                        # Initialize Gazebo model state service client
                        set_model_state = rospy.ServiceProxy('/gazebo/set_model_state', SetModelState)

                        # Create a ModelState message
                        model_state_msg = ModelState()

                        # Set the name of the model to be changed
                        model_state_msg.model_name = 'door_handle'

                        # Set the new pose of the model
                        new_pose = Pose()
                        new_pose.position = Point(x=10.0, y=10.0, z=0.0)
                    new_pose.orientation = Quaternion(x=0.0, y=0.0, z=0.0, w=1.0)
                    model_state_msg.pose = new_pose
                    resp = set_model_state(model_state_msg)

                    #WALL
                    set_model_state = rospy.ServiceProxy('/gazebo/set_model_state', SetModelState)

                    # Create a ModelState message
                    model_state_msg = ModelState()

                    # Set the name of the model to be changed
                    model_state_msg.model_name = 'grey_wall'

                    # Set the new pose of the model
                    new_pose = Pose()
                    new_pose.position = Point(x=12.85, y=-12.6, z=0.0)
                    new_pose.orientation = Quaternion(x=0.0, y=0.0, z=0, w=1.0)
                    model_state_msg.pose = new_pose
                    resp = set_model_state(model_state_msg)
                        # set_model_state = rospy.ServiceProxy('/gazebo/set_model_state', SetModelState)
```

```
313             elif self.got_key:
314                 rospy.wait_for_service('/gazebo/get_model_state')
315                 get_model_state = rospy.ServiceProxy('/gazebo/get_model_state', GetModelState)
316 
317                 model_name = "person_standing"
318                 reference_frame = 'world'
319 
320                 response = get_model_state(model_name=model_name, relative_entity_name=reference_frame)
321                 self.goal_x = response.pose.position.x
322                 self.goal_y = response.pose.position.y
323 
324                 if person_contact !=None:
325                     if person_contact.states[0].collision2_name != "ground_plane::link::collision":
326                         reward = 40000
327                         self.reached_human= True
328                         self.successes += 1
329                         done = True
```

Αυτό το κομμάτι του κώδικα εκτυπώνει τα ποσοστά επιτυχίας (πόσες φορές βρήκε τον κώνο και πόσες φορές τον άνθρωπο) . Το κάνει αυτό ανά 1000 βήματα γιατί τόσο κρατάει μια εποχή.

```
332             if self.steps == 1000 :
333                 print ("KEYS: {} / {} ".format(self.num_keys,self.iterations))
334                 print(("{}  % " .format((self.num_keys / self.iterations) *100) ))
335                 print ("SUCCESSES : {} / {} ".format(self.successes,self.iterations))
336                 print(("{}  % " .format((self.successes / self.iterations) *100) ))
337                 self.steps = 0
338                 self.iterations = 1
339                 self.num_keys = 0
340                 self.successes = 0
341                 self.current_steps = 0
```

Ακόμα ένα βασικό κομμάτι του κώδικα είναι η συνάρτηση reset. Εδώ το περιβάλλον επανέρχεται στην αρχική του κατάσταση. Η reset καλείται κάθε φορά που τελειώνει μια επανάληψη του αλγορίθμου. Αυτό γίνεται επειδή το ρομπότ βρήκε τον άνθρωπο ,επειδή το ρομπότ χτύπησε στον τοίχο ή γιατί έγιναν πάνω από 1000 βήματα σε μια επανάληψη που είναι το ανώτατο όριο. Στην reset όλα τα αντικείμενα πηγαίνουν στις αρχικές τους θέσεις και όλες οι μεταβλητές παίρνουν τις αρχικές τους τιμές.

```
371     def reset(self):
372
373         rospy.wait_for_service('/gazebo/reset_simulation')
374         try:
375             self.reset_proxy()
376         except (rospy.ServiceException) as e:
377             print ("/gazebo/reset_simulation service call failed")
378
379         # Unpause simulation to make observation
380         rospy.wait_for_service('/gazebo/unpause_physics')
381         try:
382             self.unpause()
383         except (rospy.ServiceException) as e:
384             print ("/gazebo/unpause_physics service call failed")
385
386
387
388
389
390
391         if self.got_key:
392             set_model_state = rospy.ServiceProxy('/gazebo/set_model_state', SetModelState)
393
394             # Create a ModelState message
395             model_state_msg = ModelState()
396
397             # Set the name of the model to be changed
398             model_state_msg.model_name = 'door_handle'
```

```
399
400                 # Set the new pose of the model
401                 new_pose = Pose()
402                 # new_pose.position = Point(x=2.2, y=1.5, z=  0 )      # Level 4
403                 # new_pose.position = Point(x=2, y=2, z=  0 )      # Level 3
404                 new_pose.position = Point(x=4, y=-2, z=  0 )      # Level 2
405                 # new_pose.position = Point(x=2.5, y=-3.3, z=  0 )      # Level 1
406                 new_pose.orientation = Quaternion(x=0, y= 0 , z=0, w=1.0)
407                 model_state_msg.pose = new_pose
408                 resp = set_model_state(model_state_msg)
409
410
411
412
413                 self.got_key = False
414                 self.got_key_reward = False
415                 self.opened_door = False
416                 self.reached_human = False
417
418
419
420             self.got_key = False
421             self.got_key_reward = False
422             self.opened_door = False
423             self.reached_human = False
424
425
426             self.got_key = False
```

```
            self.got_key_reward = False
            self.opened_door = False
            self.reached_human = False
            self.current_steps = 0


            # # Level 4
            # self.goal_x = 2.2
            # self.goal_y = 1.5

            # Level 3
            # self.goal_x = 2
            # self.goal_y = 2

            # # Level 2
            self.goal_x = 4
            self.goal_y = -2

            #Level 1
            # self.goal_x = 2.5
            # self.goal_y = -3.3

            self.heading = -1


            data = None
            while data is None:
                try:
                    data = rospy.wait_for_message('/scan', LaserScan, timeout=5)
                except:
                    pass


        rospy.wait_for_service('/gazebo/get_model_state')
        get_model_state = rospy.ServiceProxy('/gazebo/get_model_state', GetModelState)

        model_name = 'mobile_base'
        reference_frame = 'world'

        response = get_model_state(model_name=model_name, relative_entity_name=reference_frame)
        turtle_x = response.pose.position.x
        turtle_y = response.pose.position.y
        turtle_z = response.pose.position.z


        #DISTANCE TO HANDLE

        goal_distance = math.sqrt((turtle_x - self.goal_x)**2 + (turtle_y - self.goal_y)**2)


        rospy.wait_for_service('/gazebo/pause_physics')
        try:
            self.pause()
        except (rospy.ServiceException) as e:
            print ("/gazebo/pause_physics service call failed")

        set_model_state = rospy.ServiceProxy('/gazebo/set_model_state', SetModelState)

```

```
            # Create a ModelState message
            model_state_msg = ModelState()

            # Set the name of the model to be changed
            model_state_msg.model_name = 'grey_wall'

            # Set the new pose of the model
            new_pose = Pose()
            # new_pose.position = Point(x=-20, y=-10, z=0.0)  # Level 4
            # new_pose.position = Point(x=-2, y=-1, z=0.0)  # Level 3
            new_pose.position = Point(x=4, y=1, z=0.0)  # Level 2
            # new_pose.position = Point(x=2.85, y=-2.6, z=0.0)  # Level 1
            new_pose.orientation = Quaternion(x=0.0, y=0.0, z=0, w=1.0)
            # new_pose.orientation = Quaternion(x=0.0, y=0.0, z=1.57, w=1.0) # Level 1
            model_state_msg.pose = new_pose
            resp = set_model_state(model_state_msg)

            state ,done= self.discretize_observation(data,2)

            state = state + [ goal_distance , self.heading]

            return np.asarray(state,dtype=float)
```

Εδώ βλέπουμε κομμάτι κώδικα xml της προσομοίωσης του gazebo. Στο αρχείο αυτό βρίσκονται πληροφορίες για όλα τα στοιχεία της προσομοίωσης. Αυτός ο κώδικας περιγράφει τον αισθητήρα επαφής που θα μπει στον κώνο.

