Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Πλοήγηση αυτόνομου οχήματος στον χώρο, με χρήση αλγορίθμων βαθιάς ενισχυτικής μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΑΒΑΛΑΣ ΧΡΙΣΤΟΦΟΡΟΣ

Επιβλέπων : Ανδρέας - Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2019

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Πλοήγηση αυτόνομου οχήματος στον χώρο, με χρήση αλγορίθμων βαθιάς ενισχυτικής μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΑΒΑΛΑΣ ΧΡΙΣΤΟΦΟΡΟΣ

Επιβλέπων : Ανδρέας - Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 30η Οκτωβρίου 2019.

Ανδρέας - Γεώργιος Σταφυλοπάτης Καθηγητής Ε.Μ.Π.	Γεώργιος Στάμου Καθηγητής Ε.Μ.Π.	Παναγιώτης Τσανάκας Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2019

Γαβαλάς Χριστόφορος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Το επιστημονικό πεδίο της Ενισχυτικής Μάθησης έχει πετύχει αξιοσημείωτα αποτελέσματα σε πολλούς διαφορετικούς κλάδους (Βιοϊατρική, επιχειρηματικότητα, όραση υπολογιστών, παιχνίδια κ.α.). Συγκεκριμένα στον χώρο της ρομποτικής, τα τελευταία χρόνια, έχει γίνει σημαντική πρόοδος στην εφαρμογή αλγορίθμων ενισχυτικής μάθησης και έχουν εξαχθεί εξαιρετικά αποτελέσματα. Στα πλαίσια της παρούσας εργασίας, μελετάται η πλοήγηση αυτόνομου οχήματος στον χώρο με χρήση αλγορίθμων βαθιάς ενισχυτικής μάθησης. Το πρόβλημα που αντιμετωπίστηκε, αφορά κίνηση σε συνεχή χώρο με το όχημα να λαμβάνει μετρήσεις από το περιβάλλον και να διαμορφώνει σύμφωνα με αυτές την ταχύτητά του. Συγκεκριμένα, σχεδιάστηκαν τρία μοντέλα ενισχυτικής μάθησης (DQN, REINFORCE, A2C) και συγκρίθηκαν τα αποτελέσματα που έδωσαν σε διάφορες συνθήκες περιβάλλοντος κατά την πλοήγηση τους σε αυτό. Κυρίως, μελετήθηκε η προσέγγιση κάποιου στόχου από το όχημα, σε περιβάλλοντα με ή χωρίς ύπαρξη εμποδίων.

Λέξεις κλειδιά

Τεχνητή Νοημοσύνη, Ενισχυτική Μάθηση, Βαθιά Ενισχυτική Μάθηση, Αυτόνομη Πλοήγηση, Νευρωνικά Δίκτυα, DQN, REINFORCE, Δράστης - Κριτής, A2C.

Abstract

The field of Reinforcement Learning has achieved remarkable results in many different fields such as bio medicine, business, computer vision, games, etc. Particularly in the field of robotics there has been made significant progress and the obtained results are remarkable. In this thesis, we study autonomous vehicle navigation using deep reinforcement learning algorithms. The problem we encountered, relates to navigating in a continuous space area with the vehicle receiving measurements from the environment and adjusting its speed accordingly. Specifically, we designed three reinforcement learning models (DQN, REINFORCE, A2C) and compared their performance in various environments. In particular, the task of the autonomous vehicle is to approach a specific goal point in maps with or without obstacles.

Key words

Artificial Intelligence, Reinforcement Learning, Deep Reinforcement Learning, Autonomous Navigation, Neural Networks, DQN, REINFORCE, Actor - Critic, A2C

Ευχαριστίες

Ευχαριστώ θερμά τον επιβλέποντα καθηγητή αυτής της διατριβής, κ.Ανδρέα - Γεώργιο Σταφυλοπάτη για την εμπιστοσύνη του. Ευχαριστώ επίσης τους κ. Τάσο Παπαγιάννη και κ. Θάνο Τασάκο για την πρόθυμη και πάντα αποτελεσματική βοήθειά τους και την εξαιρετική συνεργασία που είχαμε. Θέλω επιπλέον να ευχαριστήσω την οικογένειά μου για την υποστήριξη που μου έδειξαν καθ' όλη τη διάρκεια των σπουδών μου. Τέλος θέλω να ευχαριστήσω τους φίλους μου, οι οποίοι με βοήθησαν, ο καθένας με τον τρόπο του καθ' όλη τη διάρκεια των φοιτητικών μου χρόνων.

Γαβαλάς Χριστόφορος,

Αθήνα, 30η Οκτωβρίου 2019

Περιεχόμενα

Περίληψη 7

Abstract 9

Ευχαριστίες 11

Περιεχόμενα 13

Κατάλογος πινάκων 15

Κατάλογος σχημάτων 17

1. Εισαγωγή 19
1.1 Αυτόνομη Πλοήγηση 19
1.2 Αντικείμενο της διπλωματικής 19
1.3 Οργάνωση κειμένου 20

2. Θεωρητικό υπόβαθρο 21
2.1 Μηχανική μάθηση 21
2.1.1 Τι είναι η Μηχανική μάθηση; 21
2.1.2 Χρησιμότητα Μηχανικής μάθησης 21
2.1.3 Μέθοδοι Μηχανικής μάθησης 21
2.1.4 Βαθιά Μάθηση 22
2.2 Νευρωνικά Δίκτυα 22
2.2.1 Τεχνητός Νευρώνας 22
2.2.2 Τεχνητά νευρωνικά δίκτυα 23
2.2.3 Συνάρτηση ενεργοποίησης 24
2.2.4 Συνάρτηση Απώλειας 24
2.2.5 Οπίσθια Διάδοση 25
2.2.6 Βελτιστοποίηση 26
2.3 Ενισχυτική μάθηση 26
2.3.1 Εισαγωγή 26
2.3.2 Μοντελοποίηση του προβλήματος 27
2.3.3 Μαρκοβιανές διαδικασίες αποφάσεων 29
2.3.4 Σχεδίαση με Δυναμικό Προγραμματισμό 31
2.3.5 Πρόβλεψη άνευ Μοντέλου 33
2.3.5.1 Αξιολόγηση πολιτικής Monte Carlo 33
2.3.5.2 Αξιολόγηση πολιτικής Χρονικών Διαφορών TD(0) 33
2.3.5.3 Ίχνη Επιλεξιμότητας 34
2.3.5.4 Αξιολόγηση πολιτικής Χρονικών Διαφορών TD(λ) 35
2.3.6 Έλεγχος άνευ μοντέλου 36
2.3.6.1 Έλεγχος Monte Carlo 36
2.3.6.2 Έλεγχος Χρονικών Διαφορών TD 38

3. Βαθιά Ενισχυτική Μάθηση . . . 41
3.1 Μηχανισμοί προσέγγισης συναρτήσεων τιμών . . . 41
3.2 Μέθοδος DQN . . . 42
3.3 Αναζήτηση πολιτικής . . . 44
3.4 Policy Gradients . . . 44
3.5 Μέθοδος REINFORCE . . . 45
3.6 Μέθοδοι Δράστη - Κριτή . . . 46

4. Υλοποίηση και Αξιολόγηση Πειραμάτων . . . 49
4.1 Εισαγωγή . . . 49
4.2 Μοντελοποίηση του Προβλήματος . . . 49
4.2.1 Αυτόνομη Πλοήγηση . . . 49
4.2.2 Προσομοίωση του Περιβάλλοντος . . . 50
4.2.3 Μοντελοποίηση των Ανταμοιβών . . . 51
4.2.4 Σχεδίαση των Πρακτόρων . . . 52
4.2.4.1 Πράκτορας DQN . . . 52
4.2.4.2 Πράκτορας REINFORCE . . . 52
4.2.4.3 Πράκτορας A2C . . . 53
4.3 Πειραματικά Αποτελέσματα . . . 53
4.3.1 Προσέγγιση των Πειραμάτων . . . 53
4.3.2 Πράκτορας DQN . . . 53
4.3.2.1 Σταθερή θέση Εκκίνησης και Στόχου . . . 54
4.3.2.2 Τυχαία θέση Εκκίνησης και Στόχου . . . 56
4.3.3 Πράκτορας Policy Gradient . . . 57
4.3.3.1 Σταθερή θέση Εκκίνησης και Στόχου . . . 58
4.3.3.2 Τυχαία θέση Εκκίνησης και Στόχου . . . 61
4.3.4 Πράκτορας Δράστη - Κριτή . . . 62

5. Συμπεράσματα . . . 69
5.1 Απόδοση Αλγορίθμων . . . 69
5.2 Μοντελοποίηση πρακτόρων με ταχύτητα Τύπου 2 . . . 71
5.3 Η αποτυχία υλοποίησης της αποφυγής εμποδίων . . . 72
5.4 Κατεύθυνση Μελλοντικής Έρευνας . . . 73

Κατάλογος πινάκων

Πίνακας 1: Σύνολο των πράξεων του πράκτορα για τις δύο μοντελοποιήσεις ταχυτήτων. . . 52
Πίνακας 2: Παράμετροι των πειραμάτων 54
Πίνακας 3: Παράμετροι Πειράματος 1 54
Πίνακας 4: Παράμετροι Πειράματος 2 55
Πίνακας 5: Παράμετροι Πειράματος 3 56
Πίνακας 6: Παράμετροι Πειράματος 4 57
Πίνακας 7: Παράμετροι Πειράματος 5 58
Πίνακας 8: Παράμετροι Πειράματος 6 59
Πίνακας 9: Παράμετροι Πειράματος 7 60
Πίνακας 10Παράμετροι Πειράματος 8 61
Πίνακας 11Παράμετροι Πειράματος 9 62
Πίνακας 12Παράμετροι Πειράματος 10 63
Πίνακας 13Παράμετροι Πειράματος 11 64
Πίνακας 14Παράμετροι Πειράματος 12 65
Πίνακας 15Παράμετροι Πειραμάτων A2C 66

Κατάλογος σχημάτων

Σχήμα 1: Δομή τεχνητού νευρώνα . . . 22
Σχήμα 2: Feed Forward Neural Network . . . 23
Σχήμα 3: Σιγμοειδής συνάρτηση . . . 24
Σχήμα 4: Υπερβολική Εφαπτομένη συνάρτηση . . . 25
Σχήμα 5: Rectified Linear Unit (ReLU) . . . 25
Σχήμα 6: Διαφορετικά επιστημονικά πεδία που συνθέτουν την Ενισχυτική Μάθηση . . . 27
Σχήμα 7: Αλληλεπίδραση Πράκτορα - Περιβάλλοντος . . . 28
Σχήμα 8: Αλγόριθμος αξιολόγησης πολιτικής Monte Carlo . . . 34
Σχήμα 9: Αλγόριθμος αξιολόγησης πολιτικής Χρονικών Διαφορών TD(0) . . . 34
Σχήμα 10: Συνάρτηση Βάρους για την μέθοδο TD(λ) . . . 35
Σχήμα 11: Ίχνη επιλεξιμότητας . . . 36
Σχήμα 12: Αλγόριθμος αξιολόγησης πολιτικής Χρονικών Διαφορών TD(λ) . . . 36
: (a) . . . 37
: (b) . . . 37
Σχήμα 14: Σχηματική αναπαράσταση της γενικευμένης επανάληψης με βάση την πολιτική (GPI), με δύο διαφορετικούς τρόπους . . . 37
Σχήμα 15: On-policy αλγόριθμος ελέγχου πολιτικής Monte Carlo . . . 38
Σχήμα 16: Off-policy αλγόριθμος ελέγχου πολιτικής Monte Carlo . . . 38
: (a) SARSA . . . 40
: (b) SARSA-λ . . . 40
Σχήμα 18: Αλγόριθμοι on-policy ελέγχου χρονικών διαφορών . . . 40
Σχήμα 19: Off-policy αλγόριθμος ελέγχου πολιτικής Q learning . . . 40

Σχήμα 20: Ενοποιημένη επισκόπηση των μεθόδων Ενισχυτικής Μάθησης . . . 41
Σχήμα 21: Αλγόριθμος DQN . . . 43
Σχήμα 22: Αλγόριθμος REINFORCE . . . 46
Σχήμα 23: Δομή Αλγορίθμου A2C . . . 47

Σχήμα 24: Πείραμα 1:Τροχιά αυτόνομου οχήματος . . . 55
Σχήμα 25: Πείραμα 2:Τροχιά αυτόνομου οχήματος . . . 56
Σχήμα 26: Πείραμα 3:Τροχιά αυτόνομου οχήματος . . . 57
Σχήμα 27: Πείραμα 4:Τροχιά αυτόνομου οχήματος . . . 58
Σχήμα 28: Πείραμα 5:Επιτυχημένες τροχιές αυτόνομου οχήματος . . . 59
Σχήμα 29: Πείραμα 5:Αποτυχημένες κυκλικές τροχιές αυτόνομου οχήματος . . . 60
Σχήμα 30: Πείραμα 7:Τροχιά αυτόνομου οχήματος . . . 61
Σχήμα 31: Πείραμα 8:Τροχιά αυτόνομου οχήματος . . . 62
Σχήμα 32: Πείραμα 9:Τροχιά αυτόνομου οχήματος . . . 63
Σχήμα 33: Πείραμα 10:Τροχιά αυτόνομου οχήματος . . . 64
Σχήμα 34: Πείραμα 11:Τροχιές αυτόνομου οχήματος . . . 65
Σχήμα 35: Πείραμα 12:Τροχιές αυτόνομου οχήματος . . . 66
Σχήμα 36: Πείραμα 13:Τροχιές αυτόνομου οχήματος . . . 66
Σχήμα 37: Πείραμα 14:Τροχιές αυτόνομου οχήματος . . . 67
Σχήμα 38: Πείραμα 14:Τροχιές αυτόνομου οχήματος . . . 67