```
<sensor name='my_contact_handle' type='contact'>
  <always_on>1</always_on>
  <update_rate>5</update_rate>
  <contact>
      <collision>handle_collision</collision>
      <topic>/my_contact_handle</topic>
  </contact>
  <plugin name='my_contact_plugin' filename='libgazebo_ros_bumper.so'>
      <bumperTopicName>my_contact_handle</bumperTopicName>
      <frameName>world</frameName>
  </plugin>
</sensor>
```

4.2 Εκπαίδευση

Η εκπαίδευση του μοντέλου γίνεται με τη βοήθεια του spinup. Αρχικά κάνουμε import τους αλγορίθμους που θέλουμε από το spinup καθώς και τις υπόλοιπες βιβλιοθήκες που χρειαζόμαστε. Δημιουργούμε το περιβάλλον με την εντολή gym.make. Τέλος τρέχουμε τον αλγόριθμο που θέλουμε με τις παραμέτρους που επιλέγουμε.

```
env_fn = lambda : gym.make('GazeboRoundTurtlebotLidar-v0')

ac_kwargs = dict(hidden_sizes=[256,256], activation=tf.nn.relu)#64,64

logger_kwargs = dict(output_dir='/home/jimycool/Training/Level_2_ppo2', exp_name='PPO')

ppo(env_fn=env_fn,    ac_kwargs=ac_kwargs,    steps_per_epoch=1000,    epochs=200,    pi_lr=0.0005,
vf_lr=0.003, clip_ratio=0.3, target_kl=0.05, lam=0.999,gamma=0.999,logger_kwargs=logger_kwargs)
```

4.3 Βραβεία

Τα βραβεία είναι ένα από τα πιο σημαντικά τμήματα ενός αλγορίθμου RL. Η σωστή επιλογή των βραβείων μπορεί να οδηγήσουν στην σωστή, γρήγορη και σταθερή λύση ενός προβλήματος. Επομένως η επιλογή των βραβείων είναι πολύ σημαντική και συχνά μια διαδικασία που χρειάζεται πολλές δοκιμές μέχρι να βρεθεί ικανοποιητικό αποτέλεσμα. Στο δικό μου περιβάλλον επέλεξα να χωρίσω το βραβείο σε 3 κομμάτια. Το ένα κομμάτι δίνει βραβείο με βάση τον σωστό προσανατολισμό του ρομπότ. Το δεύτερο κομμάτι δίνει βραβείο ανάλογα με το πόσο κοντά βρίσκεται το ρομπότ στον στόχο. Το τρίτο κομμάτι εξαρτάται από την δράση που θα επιλέξει ο αλγόριθμος. Αν επιλέξει να πάει ευθεία παίρνει +100 , αν στρίψει -150 , αν ακουμπήσει σε τοίχο -10.000 , αν βρει τον κώνο +10.000 και αν βρει τον άνθρωπο +40.000 .

```
178             #TURTLEBOT POSITION
179
180             rospy.wait_for_service('/gazebo/get_model_state')
181             get_model_state = rospy.ServiceProxy('/gazebo/get_model_state', GetModelState)
182
183             model_name = 'mobile_base'
184             reference_frame = 'world'
185
186             response = get_model_state(model_name=model_name, relative_entity_name=reference_frame)
187             turtle_x = response.pose.position.x
188             turtle_y = response.pose.position.y
189             turtle_z = response.pose.position.z
190
191
192             orientation = response.pose.orientation
193             orientation_list = [orientation.x, orientation.y, orientation.z, orientation.w]
194             x=orientation.x
195             y=orientation.y
196             z=orientation.z
197             w=orientation.w
198             # _, _, yaw = euler_from_quaternion(orientation_list)
199             t0 = +2.0 * (w * x + y * z)
200             t1 = +1.0 - 2.0 * (x * x + y * y)
201             roll = math.atan2(t0, t1)
202
203             t2 = +2.0 * (w * y - z * x)
204             t2 = +1.0 if t2 > +1.0 else t2
205             t2 = -1.0 if t2 < -1.0 else t2
206               pitch = math.asin(t2)
207
208               t3 = +2.0 * (w * z + x * y)
209               t4 = +1.0 - 2.0 * (y * y + z * z)
210               yaw = math.atan2(t3, t4)
211
212               goal_angle = math.atan2(self.goal_y - turtle_y, self.goal_x - turtle_x)
213
214               heading = goal_angle - yaw
215               if heading > pi:
216                   heading -= 2 * pi
217
218               elif heading < -pi:
219                   heading += 2 * pi
220
221
222               #if heading = 0 we are facing the target
223
224               heading = round(heading, 2)         # we want this close to 0 ( low abs value)
225
226               self.heading = heading
227
228               if heading != 0 :
229                   heading_reward = 5* 1 / abs(heading)  #5
230               else :
231                   heading_reward = 100  #100
```

```
234             #DISTANCE TO HANDLE
235
236             goal_distance = math.sqrt((turtle_x - self.goal_x)**2 + (turtle_y - self.goal_y)**2)
237             # Define the scale factor for the reward
238             scale = 200  #100   200 4room
239
240             # Compute the reward as a negative exponential of the distance
241             distance_reward =scale* math.exp(- goal_distance*2)
```

```
344               if not done:
345                   if  self.got_key_reward == False and self.got_key==True :
346                       reward = 10000 #10000
347                       self.got_key_reward = True
348                   elif action == 0:
349                       reward = 100 #-0.5*self.current_steps  # +5   100
350                   else:
351                       reward =   -150 #-1*self.current_steps   # +1    -150
352
353               else:
354                   self.iterations += 1
355                   if  self.got_key_reward == False and self.got_key==True :
356                       reward = 10000
357                       self.got_key_reward = True
358                   if self.opened_door:                          #reward for o
359                       reward = 2000
360                   if self.reached_human:
361                       reward = 40000  #20000
362                   else:                                  # negative reward
363                       reward = -10000 # -6000
364
365
366
367               reward = reward + distance_reward + heading_reward
```

4.4 Υπερπαράμετροι

Κάθε αλγόριθμος RL έχει τις δικές τους υπερπαραμέτρους. Παρακάτω βλέπουμε τις υπερπαραμέτρους της PPO σύμφωνα με την εκδοχή του SpinningUp. Το ac_kwargs έχει να κάνει με το μέγεθος των κρυφών επιπέδων του νευρωνικού δικτύου (256,256) εδώ. Steps_per_epoch δείχνει πόσα βήματα του αλγορίθμου θα τρέχουν σε κάθε

εποχή (εδώ 1000). Αν είναι πολύ μικρό ο αλγόριθμος δεν προλαβαίνει να μάθει και αν είναι πολύ μεγάλο αργεί πολύ να μάθει. Οι εποχές δείχνουν πόσο συχνά ανανεώνεται η πολιτική του αλγορίθμου. Το clip_ratio (ανάμεσα στο 0.1-0.3) δείχνει το πόσο μπορεί να απομακρυνθεί η καινούργια πολιτική από την παλιά.

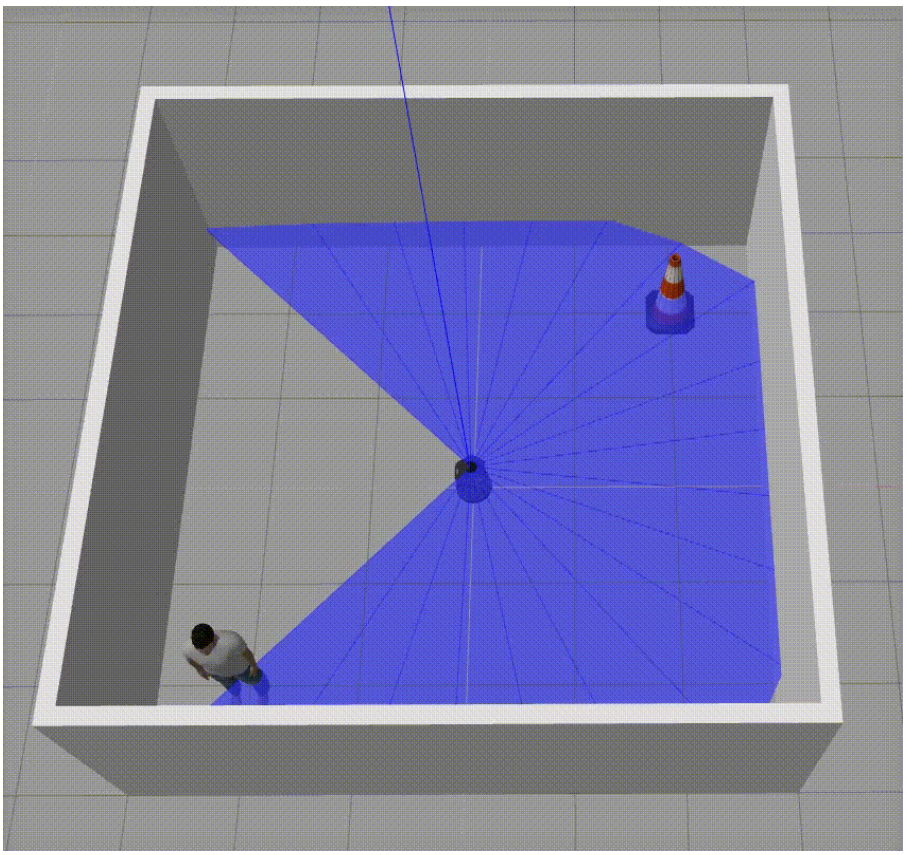
```
ppo(env_fn=env_fn, ac_kwargs=ac_kwargs, steps_per_epoch=1000,
epochs=200, pi_lr=0.0005, vf_lr=0.003, clip_ratio=0.3, target_kl=0.05,
lam=0.999,gamma=0.999,logger_kwargs=logger_kwargs)
```

5 Μετρήσεις

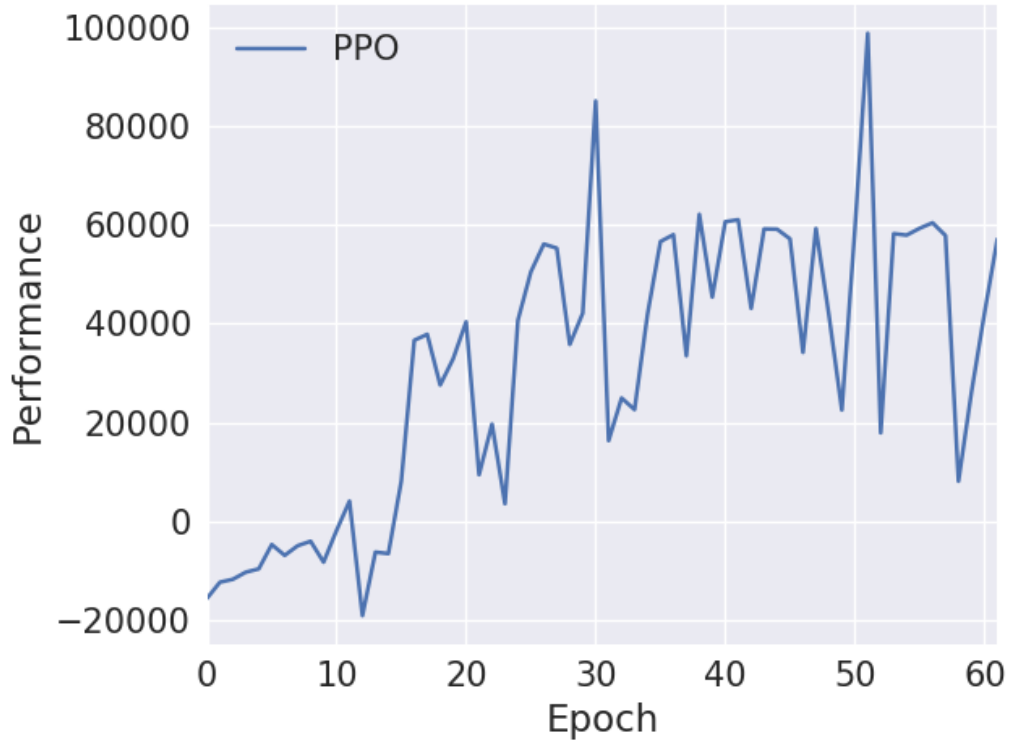
5.1 Εκμάθηση με PPO