Σχήμα 39: Στατιστικά απόδοσης των μοντέλων 70
Σχήμα 40: Γραφικές Παραστάσεις Αξίας - Επεισοδίου 71
Σχήμα 41: Τροχιά πράκτορα DQN, με ύπαρξη εμποδίου 72

Κεφάλαιο 1

Εισαγωγή

1.1 Αυτόνομη Πλοήγηση

Ο όρος αυτόνομη πλοήγηση σημαίνει ότι ένα όχημα είναι σε θέση να σχεδιάσει την πορεία του και να εκτελέσει μια συγκεκριμένη τροχιά χωρίς ανθρώπινη παρέμβαση. Σε μερικές περιπτώσεις, χρησιμοποιούνται στη διαδικασία σχεδιασμού απομακρυσμένα βοηθήματα πλοήγησης, ενώ άλλες φορές οι μόνες διαθέσιμες πληροφορίες για τον υπολογισμό μιας διαδρομής βασίζονται σε εισροές αισθητήρων στο ίδιο το όχημα. Ένα αυτόνομο ρομπότ είναι αυτό που όχι μόνο μπορεί να διατηρήσει τη δική του σταθερότητα καθώς κινείται αλλά και μπορεί να προγραμματίσει τις κινήσεις του. Τα αυτόνομα ρομπότ χρησιμοποιούν βοηθήματα πλοήγησης όταν είναι δυνατόν, αλλά μπορούν επίσης να βασίζονται σε οπτικά, ακουστικά και οσφρητικά σημάδια. Μόλις συγκεντρωθούν βασικές πληροφορίες θέσης, ο αλγόριθμος του οχήματος πρέπει να εφαρμοστεί για να μεταφράσει κάποια βασικά κίνητρα (λόγος για να εγκαταλείψει την παρούσα θέση) σε ένα σχέδιο διαδρομής και κίνησης. Η σχεδίαση της τροχιάς ίσως χρειαστεί να ικανοποιήσει τις εκτιμώμενες ή ανακοινωθέντες προθέσεις άλλων αυτόνομων ρομπότ, προκειμένου να αποφευχθούν οι συγκρούσεις, λαμβάνοντας υπόψη τη δυναμική του περιβλήματος κίνησης του ρομπότ.[1]

Έχουν γίνει προσπάθειες επίλυσης προβλημάτων αυτόνομης πλοήγησης από την σκοπιά πολλών διαφορετικών επιστημονικών πεδίων και τεχνικών. Μια κλασική μέθοδος είναι η τεχνική *'Ταυτόχρονου εντοπισμού θέσης και χαρτογράφησης'*(Simultaneous Localization and Mapping)[2]. Από την οπτική της τεχνητής νοημοσύνης, έχουν εκπονηθεί πολλές μοντελοποίησεις και πειράματα, σε ανάλογα προβλήματα. Η χρήση τεχνητών νευρωνικών δικτύων έχει δώσει αξιόλογα αποτελέσματα στην πλοήγηση στον χώρο [3],[4] και στην αποφυγή εμποδίων [5],[6],[7]. Μια ακόμα προσέγγιση επίλυσης προβλημάτων αυτόνομης πλοήγησης είναι η εφαρμογή αλγορίθμων ενισχυτικής μάθησης. Συγκεκριμένα, έχει γίνει μελέτη για αποφυγή εμποδίων [8], πλοήγηση στον χώρο [9], αντιμετώπιση δυναμικών εμποδίων [10],[11] και χρήση δεδομένων εικόνας για την επίτευξη του επιθυμητού στόχου [12]. Στην παρούσα εργασία θα χρησιμοποιηθούν μοντέλα Βαθιάς Ενισχυτικής Μάθησης για την επίλυση του προβλήματος. Σε αυτόν τον τομέα έχουν επιτευχθεί αξιοσημείωτα αποτελέσματα. Κάποιες από τις βασικές δουλείες που έχουν πραγματοποιηθεί, αφορούν την πλοήγηση σε άγνωστο χάρτη [13], χρήση δεδομένων εικόνας και βίντεο ως δεδομένων εισόδου για την επίλυση του προβλήματος [14],[15],[16],[17], καθώς και επίλυση προβλημάτων πολυπρακτορικών συστημάτων [18]. Τέλος, αλγόριθμοι βαθιάς ενισχυτικής μάθησης έχουν χρησιμοποιηθεί σε προβλήματα εξερεύνησης και χαρτογράφησης του περιβάλλοντος [19],[20],[21].

1.2 Αντικείμενο της διπλωματικής

Στην παρούσα διπλωματική θα εφαρμοστούν αλγόριθμοι βαθιάς ενισχυτικής μάθησης σε προβλήματα πλοήγησης, με την λογική *'από άκρη σε άκρη'*. Με τον όρο αυτόν, εννοούμε πως το μοντέλο, λαμβάνει όλα τα δεδομένα εισόδου και αποφασίζει μια δράση εξ' ολοκλήρου, η οποία συνδυάζει όλες τις παραμέτρους εσωτερικά. Η προσέγγιση αυτή επιλέχθηκε καθώς η προσθήκη νέων περιορισμών και παραμέτρων σε κάθε πείραμα, δεν αλλάζει την φύση της μοντελοποίησης του προβλήματος. Αντίθετα προσθέτοντας απλά τα νέα δεδομένα στην είσοδο και αλλάζοντας ορισμένες παραμέτρους

του μοντέλου, είναι σε θέση να ανταποκριθεί με επιτυχία στις νέες προδιαγραφές και δυσκολίες του προβλήματος.

Στόχος, λοιπόν, της συγκεκριμένης εργασίας, αφορά στην αντιμετώπιση προβλημάτων αυτόνομης πλοήγησης, εκκινώντας από πολύ απλούς στόχους και προσθέτοντας συνεχώς βαθμούς δυσκολίας και πολυπλοκότητας. Με αυτόν τον τρόπο αξιοποιούμε την τεχνική *'από άκρη σε άκρη'* που αναφέραμε. Για την εκπαίδευση των οχημάτων μοντελοποιήθηκαν τρεις διαφορετικές τεχνικές βαθιάς ενισχυτικής μάθησης, (DQN, REINFORCE και A2C) και αξιολογήθηκαν οι αποδόσεις τους κατά την πλοήγηση τους στον χώρο σε μια σειρά προκλήσεων. Επιπλέον, εκτός από τους αλγορίθμους, μοντελοποιήθηκε και το περιβάλλον στο οποίο δρα το αυτόνομο όχημα.

1.3 Οργάνωση κειμένου

Η εργασία χωρίζεται σε 5 κεφάλαια. Στο Κεφάλαιο 1(Εισαγωγή) γίνεται αναφορά στις δουλείες που έχουν επιτευχθεί στα προβλήματα αυτόνομης πλοήγησης, από την σκοπιά της τεχνητής νοημοσύνης και στην συνέχεια η ιδέα της εργασίας. Στο Κεφάλαιο 2(Θεωρητικό υπόβαθρο) αναλύονται τα επιστημονικά πεδία της Μηχανικής μάθησης, συγκεκριμένα της Ενισχυτικής μάθησης και των νευρωνικών δικτύων. Στο Κεφάλαιο 3(Βαθιά Ενισχυτική Μάθηση) επεξηγούνται οι αλγόριθμοι Βαθιάς Ενισχυτικής Μάθησης που υλοποιήθηκαν μαζί με το μαθηματικό τους υπόβαθρο. Στην συνέχεια (Κεφάλαιο 4) παρατίθενται η σχεδίαση του περιβάλλοντος και των αλγορίθμων που χρησιμοποιήθηκαν καθώς και τα αποτελέσματα των επιτυχημένων πειραμάτων. Τέλος, στο Κεφάλαιο 5 (Συμπεράσματα), ερμηνεύονται τα αποτελέσματα των πειραμάτων, γίνεται σύγκριση μεταξύ των πειραμάτων και παρουσιάζεται ο τρόπος με τον οποίο θα μπορούσε να κατευθυνθεί η μελλοντική έρευνα μέσω της συγκεκριμένης διπλωματικής.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

2.1 Μηχανική μάθηση

2.1.1 Τι είναι η Μηχανική μάθηση;

Η Μηχανική μάθηση (Machine learning) αφορά την επιστημονική μελέτη των αλγορίθμων και των στατιστικών μοντέλων, τα οποία χρησιμοποιούνται από κάποιο υπολογιστικό σύστημα, ώστε να εκτελέσουν ένα συγκεκριμένο έργο χωρίς την χρήση προκαθορισμένων οδηγιών, αλλά βασιζόμενα σε μοτίβα και τεκμήρια.[22] Με απλά λόγια, είναι η προσπάθεια να προγραμματίσουμε υπολογιστές να μαθαίνουν από εμπειρίες (δεδομένα) και να παράγουν εξειδικευμένα συμπεράσματα και γνώσεις από αυτές.[23]

2.1.2 Χρησιμότητα Μηχανικής μάθησης

Υπάρχουν αρκετοί λόγοι για τους οποίους η μηχανική μάθηση είναι σημαντική. Μερικοί από αυτούς είναι οι εξής:

- Μερικά προβλήματα δεν μπορούν να οριστούν καλά, παρά μόνο με παραδείγματα. Δηλαδή, σε ορισμένες περιπτώσεις είναι εύκολο να προσδιοριστούν ζεύγη εισόδου/εξόδου, αλλά όχι ένας συνοπτικός τρόπος συσχέτισης αυτών. Σε αυτές τις περιπτώσεις αλγόριθμοι μηχανικής μάθησης είναι ικανοί να προσαρμόσουν την εσωτερική τους δομή, για να παράξουν σωστά αποτελέσματα για μεγάλο αριθμό δεδομένων.[24]

- Είναι πιθανό, ανάμεσα σε μεγάλο όγκο δεδομένων να υπάρχει συσχέτιση, την οποία αλγόριθμοι μηχανικής μάθησης να μπορούν εύκολα να διακρίνουν. (Εξόρυξη δεδομένων)[24]

- Εργαλεία μηχανικής μάθησης, είναι εξαιρετικά χρήσιμα, όταν το περιβάλλον στο οποίο θα χρησιμοποιηθούν έχει χαρακτηριστικά και παραμέτρους άγνωστες στο σχεδιαστή. Σε αυτή την περίπτωση, αλγόριθμοι μηχανικής μάθησης ανταποκρίνονται καλύτερα από κλασικές μεθόδους επίλυσης του προβλήματος.[24]

- Τα περιβάλλοντα συνήθως είναι χρονικά μεταβαλλόμενα. Σε αυτή την περίπτωση υπολογιστικά συστήματα μηχανικής μάθησης που προσαρμόζονται στα δεδομένα εισόδου, προσφέρουν μια καλή λύση, σε αντίθεση με προγραμματισμένα εργαλεία, που υστερούν ως προς την προσαρμοστικότητα.[24]

2.1.3 Μέθοδοι Μηχανικής μάθησης

Το επιστημονικό πεδίο της μηχανικής μάθησης, έχει κατηγοριοποιηθεί σε αρκετούς κλάδους διαφορετικών τύπων εκμάθησης. Στο συγκεκριμένο κεφάλαιο θα γίνει συσχέτιση ως προς την αλληλεπίδραση του συστήματος με το περιβάλλον του, κατά την διάρκεια της εκπαίδευσης.

Επιβλεπόμενη μάθηση

Η Επιβλεπόμενη μάθηση αφορά την εκμάθηση ενός μοντέλου, το οποίο στοχεύει στην προσέγγιση μιας συνάρτησης, η οποία να συνδέει την είσοδο τού με μια έξοδο, χρησιμοποιώντας ζεύγη εισόδου-εξόδου ως ορθά παραδείγματα.[25] Πιο συγκεκριμένα, κάθε παράδειγμα μάθησης, είναι ένα ζευγάρι ενός αντικειμένου εισόδου και μίας επιθυμητής εξόδου.