Στο περιβάλλον αυτό το ρομπότ πρέπει να μάθει να βρίσκει συγκεκριμένους στόχους χωρίς να βρίσκει σε τοίχους. Και στα 3 επίπεδα δυσκολίας πρώτα πρέπει να βρει τον κώνο και μετά να πάει στον άνθρωπο. Στο πρώτο δεν έχει τοίχους ενδιάμεσα και είναι το πιο απλό. Στο δεύτερο πρέπει να μάθει να πηγαίνει μέχρι το τέρμα της ευθείας παρόλο που σημαίνει ότι θα πάρει μικρότερο βραβείο για μια μικρή περίοδο (παίρνει μεγαλύτερο βραβείο όσο πιο κοντά φτάνει στον στόχο και όσο πιο καλό προσανατολισμό έχει σε σχέση με τον στόχο). Όταν φτάσει στον κώνο φεύγει ο τοίχος μπροστά από τον άνθρωπο και πρέπει να μάθει να κάνει αναστροφή 180 μοίρες για να φτάσει στον άνθρωπο (αυτό είναι σχετικά δύσκολο γιατί παίρνει θετικό βραβείο όταν πηγαίνει ευθεία και αρνητικό όταν στρίβει). Το τρίτο επίπεδο είναι παρόμοιο απλά σε πιο δύσκολη διαδρομή.
Στις γραφικές παραστάσεις υπάρχουν 3 σημαντικές τιμές. Αρχικά ο μέσος όρος του βραβείου είναι -10.000 γιατί τόσο είναι το βραβείο για σύγκρουση με τοίχο. Μετά σημαντική είναι η περιοχή κοντά στο 0-10.000 γιατί σημαίνει ότι έχει αρχίσει να βρίσκει τον κώνο που δίνει βραβείο 10.000 (αλλά θα χάνει και 10.000 αν βρει σε τοίχο πριν βρει τον τελικό στόχο). Τέλος βραβεία κοντά στα 40.000 δείχνουν ότι έχει μάθει να βρίσκει τον άνθρωπο που δίνει βραβείο 40.000.
Στο πρώτο επίπεδο θέλει 15 εποχές για τον κώνο άλλα και τον άνθρωπο. Στο δεύτερο επίπεδο θέλει 10 εποχές για τον πρώτο στόχο και 20 για τον τελικό στόχο. Στο τρίτο επίπεδο θέλει 10 εποχές για να βρει τον κώνο αλλά χρειάζεται 50 εποχές για να βρει τον άνθρωπο.
Στο πρώτο επίπεδο 90% των φορών θα φτάσει στον τελικό στόχο. Στο δεύτερο 70% και στο τρίτο 40%.
Το τρίτο επίπεδο είναι ξεκάθαρα το πιο δύσκολο για τον agent γιατί έχει το χαμηλότερο ποσοστό (παρόλο που είχε πιο πολύ χρόνο εκπαίδευσης) και γιατί θέλει περισσότερες εποχές να αρχίσει να βρίσκει τον τελικό στόχο.

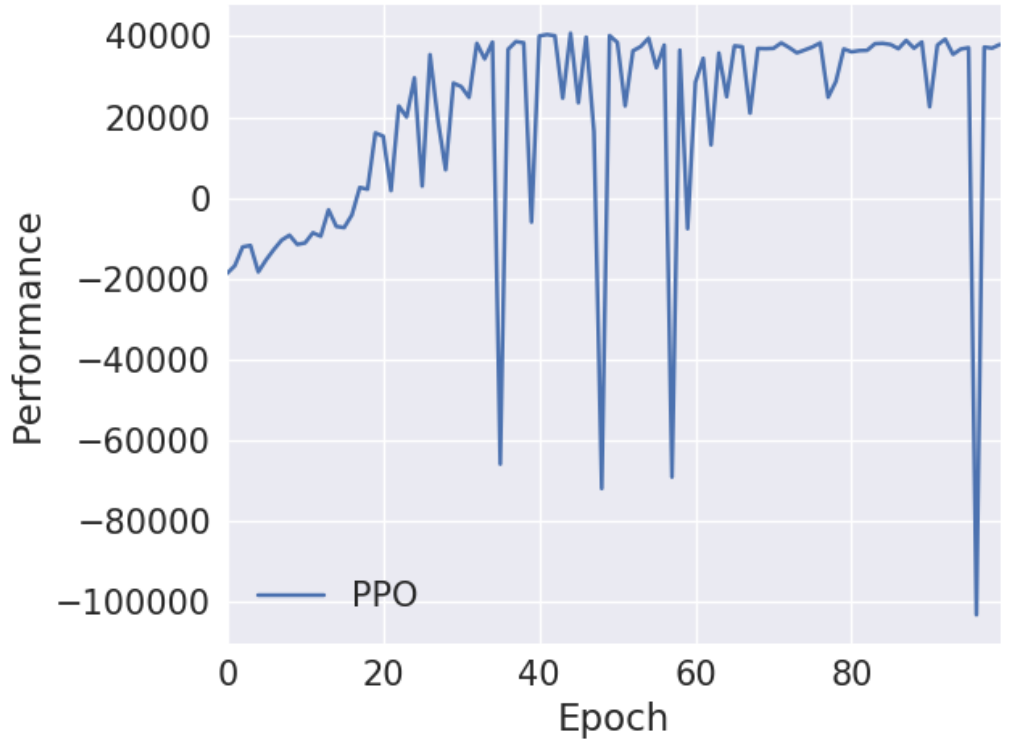
Επίπεδο 1



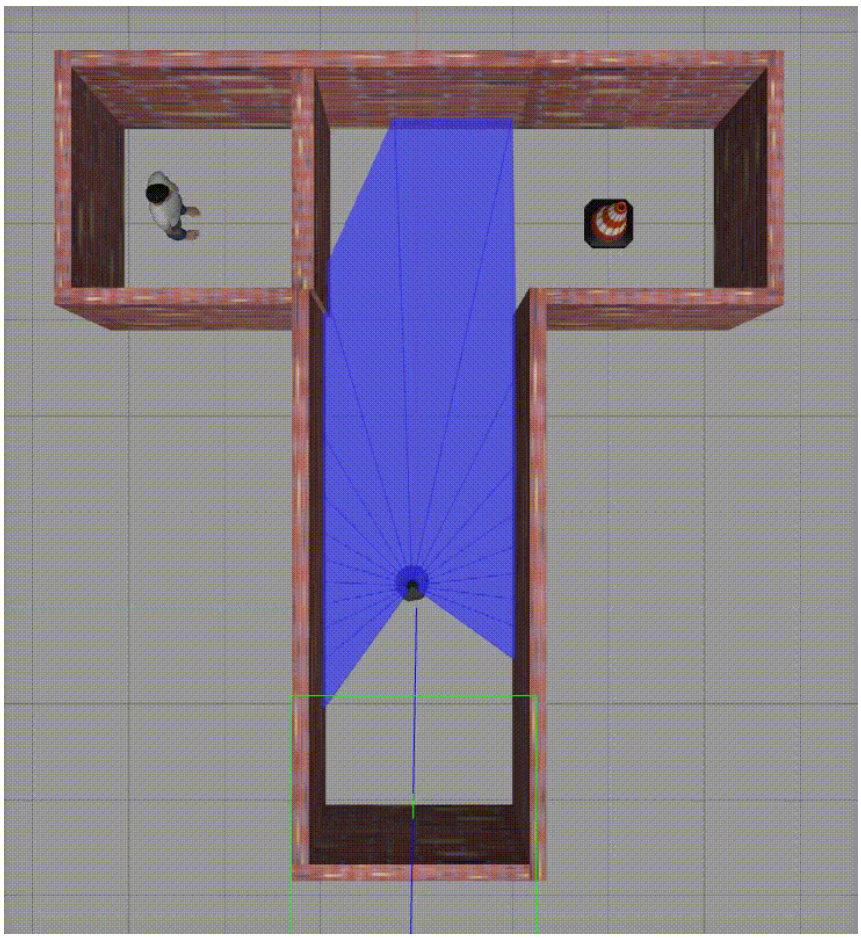
Αρχικά προπόνησα με τα ίδια βραβεία με τα υπόλοιπα επίπεδα και βγήκε αυτή η γραφική παράσταση:

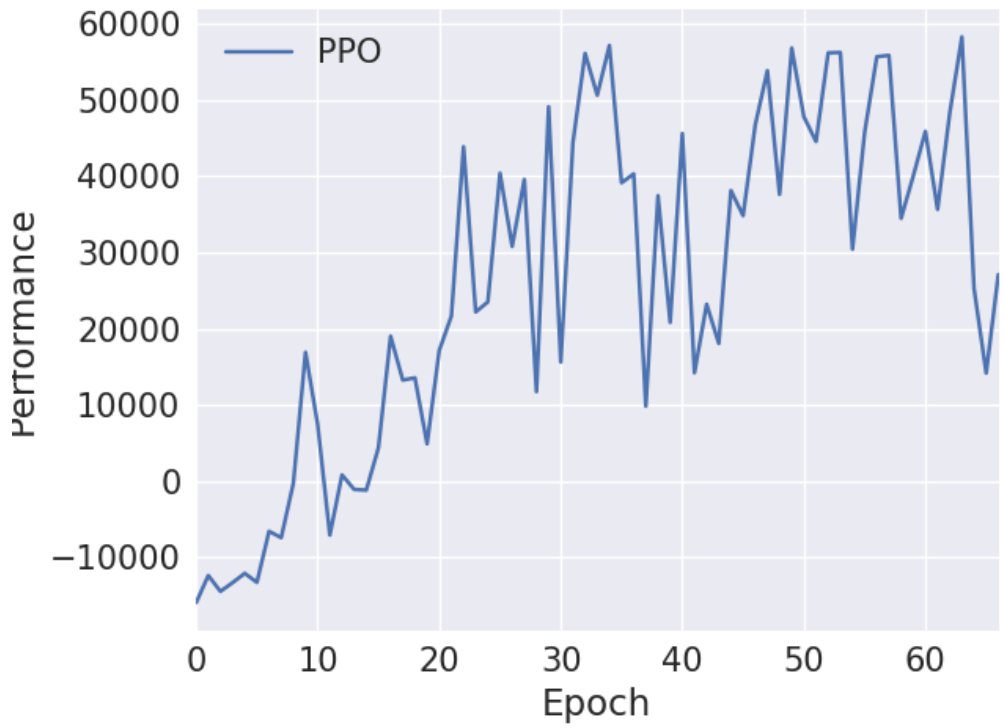


Όμως το ρομπότ πάντα έκανε έναν κύκλο την πίστα πριν πάει να πάρει τον κώνο. Άλλαξα το βραβείο που παίρνει όταν πηγαίνει ευθεία από +150 σε -steps (δηλαδή κάθε step του αλγορίθμου θα μειώνεται το βραβείο που παίρνει άρα πρέπει να μάθει να το λύνει γρήγορα).Έμαθε να το λύνει και βγήκε η παρακάτω γραφική παράσταση. Τα spikes που πηγαίνουν μέχρι και -100.000 είναι επειδή σε μερικές εποχές απλά έκανε βόλτες στην πίστα χωρίς να πάρει τον στόχο και χωρίς να χτυπίσει σε τοίχο άρα μάζεψε τεράστιο αρνητικό βραβείο.

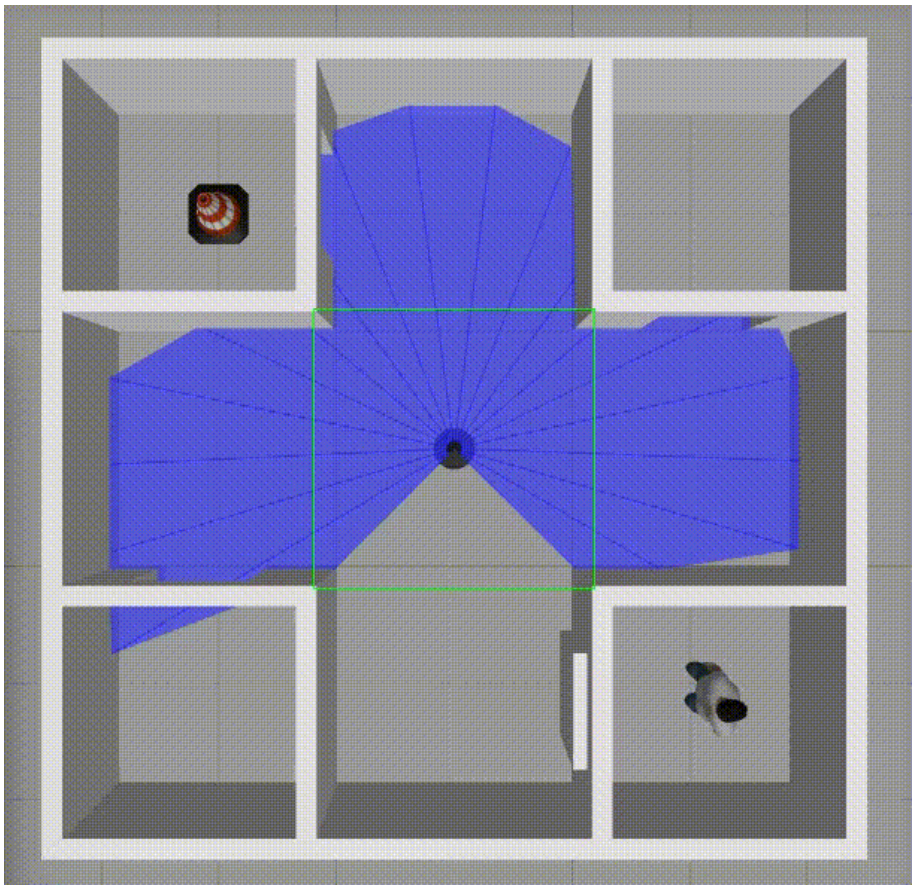


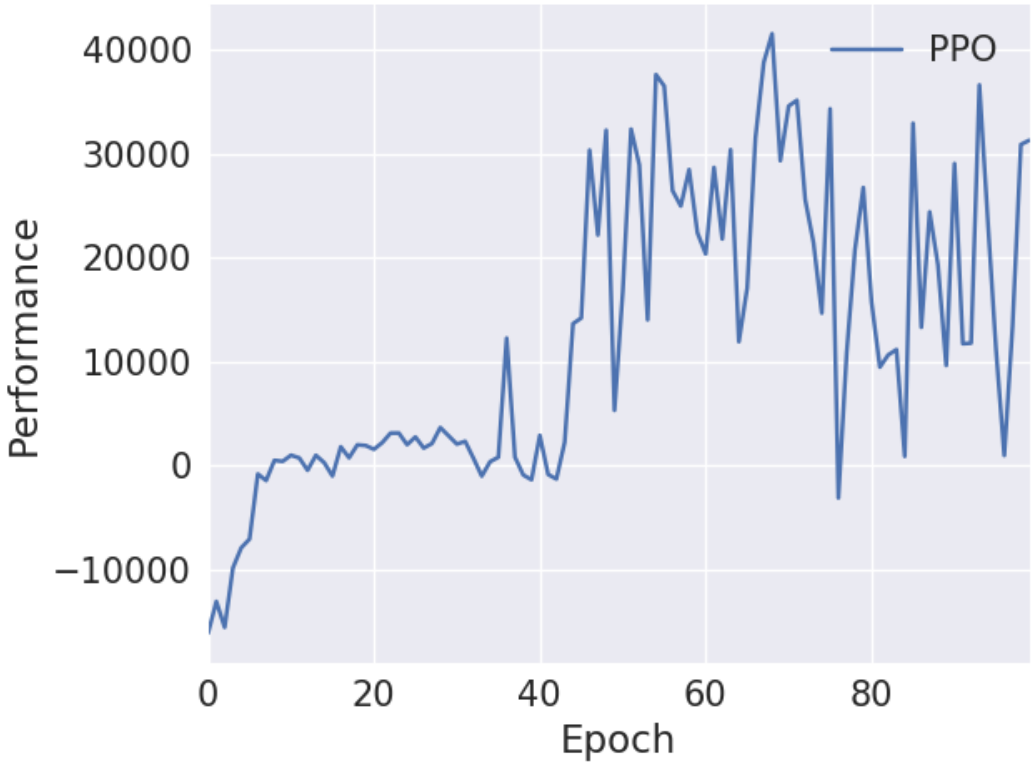
Επίπεδο 2



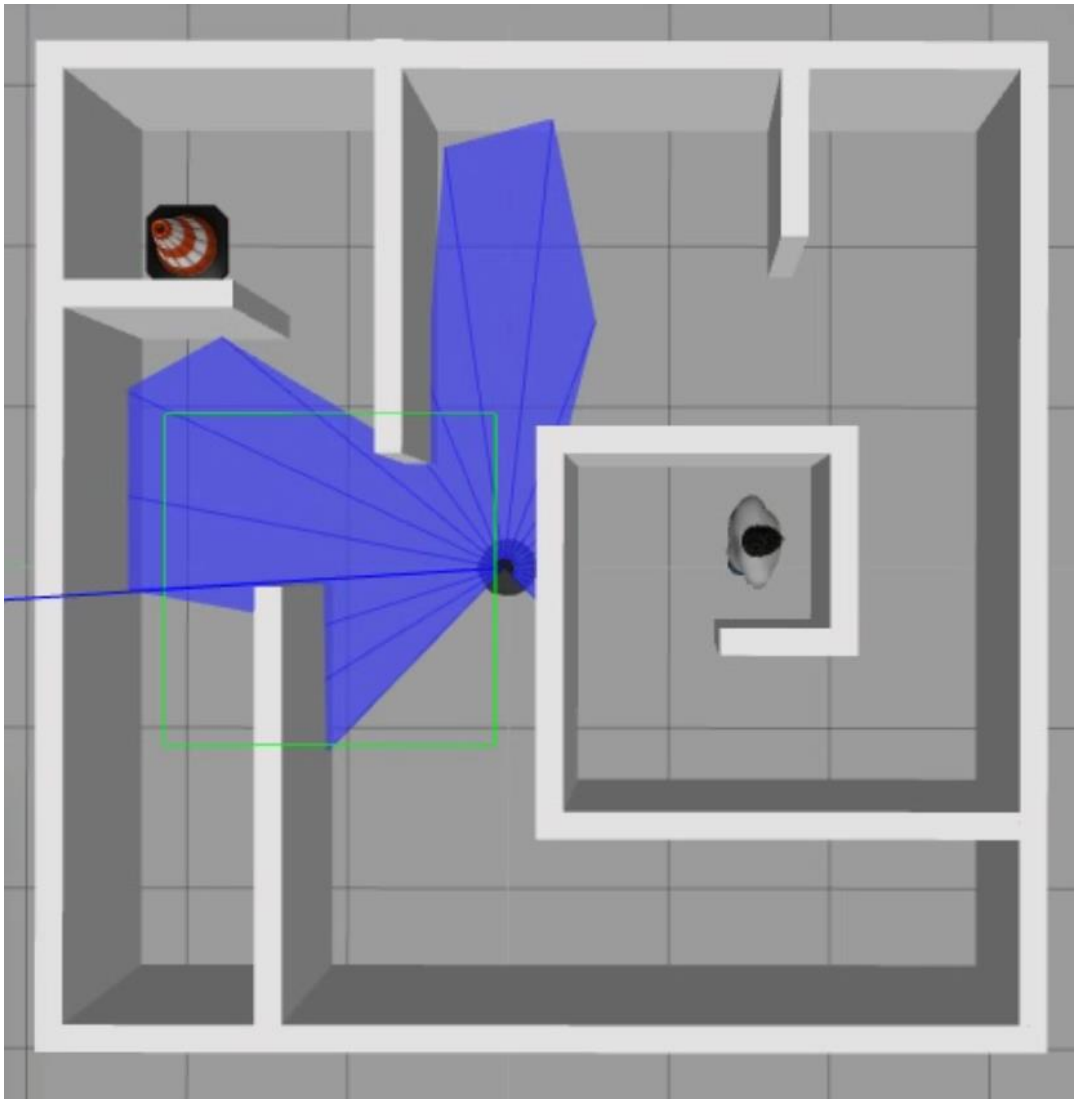

PPO
60000
50000
40000
30000
20000
10000
0
−10000
Performance
0
10
20
30
40
50
60
Epoch

Επίπεδο 3




PPO
40000
30000
20000
10000
0
−10000
Performance
0
20
40
60
80
Epoch

Επίπεδο 4

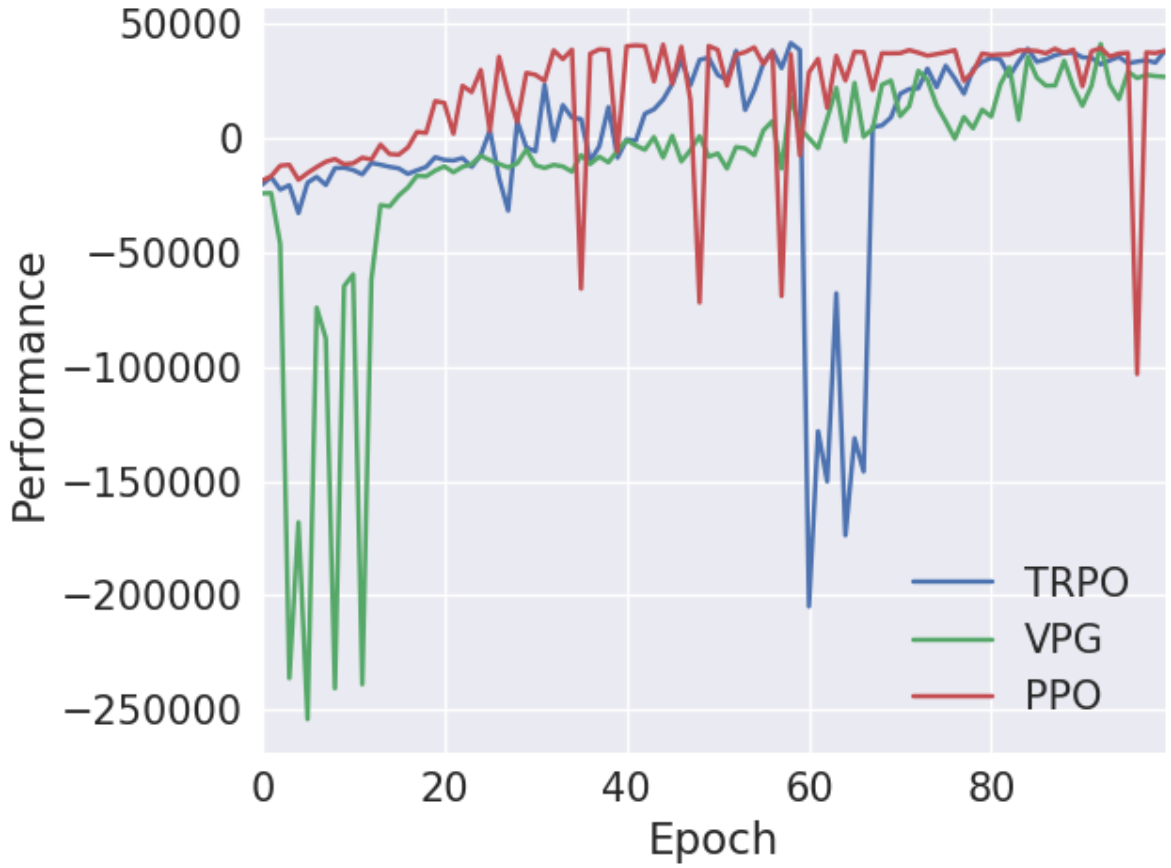


Δοκίμασα και ένα ακόμα πιο περίπλοκο περιβάλλον αλλά ο αλγόριθμος δεν έμαθε να το λύνει.

5.2 Σύγκριση αλγορίθμων

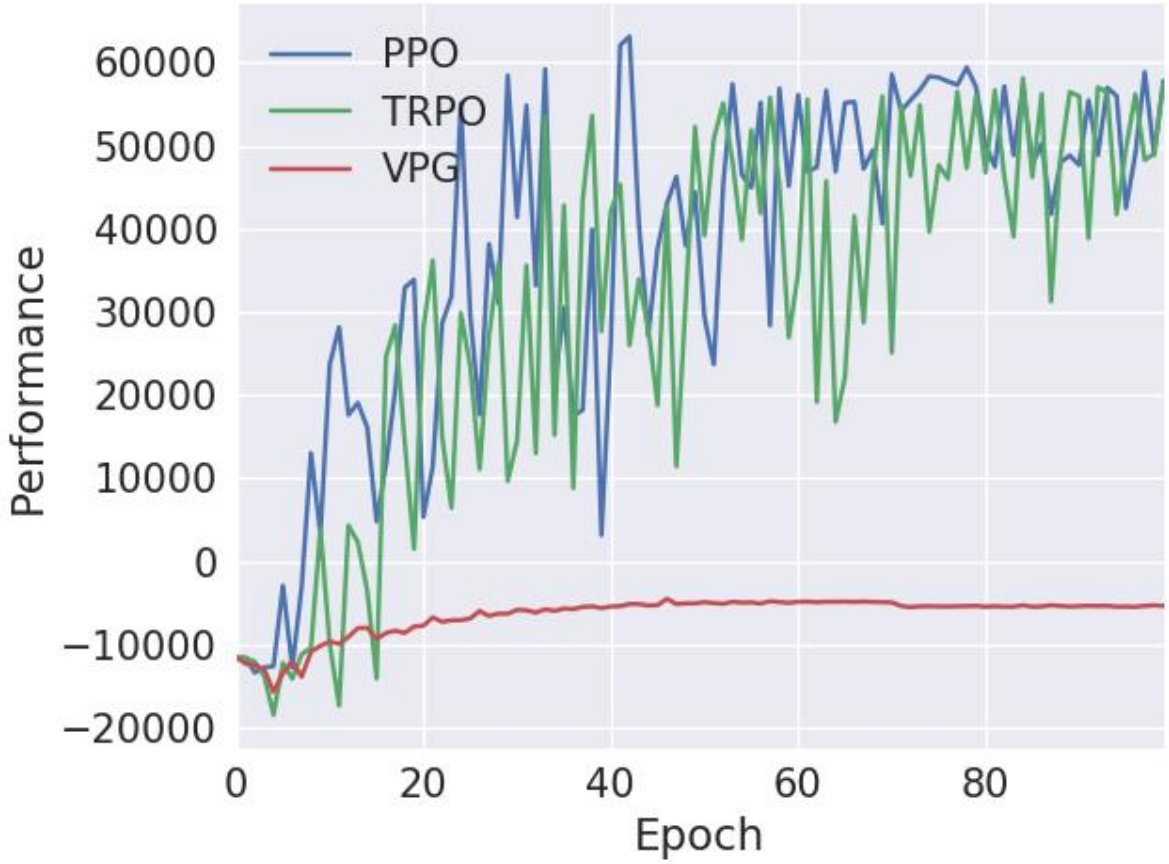
Επίπεδο 1

Βλέπουμε ότι και οι 3 αλγόριθμοι μαθαίνουν να λύνουν το περιβάλλον. Ο PPO φαίνεται να μαθαίνει γρηγορότερα και να έχει μεγαλύτερη σταθερότητα από τους άλλους δύο αλγορίθμους. Ο PPO φτάνει στην λύση κοντά στην εποχή 30 , ο TRPO στην εποχή 50 και ο VPG στην εποχή 80.


50000
0
−50000
−100000
−150000
−200000
−250000
Performance
0
20
40
60
80
Epoch
TRPO
VPG
PPO

Επίπεδο 2

Βλέπουμε 2 από τους 3 αλγόριθμους μαθαίνουν να λύνουν το περιβάλλον. Ο PPO και ο TRPO έχουν παρόμοια αποτελέσματα (με τον PPO να μαθαίνει λίγο πιο γρήγορα και να φτάνει μεγαλύτερο μέγιστο βραβείο). Ο PPO και ο TRPO φτάνουν σε λύση στην εποχή 20 αλλά τα αποτελέσματα σταθεροποιούνται στην εποχή 50. Ο VPG δεν καταφέρνει να βρει ποτέ την λύση (δεν μαθαίνει να βρίσκει ούτε τον πρώτο στόχο).



Επίπεδο 3

Βλέπουμε ότι μόνο ο PPO μαθαίνει να λύνει το περιβάλλον. Ο PPO βρίσκει την λύση κοντά στην εποχή 50. Ο TRPO δεν βρίσκει ποτέ την λύση αλλά βρίσκει τον πρώτο στόχο κοντά στην εποχή 30 .Ο VPG πάλι δεν καταφέρνει να βρει ποτέ την λύση (δεν μαθαίνει να βρίσκει ούτε τον πρώτο στόχο).