Μη - Επιβλεπόμενη μάθηση

Στην μη επιβλεπόμενη μάθηση, το μοντέλο μαθαίνει μοτίβα των δεδομένων εισόδου, παρά το γεγονός ότι δεν υπάρχει αναπληροφοριοδότηση.[25] Η πιο κοινή λειτουργία της μη επιβλεπόμενης μάθησης, είναι η ομαδοποίηση δεδομένων. (clustering)

Ενισχυτική μάθηση

Στην ενισχυτική μάθηση, η εκπαίδευση του μοντέλου, γίνεται μέσα από ”επιβραβεύσεις” ή ”τιμωρίες” που καθορίζονται από το περιβάλλον, κατά την διάρκεια της αλληλεπίδρασής του συστήματος με αυτό.[25]

2.1.4 Βαθιά Μάθηση

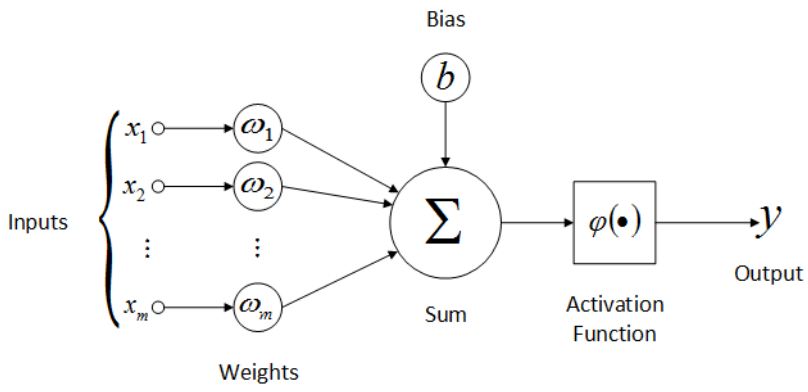
Η βαθιά μάθηση είναι ένα πεδίο της τεχνητής νοημοσύνης που μιμείται τη λειτουργία του ανθρώπινου εγκεφάλου στην επεξεργασία δεδομένων και στη δημιουργία μοτίβων, στη λήψη αποφάσεων. Αποτελεί ένα υποσύνολο της μηχανικής μάθησης που περιλαμβάνει δίκτυα ικανά να μαθαίνουν χωρίς επίβλεψη από δεδομένα που είναι αδόμητα ή μη επισημασμένα.[26]

Με την βοήθεια της βαθιάς μάθησης, καθίσταται εφικτή η μοντελοποίηση δεδομένων με πολύπλοκες αρχιτεκτονικές, συνδυάζοντας διάφορους μη γραμμικούς μετασχηματισμούς. Όραση υπολογιστών, Αναγνώριση φωνής και εικόνας, Επεξεργασία φυσικής γλώσσας, Βιοπληροφορικής είναι ορισμένα από τα πολλά πεδία εφαρμογής της.[27][28][26]

2.2 Νευρωνικά Δίκτυα

2.2.1 Τεχνητός Νευρώνας

Σε ένα νευρωνικό δίκτυο, ο κάθε νευρώνας συνθέτει την βασική υπολογιστική μονάδα του συστήματος. Η δομή του, αποτελεί μια απλοποιημένη μαθηματική προσέγγιση του βιολογικού νευρώνα του νευρικού συστήματος του ανθρώπου. Όπως φαίνεται και στο Σχήμα 1 ένας νευρώνας απαρτίζεται από έναν γραμμικό συνδυασμό των εισόδων του, ο οποίος στην συνέχεια περνά από μια συνάρτηση (συνήθως μη γραμμική). Αναλυτικότερα, ο αθροιστής υπολογίζει τον γραμμικό συνδυασμό της εισόδου $\mathbf{x_i}$ με τα βάρη $\mathbf{w_i}$ του νευρώνα και προσθέτει και έναν σταθερό όρο $\mathbf{b}$. Εν συνεχεία, το αποτέλεσμα του αθροιστή περνά ως είσοδος στον μη γραμμικό όρο $\boldsymbol{\varphi}()$, που ονομάζεται συνάρτηση ενεργοποίησης.



Σχήμα 1: Δομή τεχνητού νευρώνα.[29]

Εάν η συνάρτηση ενεργοποίησης ισούται με $+\mathbf{1}$ όταν η είσοδος είναι θετική και με $-\mathbf{1}$ όταν η είσοδος είναι αρνητική, τότε ο νευρώνας αυτός αναφέρεται ως perceptron. Από το μοντέλο φαίνεται πως η είσοδος της συνάρτησης ενεργοποίησης του νευρώνα είναι

$$\upsilon = \sum_{i=1}^{m} x_i w_i + b \tag{2.1}$$

Ο στόχος του perceptron είναι να ταξινομήσει ορθά, ομάδες δεδομένων $[\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_m}]$ σε δύο διαφορετικές κλάσεις. Αυτό επιτυγχάνεται με την δημιουργία ενός υπερεπιπέδου το οποίο θα χωρίζει τον χώρο σε δύο υποχώρους. Το υπερεπίπεδο αυτό ορίζεται ως

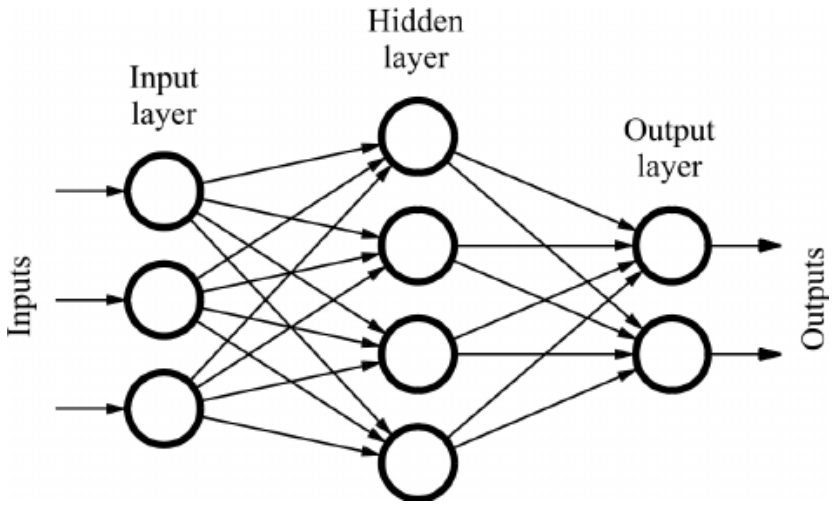
$$\sum_{i=1}^{m} x_i w_i + b = 0 \tag{2.2}$$

Συνεπώς, στόχος του perceptron είναι να προσαρμόσει τα βάρη του $[\mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_m}]$ για την δημιουργία ενός υπερεπιπέδου που να ομαδοποιεί σωστά τα δεδομένα σε δύο κατηγορίες. Για να συγκλίνει το perceptron σε κάποιο υπερεπίπεδο, αναγκαία προϋπόθεση είναι τα δεδομένα να είναι γραμμικώς διαχωρίσιμα. [30]

2.2.2 Τεχνητά νευρωνικά δίκτυα

Στο προηγούμενο κεφάλαιο μελετήθηκε πλήρως η δομική μονάδα ενός νευρωνικού δικτύου, δηλαδή ο νευρώνας. Για να λυθούν όμως προβλήματα με μεγάλη πολυπλοκότητα, είναι απαραίτητο να συνδυαστούν πολλοί διαφορετικοί νευρώνες μεταξύ τους, με κάποια συγκεκριμένα αρχιτεκτονική. Η αρχιτεκτονική αυτή συνθέτει ένα τεχνητό νευρωνικό δίκτυο. Υπάρχουν πολλά διαφορετικά είδη νευρωνικών δικτύων, όπως τα Convolutional neural networks, Hopfield networks, Generative adversarial network κ.α. Στο συγκεκριμένο κεφάλαιο θα μελετηθεί η δομή και η λειτουργία του Feed Forward Neural Network.

Το συγκεκριμένο νευρωνικό δίκτυο είναι χωρισμένο σε επίπεδα, οπού το κάθε επίπεδο αποτελείται απο έναν ορισμένο αριθμό νευρώνων. Η ροή επεξεργασίας της πληροφορίας, γίνεται σειριακά ανά επίπεδο από την αρχή προς το τέλος. Η δομή ενός Feed Forward Neural Network φαίνεται αναλυτικά και στο Σχήμα 2. Κάθε νευρωνικό δίκτυο, εμπεριέχει τις εξής κατηγορίες επιπέδων:



Σχήμα 2: Feed Forward Neural Network

- **Επίπεδο Εισόδου.** Αυτό το επίπεδο, δέχεται τα δεδομένα εισόδου. Παρέχει δηλαδή πληροφορία από το περιβάλλον στο δίκτυο, χωρίς περαιτέρω επεξεργασία. Ο νευρώνες αυτοί, δηλαδή, μεταβιβάζουν απλώς την πληροφορία στο κρυφό επίπεδο.

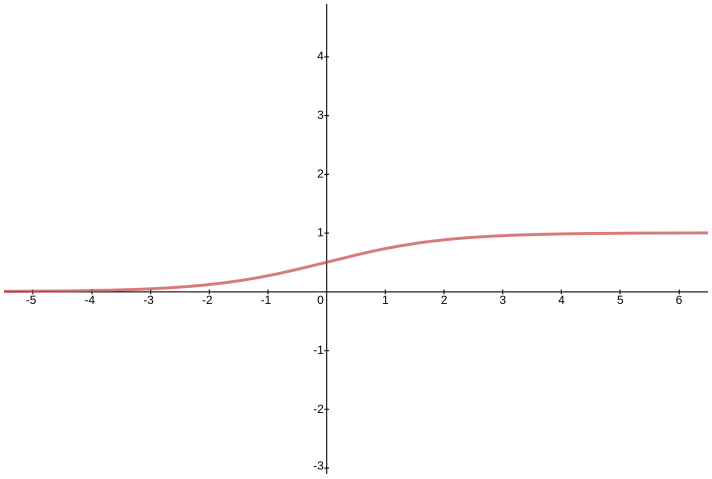
- **Κρυφό Επίπεδο.** Μπορεί να είναι παραπάνω από ένα επίπεδο, τα οποία επεξεργάζονται τα δεδομένα εισόδου και εξάγουν τα κατάλληλα χαρακτηριστικά, τα οποία μεταβιβάζουν στο επίπεδο εξόδου. Καθώς αυξάνεται το βάθος των κρυφών επιπέδων τόσο αυξάνεται και το βάθος των χαρακτηριστικών που εξάγονται.

- **Επίπεδο Εξόδου.** Με την επεξεργασία των δεδομένων από τα ενδιάμεσα επίπεδα,σε αυτό το επίπεδο λαμβάνεται μία απόφαση από το δίκτυο.

2.2.3 Συνάρτηση ενεργοποίησης

Όπως έχει ήδη αναφερθεί, η συνάρτηση ενεργοποίησης έχει την ευθύνη να αποφασίσει εάν ένας νευρώνας πρέπει να ενεργοποιηθεί ή όχι, εισάγοντας μια μη γραμμικότητα στην έξοδο. Στα νευρωνικά δίκτυα είναι απαραίτητη η συνάρτηση ενεργοποίησης ώστε να έχει την δυνατότητα να πραγματοποιήσει πολύπλοκα έργα και να δώσει εύστοχα αποτελέσματα. Παρακάτω παρουσιάζονται ορισμένες κλασσικές συναρτήσεις ενεργοποίησης.
Σιγμοειδής συνάρτηση. Η σιγμοειδής συνάρτηση ενεργοποίησης εκφράζεται από τον μαθηματικό τύπο.

$$S(x) = \frac{1}{1 - e^{-x}} \tag{2.3}$$



Σχήμα 3: Σιγμοειδής συνάρτηση

Υπερβολική Εφαπτομένη συνάρτηση. Η tanh συνάρτηση ενεργοποίησης εκφράζεται από τον μαθηματικό τύπο

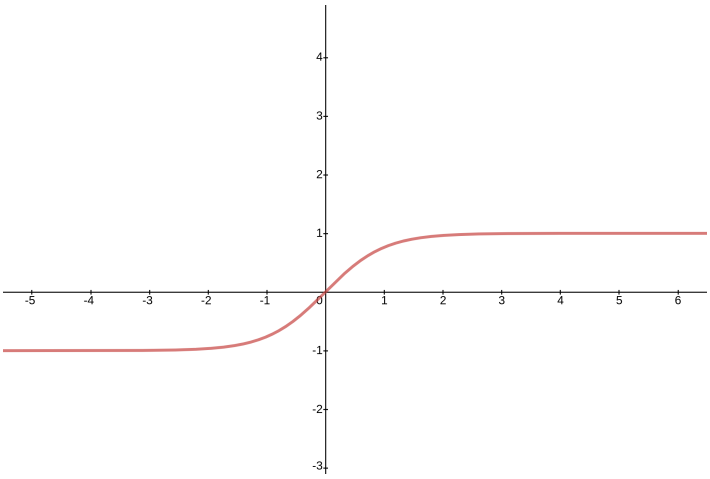
$$f(x) = tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.4}$$

Rectified Linear Unit (ReLU). Η ReLU συνάρτηση ενεργοποίησης εκφράζεται από τον μαθηματικό τύπο

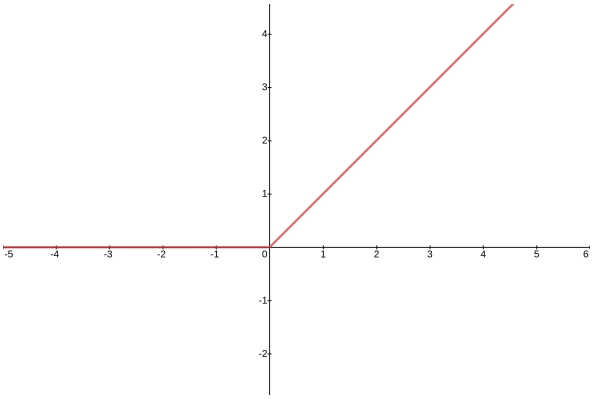
$$f(x) = x^{+} = max(o, x) \tag{2.5}$$

2.2.4 Συνάρτηση Απώλειας

Στόχος στην επιβλεπόμενη και στην ενισχυτική μάθηση είναι να επιστρέφει μια συνάρτηση, η οποία θα συσχετίζει εύστοχα τα δεδομένα εισόδου με τις επιθυμητές εξόδους. Με αυτό τον τρόπο εισάγεται η έννοια της *Συνάρτησης απώλειας* $\mathcal{L}$**(ŷ,y)** η οποία ποσοτικοποιεί την ζημιά που προκύπτει, όταν ο αλγόριθμος προβλέπει **ŷ** , ενώ η πραγματική τιμή είναι **y**. Η Συνάρτηση απώλειας, πρέπει



Σχήμα 4: Υπερβολική Εφαπτομένη συνάρτηση



Σχήμα 5: Rectified Linear Unit (ReLU)

να είναι κάτω φραγμένη, με το ελάχιστο να συμβαίνει, όταν η πρόβλεψη είναι σωστή. Οι παράμετροι, συνεπώς, του δικτύου μετά την εκπαίδευση, θα έχουν τις κατάλληλες τιμές ώστε η $\mathcal{L}$ να έχει ελαχιστοποιηθεί. Θεωρώντας μια συνάρτηση εισόδου - εξόδου $y = f(x;\theta)$ την οποία προσπαθεί να προσεγγίσει το δίκτυο, τότε προκύπτει ότι $\mathcal{L}(\hat{y},y) = \mathcal{L}(f(x;\theta),y) = \mathcal{L}(\theta)$, όπου $\boldsymbol{\theta}$ τα βάρη του νευρωνικού δικτύου. Συνεπώς, στόχος της εκπαίδευσης είναι η επιλογή των βαρών $\boldsymbol{\theta_d}$, τέτοια ώστε να επιτευχθεί η σχέση

$$\theta_d = \arg\min_{\theta} \mathcal{L}(\theta) \tag{2.6}$$

2.2.5 Οπίσθια Διάδοση

Όπως θα έχει γίνει ήδη αντιληπτό, η προσαρμογή των βαρών των νευρώνων ενός νευρωνικού δικτύου είναι η πιο σημαντική παράμετρος ώστε το μοντέλο να δίνει αξιόπιστα αποτελέσματα. Συνεπώς προκύπτει άμεσα η ανάγκη μιας διαδικασίας εκπαίδευσης, δηλαδή αναπροσαρμογής των βαρών, η οποία να μας εξασφαλίζει την σύγκλιση του μοντέλου στο επιθυμητό αποτέλεσμα. Η τεχνική της *Οπίσθιας Διάδοσης* είναι μια τέτοια διαδικασία, η οποία αναλύεται συνοπτικά στην συνέχεια[30]:

1. **Αρχικοποίηση.** Αρχικοποίηση των βαρών και των κατωφλιών εντός ομοιόμορφης κατανομής με μέση τιμή 0 και διασπορά τέτοια ώστε η τυπική απόκλιση των παραμέτρων των νευρώνων να είναι ανάμεσα στα όρια της συνάρτησης ενεργοποίησης.

2. **Παρουσίαση των παραδειγμάτων εκπαίδευσης.** Παρουσίαση στο δίκτυο μιας εποχής από δεδομένα εκπαίδευσης. Ως εποχή ορίζουμε κάθε μια συνολική διαδικασία εκπαίδευσης ενός συνόλου δεδομένων. Για παράδειγμα αν εκπαιδεύσουμε μια ομάδα δεδομένων για τρεις εποχές,

σημαίνει πως θα γίνει αναπροσαρμογή των βαρών των νευρώνων τρεις φορές για κάθε ένα από τα δεδομένα. Για κάθε παράδειγμα στο σύνολο, πραγματοποιείται μια αλληλουχία από ευθύς υπολογισμούς και υπολογισμούς ανάδρασης που περιγράφονται στα βήματα 3 και 4.

3. **Ευθύς υπολογισμός.** Ορίζουμε το παράδειγμα εισόδου ως $(\mathbf{x}(\mathbf{n}), \mathbf{d}(\mathbf{n})$, όπου το $\mathbf{x}(\mathbf{n})$ εφαρμόζεται στην είσοδο του δικτύου και το $\mathbf{d}(\mathbf{n})$ αντιπροσωπεύει την επιθυμητή έξοδο για την συγκεκριμένη είσοδο. Υπολογίζονται τα τελικά αποτελέσματα στην έξοδο $\mathbf{o_j}(\mathbf{n})$ για τον $\mathbf{j}$ νευρώνα εξόδου και προκύπτει το σφάλμα εξόδου $\mathbf{e_j}(\mathbf{n}) = \mathbf{d_j}(\mathbf{n}) - \mathbf{o_j}(\mathbf{n})$

4. **Υπολογισμός ανάδρασης.** Σε αυτό το βήμα υπολογίζονται τα $\boldsymbol{\delta_s}$ (local gradients) του δικτύου σύμφωνα με τους τύπους

$$\delta_j^{(l)}(n) = \begin{cases} \mathbf{e_j^{(L)}}(\mathbf{n})\boldsymbol{\varphi}_\mathbf{j}'(\boldsymbol{\upsilon}_\mathbf{j}^{(\mathbf{L})}(\mathbf{n})) & \text{,για τους νευρώνες του επιπέδου εξόδου} \\ \boldsymbol{\varphi}_\mathbf{j}'(\boldsymbol{\upsilon}_\mathbf{j}^{(\mathbf{l})}(\mathbf{n}))\sum_\mathbf{k} \boldsymbol{\delta}_\mathbf{k}^{(\mathbf{l+1})}(\mathbf{n})\mathbf{w}_\mathbf{kj}^{(\mathbf{l+1})}(\mathbf{n}) & \text{,για τους νευρώνες των κρυφών επιπέδων} \end{cases}$$

όπου στον παραπάνω τύπο ο 'τόνος' ισοδυναμεί με παράγωγο ως προς την μεταβλητή. Εν κατακλείδι, η προσαρμογή των βαρών γίνεται σύμφωνα με τον τύπο

$$\mathbf{w_{ji}^{(l)}}(\mathbf{n+1}) = \mathbf{w_{ji}^{(l)}}(\mathbf{n}) + \boldsymbol{\alpha}[\mathbf{w_{ji}^{(l)}}(\mathbf{n-1})] + \boldsymbol{\eta}\boldsymbol{\delta_j}(\mathbf{l})(\mathbf{n})\mathbf{y_i^{(l-1)}}(\mathbf{n})$$

όπου $\boldsymbol{\eta}$ είναι η παράμετρος του ρυθμού εκμάθησης.

5. **Επανάληψη.**Επανάληψη των προηγούμενων βημάτων για νέες εποχές των παραδειγμάτων εκπαίδευσης έως ότου πληρείται κάποιο κριτήριο τερματισμού.

2.2.6 Βελτιστοποίηση

Προκειμένου να εκπαιδεύσουμε το μοντέλο, πρέπει να λύσουμε το πρόβλημα βελτιστοποίησης της συνάρτησης απώλειας. Μια κοινή λύση είναι να χρησιμοποιηθεί η *μέθοδος με βάση την κλίση (gradient descend)*. Gradient είναι η κλίση της εφαπτομένης της συνάρτησης σε αυτό το σημείο και δείχνει την κατεύθυνση της μεγαλύτερης αύξησης της. Οι μέθοδοι αυτοί προσπαθούν να ελαχιστοποιήσουν την συνάρτηση απώλειας $\mathcal{L}(\theta)$ υπολογίζοντας επανειλημμένα μια εκτίμηση της πάνω σε ένα σύνολο εκμάθησης, υπολογίζοντας τις παραγώγους των παραμέτρων θ του μοντέλου σε σχέση με την εκτίμηση των απωλειών και ενημερώνοντας τις παραμέτρους στην αντίθετη κατεύθυνση της κλίσης.[31]

Η μέθοδος με βάση την κλίση είναι μια από τους πιο δημοφιλείς αλγόριθμους για τη βελτιστοποίηση σε νευρωνικά δίκτυα. Υπολογίζει τη κλίση(παράγωγο) της συνάρτησης απωλειών σε σχέση με τις παραμέτρους θ για ολόκληρο το σύνολο δεδομένων. Ο ρυθμός εκμάθησης $\boldsymbol{\eta}$ είναι μια '*υπερπαράμετρος*' που ελέγχει το βαθμό στον οποίο οι παράμετροι του μοντέλου προσαρμόζονται σε σχέση με την παράγωγο της συνάρτησης απώλειας. Η μέθοδος με βάση την κλίση ορίζεται τυπικά ως:

$$\theta = \theta - \eta\nabla_\theta\mathcal{L}(\theta) \tag{2.7}$$

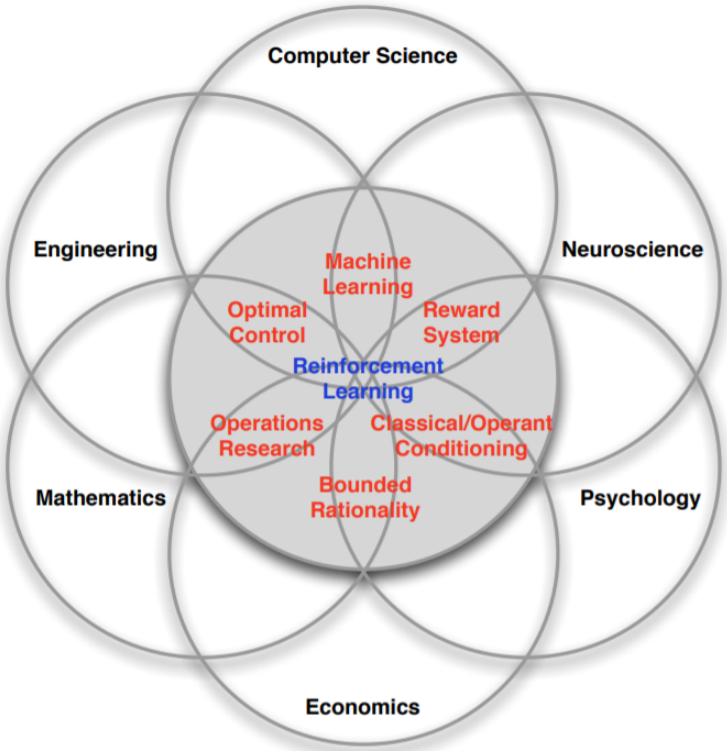
2.3 Ενισχυτική μάθηση

2.3.1 Εισαγωγή

Η ιδέα της εκμάθησης μέσω αλληλεπίδρασης με το περιβάλλον μας είναι η πρωταρχική μορφή εκμάθησης. Από την στιγμή που ένας έμβιος οργανισμός γεννιέται και ξεκινά να πράττει δεν έχει κάποιον 'δάσκαλο', αλλά έχει μια άμεση σύνδεση αίσθησης - πράξης με το περιβάλλον του. Με την εξάσκηση αυτής της επικοινωνίας, ο οργανισμός παράγει μια πληθώρα πληροφοριών αιτίου - αιτιατού που αφορούν τις συνέπειες των πράξεων του καθώς και την ανάπτυξη μιας καλύτερης στρατηγικής

για την επίτευξη στόχων. Στην διάρκεια της ζωής του, οι αλληλεπιδράσεις αυτές αποτελούν αδιαμφισβήτητα μια κύρια πηγή γνώσης για το περιβάλλον του και τον εαυτό του.[32]

Το πεδίο της ενισχυτικής μάθησης μελετά την υπολογιστική προσέγγισή για την εκμάθηση μέσω αλληλεπίδρασης. Αντί να προσπαθήσουμε άμεσα να μοντελοποιήσουμε το πως μαθαίνουν οι άνθρωποι και τα ζώα, εξερευνούμε εξιδανικευμένες καταστάσεις εκμάθησης και αξιολογούμε την αποτελεσματικότητα διαφόρων μεθόδων μάθησης. Συγκεκριμένα, η ενισχυτική μάθηση είναι πεδίο της μηχανικής μάθησης το οποίο ασχολείται με το πως αλγοριθμικοί πράκτορες θα πρέπει να πράττουν σε ένα περιβάλλον, με στόχο να μεγιστοποιούν κάποιο κέρδος.[33] Το πεδίο της ενισχυτικής μάθησης αποτελεί συνδυασμό πολλών επιστημονικών τομέων, όπως φαίνεται και στο Σχήμα 6.