5.3 Συμπεράσματα

Βλέπουμε ότι ο πιο αξιόπιστος αλγόριθμος είναι ο PPO που φτάνει πιο συχνά και πιο γρήγορα στην λύση. Ο TRPO είναι λίγο πιο αργή και δεν μπορεί να λύσει το τρίτο επίπεδο. Ο VPG είναι η πιο αργή από τους 3 και μπορεί να λύσει μόνο το απλό περιβάλλον. Όταν το περιβάλλον είναι αρκετά απλό και οι τρεις αλγόριθμοι μπορούν να βρουν την λύση σε σχετικά σύντομο χρόνο. Ακόμα και σε αυτήν την περίπτωση βλέπουμε ότι υπάρχουν διαφορές στο πόσο γρήγορα βρίσκεται η λύση καθώς και το πόσο μεγάλο είναι το μέγιστο βραβείο που φτάνει ο κάθε αλγόριθμος. Καθώς αυξάνεται η πολυπλοκότητα του περιβάλλοντος μειώνεται το ποσοστό επιτυχίας όλων των αλγορίθμων. Στη μέση δυσκολία ο VPG δεν βρίσκει λύση καθόλου αλλά και οι άλλοι δύο που βρίσκουν έχουν χαμηλότερο ποσοστό επιτυχίας από πριν. Στη μεγαλύτερη δυσκολία μόνο ο PPO βρίσκει λύση με ακόμα χαμηλότερο ποσοστό

επιτυχίας (σε σχέση με τα προηγούμενα επίπεδα δυσκολίας) . Καθώς αυξάνεται ακόμα περισσότερο η δυσκολία του περιβάλλοντος οι αλγόριθμοι δεν μπορούν να βρουν λύση και χρειάζονται διαφορετικές τεχνικές αντιμετώπισης του προβλήματος.

6 Συζήτηση

6.1 Δυσκολίες

Κατά τη διάρκεια της εργασίας συνάντησα αρκετές δυσκολίες. Οι περισσότερες από αυτές ήταν στο πως θα μπορέσει ο αλγόριθμος να μάθει να λύνει σωστά τις πίστες που έφτιαξα. Εκτός από αυτό υπήρξαν κάποιες δυσκολίες στην εγκατάσταση και χρήση των εργαλείων που χρησιμοποιήθηκαν.
Για την εγκατάσταση του ROS Kinetic και Gazebo 7 ,που ήταν απαραίτητα για το gym_gazebo_kinetic , χρειαζόταν να βάλω Linux (Ubuntu 16.04). Επέλεξα να κάνω dual boot το λαπτοπ μου γιατί μου φάνηκε καλύτερη εναλλακτική από docker και virtual machine. Κατέληξα εκεί επειδή και οι δύο άλλες επιλογές δεν είχαν καλή συμβατότητα με τα εργαλεία που χρειαζόμουν αλλά και επειδή μου φάνηκε η πιο απλή λύση. Το λαπτοπ μου δεν είχε πολύ ελεύθερο χώρο στον σκληρό δίσκο μετά την εγκατάσταση δεύτερου λογισμικού και παρά τις προσπάθειες μου να αδειάσω χώρο πολύ συχνά τα Linux δεν είχαν ελεύθερο χώρο στο δίσκο και αυτό μερικές φορές οδηγούσε στο πρόγραμμα εκμάθησης να