Σχήμα 6: Διαφορετικά επιστημονικά πεδία που συνθέτουν την Ενισχυτική Μάθηση

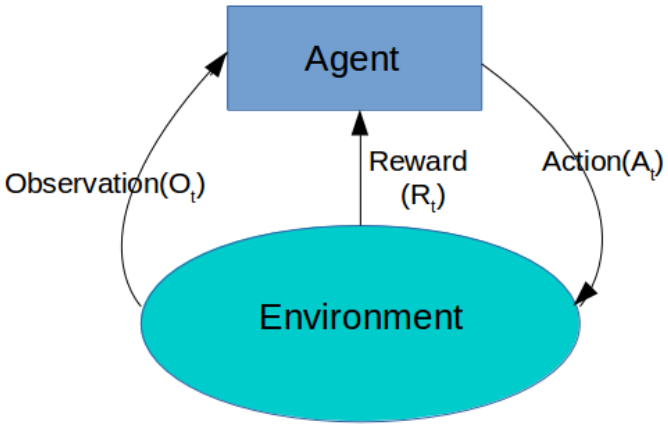
Υπάρχουν ορισμένες βασικές διαφορές μεταξύ της ενισχυτικής μάθησης και των υπολοίπων αλγορίθμων μηχανικής μάθησης. Συγκεκριμένα,

- Σε αντίθεση με την επιβλεπόμενη μάθηση, δεν υπάρχει ’δάσκαλος’, παρά μόνο ένα σήμα επιβράβευσης ή τιμωρίας.
- Η ανάδραση κατά την εκπαίδευση είναι καθυστερημένη, όχι άμεση.
- Η χρονική διάρκεια είναι αρκετά σημαντική. Υπάρχει η έννοια της αλληλουχίας και της συσχέτισης στα δεδομένα. Δεν είναι ,δηλαδή, ανεξάρτητα και τυχαία κατανεμημένα.
- Οι πράξεις του πράκτορα επηρεάζουν τις μεταβάσεις του, άρα και τα μελλοντικά δεδομένα που θα λάβει και με τα οποία θα εκπαιδευτεί.

2.3.2 Μοντελοποίηση του προβλήματος

Η στρατηγική για την επίλυση ενός προβλήματος ενισχυτικής μάθησης είναι η χρήση στατιστικών μοντέλων και μεθόδων δυναμικού προγραμματισμού για να προσεγγιστεί η χρησιμότητα μιας ορισμένης πράξης στις διάφορες καταστάσεις του περιβάλλοντος.

Στο κλασσικό μοντέλο ενισχυτικής μάθησης, ένας πράκτορας είναι συνδεδεμένος με το περιβάλλον του μέσω της παρατήρησης του και της πράξης, όπως φαίνεται αναλυτικά στο Σχήμα 7. Σε κάθε βήμα **t** της αλληλεπίδρασης, ο πράκτορας λαμβάνει ως είσοδο, κάποια παρατήρηση της τωρινής του κατάστασης $\mathbf{O_t}$ στο περιβάλλον. Εν συνεχεία, αποφασίζει μια πράξη $\mathbf{A_t}$ μέσω κάποια στρατηγικής, η οποία θεωρείται η έξοδος του. Η κάθε πράξη με την σειρά της, αλλάζει την κατάσταση του, και παράγει και μια αξία της συγκεκριμένης κατάστασης η οποία δίνεται στον πράκτορα ως ένα βαθμωτό σήμα ενίσχυσης (ανταμοιβή) $\mathbf{R_t}$. Στόχος του πράκτορα είναι η επιλογή πράξεων που στοχεύουν στην αύξηση του μακροπρόθεσμου αθροίσματος των ανταμοιβών. Η εκπαίδευση επιτυγχάνεται μέσω προσπάθειας - λάθους για αρκετά βήματα, με χρήση διαφόρων αλγορίθμων και μεθόδων που θα αναλυθούν στα επόμενα κεφάλαια.[33]



Σχήμα 7: Αλληλεπίδραση Πράκτορα - Περιβάλλοντος

Πράξη

Ως πράξη $\mathbf{A_t}$, ορίζουμε τις δυνατές επιλογές που έχει ο πράκτορας για να αλληλεπιδράσει με το περιβάλλον του.

Ανταμοιβή

Ως ανταμοιβή $\mathbf{R_t}$, ορίζουμε ένα βαθμωτό σήμα ανάδρασης, το οποίο υποδεικνύει το πόσο καλά έπραξε ένας πράκτορας σε κάθε βήμα **t**. Όπως ήδη αναφέρθηκε, στόχος του πράκτορα είναι η επίτευξη της μέγιστης μακροπρόθεσμης ανταμοιβής. Η ενισχυτική μάθηση βασίζεται στην **υπόθεση ενίσχυσης**.

Ορισμός

Όλοι οι στόχοι μπορούν να περιγραφούν μέσω της μεγιστοποίησης των αναμενόμενων μακροπρόθεσμων ενισχύσεων (ανταμοιβών).

Ιστορικό & Κατάσταση

Ως ιστορικό ορίζουμε μια αλληλουχία από παρατηρήσεις $\mathbf{O_t}$, πράξεις $\mathbf{A_t}$, ανταμοιβές $\mathbf{R_t}$

$$\mathbf{H_t} = \mathbf{O_1}, \mathbf{R_1}, \mathbf{A_1}, ..., \mathbf{A_{t-1}}, \mathbf{O_t}, \mathbf{R_t} \tag{2.8}$$

Ως κατάσταση $\mathbf{S_t}$ ορίζουμε την πληροφορία που χρησιμοποιείται για να καθοριστεί το τι θα συμβεί μελλοντικά. Η κατάσταση, συνεπώς είναι συνάρτηση του Ιστορικού:

$$\mathbf{S_t} = \mathbf{f}(\mathbf{H_t}) \tag{2.9}$$

Ορίζουμε ως κατάσταση περιβάλλοντος $\mathbf{S_t^e}$ το σύνολο των δεδομένων που χρησιμοποιεί ώστε να διαλέξει την επόμενη παρατήρηση και ανταμοιβή. Η κατάσταση του περιβάλλοντος δεν είναι συνήθως ορατή στον πράκτορα, αλλά ακόμα και στις περιπτώσεις που έχει πλήρη γνώση, είναι πιθανό να εμπεριέχει περιττή πληροφορία.

Αντιστοίχως, ορίζουμε ως κατάσταση του πράκτορα $\mathbf{S_t^a}$, το σύνολο των πληροφοριών που αφορούν την εσωτερική δομή του και την λειτουργία του. Μπορεί να αναπαρασταθεί από κάθε συνάρτηση ως προς το ιστορικό

$$\mathbf{S_t^a} = \mathbf{f}(\mathbf{H_t}) \tag{2.10}$$

Πλήρως παρατηρήσιμο περιβάλλον

Σε ένα πλήρως παρατηρήσιμο περιβάλλον, ο πράκτορας παρατηρεί απευθείας την κατάσταση του περιβάλλοντος.

$$\mathbf{O_t} = \mathbf{S_t^a} = \mathbf{S_t^e} \tag{2.11}$$

Μερικώς παρατηρήσιμο περιβάλλον

Σε ένα μερικώς παρατηρήσιμο περιβάλλον, ο πράκτορας παρατηρεί έμμεσα την κατάσταση του περιβάλλοντος.

$$\mathbf{S_t^a} \neq \mathbf{S_t^e} \tag{2.12}$$

Συνεπώς, πρέπει να κατασκευάσει την δική του αναπαράσταση του περιβάλλοντος.

2.3.3 Μαρκοβιανές διαδικασίες αποφάσεων

Στα προβλήματα Ενισχυτικής μάθησης, ο πράκτορας λαμβάνει αποφάσεις από παρατηρήσεις της κατάστασης του περιβάλλοντος στο οποίο ενεργεί. Καθίσταται ,λοιπόν, σαφής η εξάρτηση των πράξεων που θα πραγματοποιήσει με τις παρατηρήσεις του. Στην ιδανική περίπτωση, θα θέλαμε το σήμα που λαμβάνει ο πράκτορας από το περιβάλλον ως είσοδο(παρατήρηση) να μην περιέχει μόνο τις στιγμιαίες μετρήσεις, αλλά και τις παρελθοντικές καταστάσεις που είναι απαραίτητες για την λήψη ικανοποιητικών αποφάσεων. Μια παρατήρηση που πληροί την παραπάνω προϋπόθεση λέμε ότι πληρεί την *Μαρκοβιανή Ιδιότητα*.[32]

Ορισμός

Μια κατάσταση $\mathbf{S_t}$ ορίζεται ως *Μαρκοβιανή* αν και μόνο αν

$$\Pr[\mathbf{S_{t+1}}|\mathbf{S_t}] = \Pr[\mathbf{S_{t+1}}|\mathbf{S_1}, ..., \mathbf{S_t}] \tag{2.13}$$

Με απλά λόγια, μια κατάσταση για να είναι Μαρκοβιανή, θα πρέπει το μέλλον να είναι ανεξάρτητο από το παρελθόν δεδομένου του παρόντος.

Για μια Μαρκοβιανή κατάσταση $\mathbf{s}$ και την διάδοχη $\mathbf{s}^{'}$ ορίζουμε ως *Πιθανότητα Μεταβατικής Κατάστασης* και ως *Πίνακα Μεταβατικής Κατάστασης*, αντίστοιχα

$$\mathcal{P}_{\mathbf{ss}'} = \Pr[\mathbf{S_{t+1}} = \mathbf{s}^{'}|\mathbf{S_t} = \mathbf{s}], \mathcal{P} = \begin{bmatrix} \mathcal{P}_{11} & ... & \mathcal{P}_{1n} \\ . & & . \\ \mathcal{P}_{n1} & ... & \mathcal{P}_{nn} \end{bmatrix} \tag{2.14}$$

Με την βοήθεια των παραπάνω μεγεθών είμαστε, πλέον, σε θέση αναλύσουμε τις *Μαρκοβιανές Διαδικασίες*. Από την οπτική της στατιστικής, *Μαρκοβιανή Αλυσίδα ή Διαδικασία* είναι ένα στοχαστικό μοντέλο το οποίο περιγράφει μια αλληλουχία από δυνατά γεγονότα, στα οποία η πιθανότητα του καθενός, εξαρτάται μόνο από την κατάσταση που πραγματοποιήθηκε στο προηγούμενο γεγονός.[34] Από την σκοπιά του προβλήματος της Ενισχυτικής μάθησης, ορίζουμε ως μια Μαρκοβιανή διαδικασία μια τυχαία διαδικασία χωρίς μνήμη η οποία πληρεί την Μαρκοβιανή ιδιότητα. Συγκεκριμένα

Ορισμός

Μια *Μαρκοβιανή Διαδικασία (Αλυσίδα)* ορίζεται ως ένα σύνολο $< \mathcal{S}, \mathcal{P} >$

1. $\mathcal{S}$ είναι ένα πεπερασμένο σύνολο από καταστάσεις
2. $\mathcal{P}$ είναι ένας πίνακας μεταβατικών καταστάσεων πιθανότητας, $\mathcal{P}_{\mathbf{ss'}} = \Pr[\mathbf{S_{t+1}} = \mathbf{s'}|\mathbf{S_t} = \mathbf{s}]$

Αν εμπλουτίσουμε μια Μαρκοβιανή διαδικασία, προσθέτοντας ένα ποσό ανταμοιβής σε κάθε κατάσταση τότε δημιουργήσαμε μια *Μαρκοβιανή Διαδικασία Ανταμοιβής*.

Ορισμός

Μια *Μαρκοβιανή Διαδικασία Ανταμοιβής* ορίζεται ως ένα σύνολο $< \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma >$

1. $\mathcal{S}$ είναι ένα πεπερασμένο σύνολο από καταστάσεις
2. $\mathcal{P}$ είναι ένας πίνακας μεταβατικών καταστάσεων πιθανότητας, $\mathcal{P}_{\mathbf{ss'}} = \Pr[\mathbf{S_{t+1}} = \mathbf{s'}|\mathbf{S_t} = \mathbf{s}]$
3. $\mathcal{R}$ αποτελεί μια συνάρτηση ανταμοιβής, $\mathcal{R}_s = \mathbb{E}[R_{t+1}|S_t = s]$
4. γ είναι ένας παράγοντας έκπτωσης, όπου $\gamma \in [0, 1]$

Όπως έχει ήδη αναφερθεί, στόχος ενός πράκτορα είναι να διαλέξει την βέλτιστη πολιτική, ώστε να μεγιστοποιήσει την συνολική ανταμοιβή που θα λάβει από το περιβάλλον. Αν υποθέσουμε πως κάθε χρονική στιγμή **t** λαμβάνει ανταμοιβή $\mathbf{R_t}$, τότε η συνολική θα είναι

$$\textit{Συνολική Ανταμοιβή} = \sum_n R_n \tag{2.15}$$

Συνήθως, όμως, χρησιμοποιούμε έναν εκπτωτικό όρο στην κάθε μεμονωμένη ανταμοιβή. Αυτό εξυπηρετεί για αρκετούς λόγους. Αρχικά, από μαθηματική σκοπιά βεβαιώνουμε πως το άθροισμα δεν αποκλίνει όταν τα βήματα τείνουν στο άπειρο. Επιπλέον, η αβεβαιότητα σχετικά με το μέλλον ίσως να μην περιγράφεται πλήρως. Από την οπτική της ενισχυτικής μάθησης, η χρονική έκπτωση των ανταμοιβών προσομοιώνει την συμπεριφορά των ζωντανών οργανισμών, οι οποίοι δίνουν μεγαλύτερη έμφαση στην άμεση επιβράβευση. Τέλος, υπάρχουν ορισμένες περιπτώσεις που μπορεί να χρησιμοποιηθεί η συνολική ανταμοιβή χωρίς έκπτωση, όπως ορίστηκε προηγουμένως, αν όλες οι καταστάσεις σε μια μαρκοβιανή αλυσίδα, τερματίζουν. Η εκπτωτική αυτή συνολική ανταμοιβή ονομάζεται και *Συνάρτηση Επιστροφής* (G_t) και ορίζεται ως

$$G_t = \sum_{n=0}^{\infty} \gamma^n R_{t+n+1}, \text{όπου } \gamma \in [0, 1] \tag{2.16}$$

Ένα από τα σημαντικότερα μεγέθη για την ενισχυτική μάθηση είναι η συνάρτηση τιμών (Value Function) η οποία συνήθως ορίζεται ως $V(t)$. Η συνάρτηση αυτή αντιπροσωπεύει την αξία μιας κατάστασης στην οποία μπορεί να βρεθεί ο πράκτορας. Ισούται με την αναμενόμενη συνολική ανταμοιβή ενός πράκτορα που ξεκινά από την κατάσταση s. Καθίσταται, λοιπόν, σαφής η εξάρτηση της με την πολιτική που θα ακολουθήσει ο πράκτορας για την επιλογή πράξεων. Συνεπώς,

$$V(s) = \mathbb{E}[G_t|S_t = s] \tag{2.17}$$

Με την βοήθεια της συνάρτησης τιμών που μόλις ορίσαμε, είμαστε σε θέση να αναλύσουμε την *εξίσωση του Bellman* για τις Μαρκοβιανές Διαδικασίες Ανταμοιβής. Η εξίσωση αυτή προκύπτει από

απλή αλγεβρική αντικατάσταση στον τύπο της συνάρτησης τιμών, αλλά το αποτέλεσμα είναι ιδιαιτέρως σημαντικό. Συγκεκριμένα, η $V(s)$ μπορεί να χωριστεί σε δύο τμήματα, το άμεσο κέρδος R_{t+1} και την συνάρτηση τιμών για την επόμενη κατάσταση αλλά με έκπτωση.

$$V(s) = \mathbb{E}[G_t|S_t = s] = \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ...|S_t = s] =$$
$$\mathbb{E}[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + ...)|S_t = s] = \mathbb{E}[R_{t+1} + \gamma G_{t+1}|S_t = s] = \mathbb{E}[R_{t+1} + \gamma V(S_{t+1})|S_t = s]$$

Τέλος, είμαστε πλέον έτοιμοι να ορίσουμε πλήρως την *Μαρκοβιανή Διαδικασία Αποφάσεων(Markov Decision Process - MDP)* ως μια Μαρκοβιανή Διαδικασία Ανταμοιβής με την δυνατότητα λήψης αποφάσεων.

Ορισμός

Μια *Μαρκοβιανή Διαδικασία Αποφάσεων* ορίζεται ως ένα σύνολο $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$

1. $\mathcal{S}$ είναι ένα πεπερασμένο σύνολο από καταστάσεις
2. είναι ένα πεπερασμένο σύνολο από πράξεις
3. $\mathcal{P}$ είναι ένας πίνακας μεταβατικών καταστάσεων πιθανότητας, $\mathcal{P}^a_{ss'} = \Pr[S_{t+1} = s'|S_t = s, A_t = a]$
4. $\mathcal{R}$ αποτελεί μια συνάρτηση ανταμοιβής, $\mathcal{R}^a_s = \mathbb{E}[R_{t+1}|S_t = s, A_t = a]$
5. γ είναι ένας παράγοντας έκπτωσης, όπου $\gamma \in [0, 1]$

2.3.4 Σχεδίαση με Δυναμικό Προγραμματισμό

Ο όρος δυναμικός προγραμματισμός (dynamic programming) αποδίδεται σε αλγορίθμους, που έχουν ως σκοπό τον υπολογισμό βέλτιστων πολιτικών σε μια Μαρκοβιανή Διαδικασία Αποφάσεων, δεδομένου ενός πλήρους μοντέλου του περιβάλλοντος. Οι κλασσικοί αλγόριθμοι δυναμικού προγραμματισμού έχουν περιορισμένη πρακτική χρησιμότητα εξαιτίας της μεγάλης υπολογιστικής πολυπλοκότητας τους και της υπόθεσης της ύπαρξης πλήρους μοντέλου του περιβάλλοντος. Ωστόσο, η σπουδαιότητά τους από θεωρητικής άποψης είναι μεγάλη καθώς παρέχουν το απαραίτητο υπόβαθρο για την κατανόηση της πλειονότητας των πρακτικά εφαρμόσιμων μεθόδων ενισχυτική μάθησης, οι οποίες στην ουσία μπορούν να ιδωθούν ως απόπειρες επίτευξης του ίδιου στόχου, με μικρότερη υπολογιστική πολυπλοκότητα και χωρίς να απαιτείται πλήρες μοντέλο του περιβάλλοντος. Στην ουσία, η κύρια ιδέα του δυναμικού προγραμματισμού (αλλά και της Ενισχυτικής Μάθησης γενικότερα) είναι η χρήση των συναρτήσεων τιμών για την οργάνωση και δόμηση του χώρου των πολιτικών, με σκοπό την αποδοτική αναζήτηση των βέλτιστων εξ αυτών.[35][36]

Βασικοί όροι για το δυναμικό προγραμματισμό είναι η αξιολόγηση πολιτικής (policy evaluation) και η βελτίωση πολιτικής (policy improvement). Η αξιολόγηση πολιτικής αναφέρεται στον επαναληπτικό υπολογισμό των συναρτήσεων τιμών για μια δεδομένη πολιτική και μπορεί να συναντηθεί και ως το πρόβλημα της πρόβλεψης (prediction problem). Χρησιμοποιώντας την εξίσωση Bellman είναι δυνατόν να υπολογιστεί η συνάρτηση τιμών κατάστασης V^π, υπό την τρέχουσα πολιτική π

$$V^\pi(s) = \sum_\alpha \pi(s, \alpha) \sum_{s'} \mathcal{P}^\alpha_{ss'}[\mathcal{R}^\alpha_s + \gamma V_k(s')] \tag{2.18}$$

Αντίστοιχα, η βελτίωση πολιτικής αναφέρεται στον υπολογισμό μιας βελτιωμένης πολιτικής, δεδομένης της συνάρτησης αξίας για την πολιτική αυτήν. Για κάποια κατάσταση s αρκεί να εξακριβωθεί, αν είναι καλό να τροποποιήσουμε την τρέχουσα πολιτική ώστε να επιλέγει ντετερμινιστικά κάποια ενέργεια $a = \pi(s)$. Ένας τρόπος για να γίνει αυτό, είναι ο υπολογισμός της αξία επιλογής της a στην

τρέχουσα κατάσταση και κατόπιν της εφαρμογής της τρέχουσας πολιτικής . Συνεπώς, χρησιμοποιώντας την παρακάτω εξίσωση για τον υπολογισμό της συνάρτησης τιμών ενέργειας,

$$Q^{\pi}(s,\alpha) = \sum_{s'} \mathcal{P}^{a}_{ss'}[\mathcal{R}^{a}_{s} + \gamma V^{\pi}(s')] \tag{2.19}$$

αρκεί να διερευνηθεί εάν ισχύει $\mathbf{Q^{\pi}}(\mathbf{s},\boldsymbol{\alpha}) \geq \mathbf{V^{\pi}}(\mathbf{s})$. Εάν αυτό ισχύει –πράγμα που σημαίνει ότι είναι καλύτερο να επιλεχθεί η α και να ακολουθηθεί η π έπειτα, από το να ακολουθείται η π συνέχεια– τότε μπορεί να υποτεθεί ότι κάποιος θα ανέμενε ότι θα ήταν καλύτερο να επιλέγεται η α κάθε φορά που ο πράκτορας αντιμετωπίζει την s, κι ότι η καινούρια πολιτική θα ήταν σε γενικές γραμμές καλύτερη από την προηγούμενη. Αυτό ισχύει σε περίπτωση που συγκρίνονται δύο οποιεσδήποτε ντετερμινιστικές πολιτικές και π_0, ως ειδική περίπτωση του λεγόμενου θεωρήματος βελτίωσης πολιτικής (policy improvement theorem).

Θεώρημα

Για κάθε πιθανό ζεύγος, ντετερμενιστικών πολιτικών π και $\pi_{\prime}$, εάν ισχύει $Q^{\pi}(s,\alpha) \geq V^{\boldsymbol{\pi}}(s), \forall s \in \mathbb{S}$, τότε $V^{\pi'} \geq V^{\pi}$

Αυτό σημαίνει ότι η π_0 είναι καλύτερη από την π και η εφαρμογή της επιφέρει μεγαλύτερη ή ίση συνολική ανταμοιβή από όλες τις καταστάσεις. Το συμπέρασμα αυτό μπορεί εύκολα να επεκταθεί στην περίπτωση των στοχαστικών πολιτικών, όπου σε περίπτωση που έχουμε δύο ή περισσότερες βέλτιστες ενέργειες για κάποια κατάσταση, τότε σε κάθε μια αποδίδεται ένα μέρισμα των πιθανοτήτων.

Συνδυάζοντας την αξιολόγηση πολιτικής με τη βελτίωση πολιτικής, προκύπτουν οι δύο ευρύτερα διαδεδομένες μέθοδοι δυναμικού προγραμματισμού, η επανάληψη ως προς την πολιτική (policy iteration) και η επανάληψη ως προς την αξία (value iteration), οι οποίες είναι σε θέση να υπολογίζουν βέλτιστες συναρτήσεις αξίας και πολιτικές, για πεπερασμένες Μαρκοβιανές διαδικασίες αποφάσεων για τις οποίες είναι γνωστό το πλήρες μοντέλο του περιβάλλοντος. Η επανάληψη ως προς την πολιτική, υλοποιείται με διαδοχικές εναλλαγές αξιολόγησης πολιτικής με βελτίωση πολιτικής, μέχρις ότου να επέλθει σύγκλιση. Σημαντικό μειονέκτημα του αλγορίθμου είναι ότι σε κάθε επανάληψη εκτελείται ένα βήμα αξιολόγησης πολιτικής, το οποίο συνεπάγεται μεγάλο υπολογιστικό κόστος. Ο αλγόριθμος της επανάληψης ως προς την αξία βελτιώνει την παραπάνω διαδικασία, προσεγγίζοντας επαναληπτικά τη βέλτιστη συνάρτηση V , σύμφωνα με τον παρακάτω τύπο:

$$V_{k+1} = \max_{\alpha} \sum_{s'} \mathcal{P}^{\alpha}_{ss'}[\mathcal{R}^{\alpha}_{s} + \gamma V_k(s')] \tag{2.20}$$

όπου V_{k+1} είναι η εκτίμηση της συνάρτησης αξίας στο βήμα $k+1$

Επιπλέον, γενικεύοντας τις παραπάνω ιδέες, μπορεί να οριστεί η γενικευμένη επανάληψη με βάση την πολιτική (generalized policy iteration), ως η αλληλεπίδραση 2 διεργασιών που επενεργούν πάνω σε μια προσεγγιστική συνάρτηση αξίας και μια προσεγγιστική πολιτική. Η μια διεργασία (αξιολόγηση πολιτικής) θεωρεί την πολιτική σταθερή και πραγματοποιεί μια αξιολόγησή της κατά κάποιον τρόπο, αλλάζοντας την συνάρτηση αξίας, ώστε να προσεγγίζει περισσότερο την πραγματική συνάρτηση αξίας για την πολιτική αυτή. Η άλλη (βελτίωση πολιτικής), θεωρεί την συνάρτηση αξίας σταθερή και τροποποιεί την πολιτική με σκοπό να τη βελτιώσει, θεωρώντας ότι η συνάρτηση αξίας της είναι η τρέχουσα συνάρτηση αξίας. Παρόλο που η κάθε μία διεργασία πραγματοποιεί αλλαγές στο στοιχείο βάσει του οποίου παίρνει αποφάσεις η άλλη, ουσιαστικά συνεργάζονται προκειμένου να βρουν μια κοινή λύση: μια πολιτική και μια συνάρτηση αξίας που δεν τροποποιούνται από καμία από τις διεργασίες, πράγμα που σημαίνει ότι είναι βέλτιστες. Σε αρκετές περιπτώσεις και υπό συνθήκες, η γενικευμένη επανάληψη με βάση την πολιτική έχει αποδειχθεί ότι συγκλίνει στη βέλτιστη πολιτική.

Μια ιδιότητα των μεθόδων δυναμικού προγραμματισμού που χρήζει αναφοράς είναι το γεγονός, ότι αυτές ενημερώνουν εκτιμήσεις για τις αξίες των καταστάσεων με βάση εκτιμήσεις για τις αξίες των διάδοχων καταστάσεων. Η ιδέα της ενημέρωσης εκτιμήσεων, βάσει άλλων εκτιμήσεων αναφέρεται

στη διεθνή βιβλιογραφία με τον όρο bootstrapping. Αρκετές μέθοδοι ενισχυτικής μάθησης, εφαρμόζουν bootstrapping παρ'όλο που δεν απαιτούν πλήρες μοντέλο του περιβάλλοντος, όπως οι μέθοδοι μάθησης χρονικών διαφορών που θα συζητηθούν σε επόμενη ενότητα.