σταματάει στο μέσο της εκπαίδευσης. Κάθε φορά που δεν ολοκληρωνόταν σωστά η εκπαίδευση αποθηκεύονταν το μέχρι τότε μοντέλο αλλά δεν γινόταν να συνεχίσω την εκπαίδευση του. Αυτό είχε αποτέλεσμα πολλές φορές να μην προλαβαίνει ο αλγόριθμος να μάθει αρκετά καλά και να χάνω πολλές ώρες εκπαίδευσης. Οι 100 εποχές που έτρεξα στις περισσότερες πίστες ήθελαν περίπου 5-6 ώρες εκπαίδευσης.
Τα προβλήματα σχετικά με την εκπαίδευση είχαν να κάνουν κυρίως με το ότι κάθε κύκλος εκμάθησης ήθελε τουλάχιστον 2.5 ώρες και μερικές φορές ακόμα και 9 ώρες! Οπότε για κάθε αλλαγή που ήθελα να κάνω στον κώδικα (είτε αλλαγή στην πίστα, αλλαγή στα βραβεία ή αλλαγή στις παραμέτρους εκμάθησης του αλγορίθμου) έπρεπε να περιμένω αρκετές ώρες για να δω αν είχε θετικές επιπτώσεις στην μάθηση. Αυτό είναι από μόνο του αρκετά χρονοβόρο αλλά σε συνδυασμό με το ότι το gazebo σταματούσε να λειτουργεί στη μέση της εκπαίδευσης έκανε την διαδικασία ακόμα πιο αργή και δύσκολη. Δεν μπόρεσα να βρω για ποιο λόγο ακριβώς σταματούσε να λειτουργεί το gazebo αλλά πιθανότητα να έφταιγε η κάρτα γραφικών μου. Κατά την διάρκεια της εργασίας μου άλλαξα αρκετές φορές τις πίστες, τα βραβεία αλλά και τις παραμέτρους των αλγορίθμων μέχρι να βρω έναν ικανοποιητικό συνδυασμό.

6.2 Περαιτέρω έρευνα

Μέσα από αυτή την εργασία προσπάθησα να δείξω πως κάποιοι αλγόριθμοι ενισχυτικής εκμάθησης μπορούν να μάθουν να κάνουν δύσκολες εργασίες όπως ένα ρομπότ να βρίσκει συγκεκριμένα αντικείμενα σε έναν τρισδιάστατο χώρο. Κατάφερα να λύσω κάποια απλά περιβάλλοντα αλλά σε πιο σύνθετες περιπτώσεις , όπως έναν περίπλοκο λαβύρινθο , ο αλγόριθμος δεν μαθαίνει να βρίσκει λύση. Υπάρχουν τρόποι να λυθούν ακόμα πιο δύσκολες πίστες με μεγαλύτερη ταχύτητα και πιο σταθερά

αποτελέσματα. Ένας απλός τρόπος είναι να αφήσουμε τον αλγόριθμο να τρέξει για παραπάνω χρόνο. Επίσης ίσως γίνεται να βρούμε καλύτερα βραβεία και υπερπαραμέτρους ώστε να έχουμε καλύτερα αποτελέσματα από αυτά που βρήκα εγώ.

Βιβλιογραφία

[1] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484-489.