2.3.5 Πρόβλεψη άνευ Μοντέλου

Στην προηγούμενη ενότητα, μελετήσαμε την σχεδίαση αλγορίθμου με χρήση *Δυναμικού Προγραμματισμού* για την επίλυση προβλημάτων, με γνώση του μοντέλου του προβλήματος, ή αλλιώς με γνωστή *Μαρκοβιανή Διαδικασία Αποφάσεων*. Στο παρόν κεφάλαιο θα αναλύσουμε την αξιολόγηση πολιτικών για άγνωστα μοντέλα. Συγκεκριμένα:

- Αξιολόγηση πολιτικής Monte Carlo
- Αξιολόγηση πολιτικής Χρονικών Διαφορών TD(0)
- Αξιολόγηση πολιτικής Χρονικών Διαφορών TD(λ)

2.3.5.1 Αξιολόγηση πολιτικής Monte Carlo

Οι μέθοδοι Monte Carlo αναφέρονται σε μια κατηγορία αλγορίθμων ενισχυτικής μάθησης, που αποσκοπούν στη μάθηση συναρτήσεων αξίας και βέλτιστων πολιτικών, με χρήση εμπειρίας υπό τη μορφή δειγμάτων επεισοδίων (sample episodes). Σημεία-κλειδί των μεθόδων Monte Carlo είναι η απλότητά τους και ο τρόπος με τον οποίον σχετίζονται με τις υπόλοιπες μεθόδους ενισχυτικής μάθησης, από θεωρητικής σκοπιάς. Σε αντίθεση με το δυναμικό προγραμματισμό, οι εν λόγω μέθοδοι δεν απαιτούν πλήρη γνώση σχετικά με το περιβάλλον. Συγκεκριμένα, αν και το μοντέλο του περιβάλλοντος είναι απαραίτητο, απαιτείται από αυτό μόνο η δυνατότητα παραγωγής δειγμάτων μεταβάσεων. Με άλλα λόγια, το μόνο που απαιτείται από τις μεθόδους Monte Carlo είναι να λαμβάνει δείγματα ακολουθιών καταστάσεων, ενεργειών και ανταμοιβών. Με τα δείγματα αυτά μπορεί να παράγει την αξία μας κατάστασης, σύμφωνα με τον τύπο

$$V(s_t) \leftarrow V(s_t) + \alpha[G_t - V(s_t)] \tag{2.21}$$

Συγκεκριμένα, όσον αφορά την αξιολόγηση πολιτικής, ας υποθέσουμε πως θέλουμε να υπολογίσουμε την αξία μιας κατάστασης, δεδομένου ενός συνόλου επεισοδίων που προκύπτουν, ακολουθώντας μια πολιτική π και περνώντας από τις αντίστοιχες καταστάσεις s. Κάθε εμφάνιση μιας κατάστασης s σε ένα επεισόδιο, καλείται επίσκεψη στην κατάσταση. Η μέθοδος Monte Carlo κάθε-επίσκεψης, υπολογίζει την $V^{\pi}(s)$, ως τον μέσο όρο των επιστροφών ακολουθώντας όλες τις επισκέψεις στην κατάσταση s σε ένα σύνολο επεισοδίων. Η μέθοδος Monte Carlo πρώτης-επίσκεψης, υπολογίζει την μέση τιμή μόνο για τις επιστροφές που ακολουθούν τις πρώτες επισκέψεις στην κατάσταση s. Οι δύο ,εν λόγω, τεχνικές αξιολόγησης πολιτικής είναι αρκετά παρεμφερείς, αλλά με διαφορετικές θεωρητικές ιδιότητες. Στο Σχήμα 8 παρουσιάζεται ο αλγόριθμος για την μέθοδο πρώτης επίσκεψης, που αναλύσαμε προηγουμένως.

2.3.5.2 Αξιολόγηση πολιτικής Χρονικών Διαφορών TD(0)

Η μάθηση χρονικών διαφορών (temporal difference learning) είναι μια από τις πιο βασικές και καινοτόμες ιδέες που έχει αναδείξει το επιστημονικό πεδίο της ενισχυτικής μάθησης. Οι μέθοδοι μάθησης χρονικών διαφορών, θα μπορούσαν να ιδωθούν ως συνδυασμός στοιχείων των μεθόδων δυναμικού προγραμματισμού και των μεθόδων Monte Carlo. Συγκεκριμένα, όπως και στο δυναμικό προγραμματισμό, έτσι και στις μεθόδους μάθησης χρονικών διαφορών, η ενημέρωση των εκτιμήσεων για τις συναρτήσεις αξίας γίνεται εν μέρει βάσει των εκτιμήσεων που έχει ήδη μάθει ο πράκτορας, μέχρι τη χρονική στιγμή εκείνη, χωρίς να είναι απαραίτητο να έχει προκύψει κάποιο οριστικό αποτέλεσμα. Αντίστοιχα, όπως και στις μεθόδους Monte Carlo, δεν απαιτείται πλήρης γνώση για το μοντέλο του περιβάλλοντος στο οποίο δρα ο πράκτορας, παρά μόνον εμπειρία αλληλεπίδρασης του με αυτό.[37]

```
Αρχικοποίηση:
        π ← πολιτική προς αξιολόγηση
        V ← αυθαίρετη συνάρτηση τιμής
        Επιστροφές (s) ← Άδεια λίστα
Επανάλαβε:
        παραγωγή ενός επεισοδίου χρησιμοποιώντας την πολιτική π
        για κάθε κατάσταση S που εμφανίζεται στο επεισόδιο:
                G ← επιστροφή που ακολουθεί την πρώτη εμφάνηση του S
                Προσθήκη της επιστροφής G στην λίστα Επιστροφές(s)
                V(s) ← μέση τιμή(Επιστροφές(s))
```

Σχήμα 8: Αλγόριθμος αξιολόγησης πολιτικής Monte Carlo

Οι μέθοδοι Monte Carlo προσπαθούν να εκτιμήσουν κατ' ευθείαν το R_t, κρατώντας το μέσο όρο των ενισχύσεων που λαμβάνει ο πράκτορας και μετακινώντας τη συνάρτηση τιμών προς αυτήν την κατεύθυνση, όπως αναφέρθηκε προηγουμένως. Αντιθέτως, η ιδέα πίσω από τις TD μεθόδους είναι η συνάρτηση τιμών να μετακινείται προς τον ίδιο στόχο χρησιμοποιώντας μόνο την άμεση ενίσχυση και τις ήδη αποθηκευμένες τιμές ως

$$V(s_t) \leftarrow V(s_t) + \alpha[R_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \tag{2.22}$$

Η πιο απλή μέθοδος της οικογένειας αυτής, TD(0) αξιολογεί την πολιτική όπως φαίνεται στο Σχήμα 9.

```
Αρχικοποίηση:
        α ←  μέγεθος βήματος ∈ (0,1]
        V(s) ← αυθαίρετη συνάρτηση τιμής για κάθε s, εκτός V(τελική κατάσταση) = 0
Επανάλαβε:
        Αρχικοποίηση S
        Επανάλαβε για κάθε βήμα το επεισοδίου:
                A ← μια πράξη η οποία δόθηκε από την πολιτική π για μια κατάσταση S
                Πραγματοποίηση πράξης A, παρατήρηση R, S'
                V(s) ← V(s) + a[Ra + γV(s') − V(s)]
                S ← S'
        Μέχρις ότου S είναι τερματική κατάσταση
```

Σχήμα 9: Αλγόριθμος αξιολόγησης πολιτικής Χρονικών Διαφορών TD(0)

2.3.5.3 Ίχνη Επιλεξιμότητας

Τα ίχνη επιλεξιμότητας (eligibility traces) σε συνδυασμό με τα σφάλματα χρονικών διαφορών, παρέχουν έναν αποδοτικό κι επαυξητικό τρόπο μεταβολής των χαρακτηριστικών των μεθόδων ενισχυτικής μάθησης, ώστε αυτά να μπορούν να κλιμακωθούν και να καλύπτουν ολόκληρο το εύρος της φιλοσοφίας για τη διενέργεια των ενημερώσεων των εκτιμήσεων των συναρτήσεων αξίας, από την ενημέρωση επεισόδιο προς επεισόδιο, των μεθόδων Monte Carlo έως τη βήμα προς βήμα ενημέρωση των μεθόδων μάθησης χρονικών διαφορών. Αναδιατυπώνοντάς συνοπτικά το παραπάνω, ενσωματώνοντας ίχνη επιλεξιμότητας σε μεθόδους μάθησης χρονικών διαφορών, τους αποδίδονται χαρακτηριστικά των μεθόδων Monte Carlo. Έτσι επιτυγχάνεται η διατήρηση των πλεονεκτημάτων τους, συνδυάζοντάς τα παράλληλα με την ανοχή που επιδεικνύουν οι μέθοδοι Monte Carlo, σε περιπτώσεις όπου η διαδικασία προς μάθηση δεν είναι πλήρως Markov ή χαρακτηρίζεται από μακροπρόθεσμα καθυστερούμενες ανταμοιβές.

Ο παρών τρόπος ανάλυσης παρουσιάζει τα ίχνη επιλεξιμότητας σαν μια γέφυρα ανάμεσα στις Monte Carlo και μεθόδους Χρονικών Διαφορών. Ονομάζουμε στόχο(επιστροφή) **n** βημάτων την έκφραση

$$G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + ... + \gamma^{n-1} R_{t+n} + \gamma^n V(S_{t+n}) \tag{2.23}$$

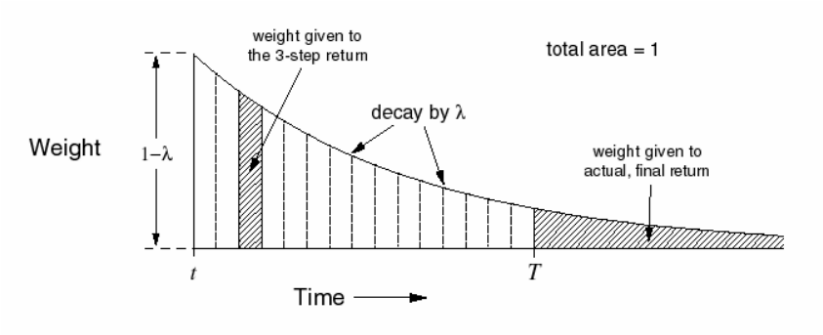
και Μέθοδο Χρονικών Διαφορών **n** την ακόλουθη σχέση

$$V(S_t) \leftarrow V(S_t) + \alpha(G_t^{(n)} - V(S_t)) \tag{2.24}$$

Εν συνεχεία, ορίζουμε την έννοια της λ-επιστροφής(λ-return), η οποία συνδυάζει όλες τις επιστροφές των n-βημάτων. Πιο συγκεκριμένα, πρόκειται για το βεβαρημένο μέσο όρο όλων των στόχων n – βημάτων

$$G_t^{\lambda} = (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)} \tag{2.25}$$

Είναι εύκολο να δούμε ότι $G_t^1 = G_t$, $G_t^0 = G_t^{(0)}$, δηλαδή μια μέθοδος με λ-επιστροφή ,περιλαμβάνει όλο το φάσμα των μεθόδων από τις Monte Carlo μέχρι τις Μεθόδους Χρονικών Διαφορών, δίνοντας βάρη στους στόχους διαφόρων βημάτων, όπως φαίνεται στο Σχήμα 10



Σχήμα 10: Συνάρτηση Βάρους για την μέθοδο TD(λ)

2.3.5.4 Αξιολόγηση πολιτικής Χρονικών Διαφορών TD(λ)

Η αξιολόγηση πολιτικής TD(λ) χρησιμοποιεί την λογική των ιχνών επιλεξιμότητας, που αναφέρθηκε στην προηγούμενη ενότητα. Θα συμβολίσουμε το ίχνος για την κατάσταση s, τη χρονική στιγμή t, με $e_t(s) \in \mathbb{R}^+$. Σε κάθε χρονικό βήμα, όλα τα ίχνη επιλεξιμότητας για όλες τις καταστάσεις φθίνουν κατά παράγοντα γλ, εκτός του ίχνους για την κατάσταση που μόλις επισκέφθηκε ο πράκτορας, το οποίο αυξάνεται κατά 1:

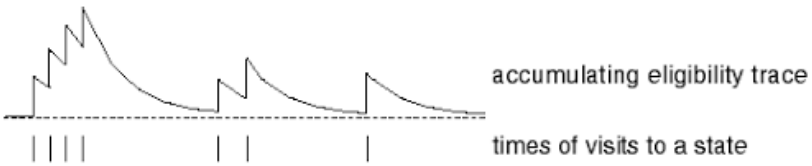
$$e_t(s) = \begin{cases} \gamma\lambda e_{t-1}(s) & , s \neq s_1 \\ \\ \gamma\lambda e_{t-1}(s) + 1 & , s = s_1 \end{cases} \tag{2.26}$$

για όλες τις καταστάσεις $s \in \mathcal{S}$ όπου $\lambda \in [0, 1]$ (Σχήμα 11). Όπως είναι φανερό, η συμπεριφορά πλησιάζει προς τις μεθόδους Monte Carlo όσο το λ πλησιάζει το 1 και το αντίστροφο. Τα ίχνη επιλεξιμότητας καταγράφουν κάθε χρονική στιγμή, ποιες καταστάσεις έχει επισκεφθεί ο πράκτορας πρόσφατα (σε όρους γλ).