[2] Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., ... & Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science, 362(6419), 1140-1144.

[3] Vinyals, O., Ewalds, T., Bartunov, S., Georgiev, P., Vezhnevets, A. S., Yeo, M., ... & Tsing, R. (2017). Starcraft ii: A new challenge for reinforcement learning. arXiv preprint arXiv:1708.04782.

[4] Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., ... & Zhang, S. (2019). Dota 2 with large scale deep reinforcement learning. arXiv preprint arXiv:1912.06680.

[5] https://gazebosim.org/home

[6] https://www.ros.org

[7] https://spinningup.openai.com/en/latest/spinningup/rl_intro.html

[8] https://www.gymlibrary.dev/api/core/

[9] https://github.com/zhaolongkzz/gym_gazebo_kinetic

[10] Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T. P., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484–489. https://doi.org/10.1038/nature16961

[11] Watkins, C., & Dayan, P. (1992). Q-learning. Machine Learning, 8(3–4), 279–292. https://doi.org/10.1007/bf00992698.

[12] Rummery, G. A., & Niranjan, M. (1994). On-Line Q-Learning using Connectionist systems. ResearchGate. https://www.researchgate.net/publication/2500611_On-Line_Q-Learning_Using_Connectionist_Systems

[13] Mnih, Volodymyr, et al. “Playing Atari with Deep Reinforcement Learning.” *arXiv.Org*, 19 Dec. 2013, arxiv.org/abs/1312.5602.

[14] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

[15] Mnih, V. (2016, February 4). Asynchronous methods for deep reinforcement learning. arXiv.org. https://arxiv.org/abs/1602.01783.

[16] Schulman, J. (2015, February 19). Trust Region Policy optimization. arXiv.org. https://arxiv.org/abs/1502.05477.

[17] Tokic, M. (2010, September). Adaptive ε-greedy exploration in reinforcement learning based on value differences. In Annual Conference on Artificial Intelligence (pp. 203-210). Springer, Berlin, Heidelberg.

[18] Thompson, W. (1933). On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. Biometrika, 25(3/4), 285. https://doi.org/10.2307/2332286

[19] Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3–4), 229–256. https://doi.org/10.1007/bf00992696