Το πνεύμα της αξιολόγησης TD(λ) είναι ότι η τιμή μιας κατάστασης ανανεώνεται ανάλογα με το πόσο πρόσφατα την επισκέφθηκε ο πράκτορας. Η ανανέωση των τιμών είναι

$$V_{t+1}(s_t) = V_t(s_t) + \alpha[R_{S_{t+1}} + \gamma V_t(s_{t+1}) - V_t(s_t)]e_t(s_t) \tag{2.27}$$

Μπορεί να αποδειχθεί ότι ο παρακάτω αλγόριθμος αξιολογεί μια πολιτική με επιστροφή το G_t^{λ}.Ο αλγόριθμος Αξιολόγησης πολιτικής Χρονικών Διαφορών TD(λ), παρουσιάζεται στο Σχήμα 12.



Σχήμα 11: Ίχνη επιλεξιμότητας

Αρχικοποίηση:
V ← αυθαίρετη συνάρτηση τιμής
e(s) = 0, ∀ s ∈ S
Επανάλαβε για κάθε επεισόδιο:
Αρχικοποίηση S
Επανάλαβε για κάθε βήμα το επεισοδίου:
A ← μια πράξη η οποία δόθηκε από την πολιτική π για μια κατάσταση S
Πραγματοποίηση πράξης A, παρατήρηση R, S'
$\delta \leftarrow R + \gamma V(s') - V(s)$
e(s) ← e(s) + 1
Για όλα τα S:
$V(s) \leftarrow V(s) + \alpha\delta e(s)$
$e(s) \leftarrow \gamma\lambda e(s)$
S ← S'
Μέχρις ότου S είναι τερματική κατάσταση

Σχήμα 12: Αλγόριθμος αξιολόγησης πολιτικής Χρονικών Διαφορών TD(λ)

2.3.6 Έλεγχος άνευ μοντέλου

Όπως αναλύθηκε προηγουμένως,η πρόβλεψη άνευ μοντέλου προσεγγίζει την συνάρτηση τιμών για άγνωστη Μαρκοβιανή διαδικασία απόφασης. Στο παρόν κεφάλαιο, θα μελετήσουμε τον *Έλεγχο άνευ μοντέλου*, ο οποίος έχει ως στόχο την βελτιστοποίηση της συνάρτησης τιμών, όταν η Μαρκοβιανή διαδικασία απόφασης είναι άγνωστη ή υπερβολικά μεγάλη για να χρησιμοποιηθεί. Αντίθετα, χρησιμοποιεί δείγματα από εμπειρίες που έχουν συγκεντρωθεί κατά την εξερεύνηση του περιβάλλοντος.

Ο μόνος γενικός τρόπος για να διασφαλιστεί ότι όλες οι ενέργειες είναι επιλεγμένες απείρως συχνά, είναι ο πράκτορας να συνεχίσει να τις επιλέγει. Υπάρχουν δύο προσεγγίσεις για να διασφαλιστεί αυτό, η μέθοδος on - policy και μέθοδος off - policy.

- **On - policy**: Πρόκειται για τρόπο εκμάθησης της πολιτικής π με εμπειρίες που συγκεντρώθηκαν από την ίδια πολιτική π
- **Off - policy**: Πρόκειται για τρόπο εκμάθησης της πολιτικής π με εμπειρίες που συγκεντρώθηκαν από διαφορετική πολιτική μ

2.3.6.1 Έλεγχος Monte Carlo

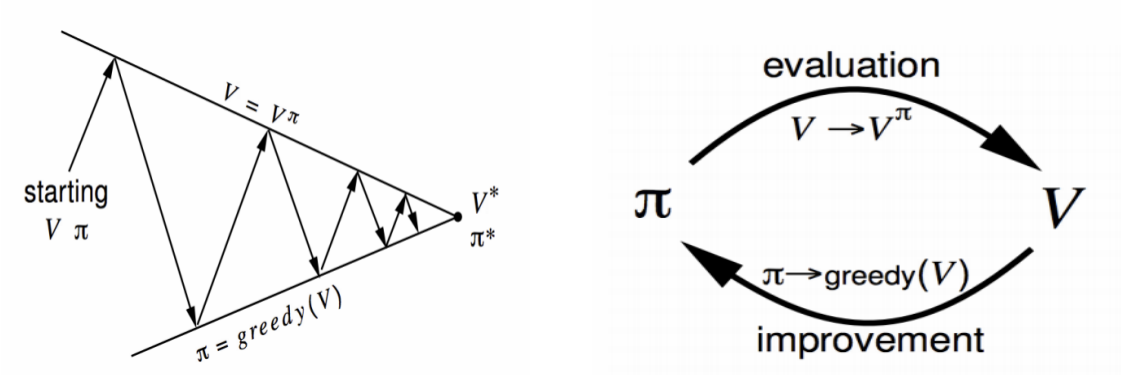
Είμαστε πλέον έτοιμοι, να εξετάσουμε, τον τρόπο με τον οποίο η εκτίμηση Monte Carlo, μπορεί να χρησιμοποιηθεί στον έλεγχο, για την προσέγγιση βέλτιστων πολιτικών. Η γενική ιδέα, είναι να εργαστούμε με τον ίδιο τρόπο, όπως στο κεφάλαιο του Δυναμικού Προγραμματισμού, δηλαδή σύμφωνα με την ιδέα της γενικευμένης επανάληψης με βάση την πολιτική (generalized policy iteration ή GPI). Στην GPI (Σχήμα 14), διατηρούμε μια προσέγγιση της πολιτικής και μια προσέγγιση της συνάρτησης τιμών. Η συνάρτηση τιμών τροποποιείται επανειλημμένα για να προσεγγίσει περισσότερο την συνάρτηση τιμών της τρέχουσας πολιτικής και η πολιτική βελτιώνεται σε σχέση με την τρέχουσα συνάρτηση τιμών.

Αυτά τα δύο είδη αλλαγών λειτουργούν ενάντια το ένα στο άλλο, σε κάποιο βαθμό, καθώς το καθένα δημιουργεί έναν κινούμενο στόχο για τον άλλο, αλλά μαζί προκαλούν την σύγκλιση των δύο συναρτήσεων ως προς τις βέλτιστες.[32]

Βελτιστοποίηση της πολιτικής επιτυγχάνεται κάνοντας την, άπληστη σε σχέση με την τρέχουσα συνάρτηση τιμών. Σε αυτή την περίπτωση έχουμε μια συνάρτηση πράξης - αξίας και συνεπώς δεν είναι απαραίτητη η ύπαρξη μοντέλου για την οικοδόμηση της. Για κάθε λειτουργία πράξης - αξίας **Q**, η αντίστοιχη άπληστη πολιτική είναι αυτή που αποφασίζει μια δράση με στόχο την μεγιστοποίηση της τιμής **Q**

$$\pi(s) = \arg\max_{\alpha} Q(s, \alpha) \tag{2.28}$$

Αποδεικνύεται, ότι ο έλεγχος Monte Carlo συγκλίνει στην βέλτιστη πολιτική, δοθέντων μόνο δειγμάτων επεισοδίων χωρίς περαιτέρω γνώση του περιβάλλοντος.



(a) (b)

Σχήμα 14: Σχηματική αναπαράσταση της γενικευμένης επανάληψης με βάση την πολιτική (GPI), με δύο διαφορετικούς τρόπους

On - Policy Έλεγχος Monte Carlo

Υπάρχουν πολλές πιθανές παραλλαγές στις on - policy μεθόδους. Μια πιθανή υλοποίηση, είναι να μετατοπίζουμε σταδιακά την πολιτική προς μια θεωρητική βέλτιστη. Η on-policy μέθοδος που παρουσιάζουμε σε αυτήν την ενότητα χρησιμοποιεί **ε**-άπληστη πολιτική, πράγμα που σημαίνει ότι τις περισσότερες φορές θα επιλεγεί μια ενέργεια που έχει μέγιστη εκτιμώμενη αξία, αλλά με πιθανότητα **ε** να επιλέξει μια τυχαία ενέργεια. Με αυτό τον τρόπο, όλες οι μη άπληστες ενέργειες δίνουν την ελάχιστη πιθανότητα επιλογής, $\frac{\varepsilon}{|A(s)|}$, και το υπόλοιπο, μεγαλύτερο μέρος του $1-\varepsilon+\frac{\varepsilon}{|A(s)|}$, δίνεται στην άπληστη πράξη. Οι ε-άπληστες πολιτικές, είναι παραδείγματα πολιτικών ορισμένων σαν πολιτικές για τις οποίες $\pi(s, \alpha) \geq \frac{\varepsilon}{|A(s)|}$, για όλες τις καταστάσεις και τις δράσεις. Στο Σχήμα 15 παρουσιάζεται αναλυτικά ο αλγόριθμος.

Off - Policy Έλεγχος Monte Carlo

Στις μεθόδους off - policy, υπάρχουν δύο διαχωρισμένες λειτουργίες. Η πρώτη είναι η πολιτική που χρησιμοποιείται για τη δημιουργία συμπεριφοράς, η οποία ονομάζεται πολιτική συμπεριφοράς. Αντίθετα, η δεύτερη πολιτική είναι η πολιτική που αξιολογείται και βελτιώνεται, η οποία ονομάζεται πολιτική εκτίμησης. Ένα πλεονέκτημα αυτού του διαχωρισμού είναι ότι η πολιτική εκτίμησης μπορεί να είναι ντετερμινιστική, π.χ. άπληστη, ενώ η πολιτική συμπεριφοράς μπορεί να συνεχίσει να δοκιμάζει όλες τις πιθανές ενέργειες.

Οι off - policy μέθοδοι ελέγχου Monte Carlo, χρησιμοποιούν την τεχνική εκτίμησης της συνάρτησης τιμών για μια πολιτική, ενώ ακολουθούν μια άλλη. Λαμβάνουν αποφάσεις, δηλαδή, με την πολιτική συμπεριφοράς, ενώ παράλληλα τροποποιούν και βελτιώνουν την πολιτική εκτίμησης. Η τεχνική αυτή, προϋποθέτει ότι η πολιτική συμπεριφοράς έχει την πιθανότητα να διαλέγει όλες τις πράξεις που μπορούν να επιλεγούν από την πολιτική εκτίμησης. Στο Σχήμα 16 παρουσιάζεται αναλυτικά ο αλγόριθμος.

Αρχικοποίηση:
π ← αυθαίρετη πολιτική
Q(s,α) ∈ ℝ , αυθαίρετο ∀ s ∈ S, α ∈ A(s)
Επιστροφές (s,α) ← Άδεια λίστα ∀ s ∈ S, α ∈ A(s)
Επανάλαβε για κάθε επεισόδιο:
παραγωγή ενός επεισοδίου χρησιμοποιώντας την πολιτική π
G ← 0
Επανάλαβε για κάθε βήμα το επεισοδίου:
$G \leftarrow G + R_{t+1}$
Μέχρις ότου S_t , A_t εμφανιστεί στο επεισόδιο:
Προσθήκη G στις Επιστροφές (S_t , A_t)
Q(S_t , A_t) ← Μέση τιμή(Επιστροφών (S_t , A_t))
$A^* = \arg\max_\alpha Q(S_t, \alpha)$
Για κάθε α ∈ A(S_t):

$$\pi(\alpha|S_t) \leftarrow \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{|A(S_t)|}, & a = A^* \\ \frac{\varepsilon}{|A(S_t)|}, & a \neq A^* \end{cases}$$

Σχήμα 15: On-policy αλγόριθμος ελέγχου πολιτικής Monte Carlo

Αρχικοποίηση:
Q(s,α) ∈ ℝ , αυθαίρετο ∀ s ∈ S, α ∈ A(s)
C(s,α) ← 0
$\pi(s) \leftarrow \arg\max_\alpha Q(s, \alpha)$
Επανάλαβε για κάθε επεισόδιο:
b ← οποιαδήποτε πολιτική
παραγωγή ενός επεισοδίου χρησιμοποιώντας την πολιτική b
G ← 0
W ← 1
Επανάλαβε για κάθε βήμα το επεισοδίου:
$G \leftarrow \gamma G + R_{t+1}$
$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(St,At)}[G - Q(St, At)]$
$\pi(S_t) \leftarrow \arg\max_\alpha Q(S_t, \alpha)$
Αν $A_t \neq \pi(S_t)$ έξοδος από την επανάληψη
$W \leftarrow W\frac{1}{b(A_t|S_t)}$

Σχήμα 16: Off-policy αλγόριθμος ελέγχου πολιτικής Monte Carlo

2.3.6.2 Έλεγχος Χρονικών Διαφορών TD

Ας υποθέσουμε, ότι μόνο ένας πεπερασμένος αριθμός από εμπειρίες είναι διαθέσιμος. Σε αυτή την περίπτωση, μια συνηθισμένη προσέγγιση του προβλήματος είναι η επανάληψη των ίδιων εμπειριών κατά την εκπαίδευση, έως ότου η μέθοδος να συγκλίνει. Δεδομένης μιας προσεγγιστικής συνάρτησης τιμών **V**, οι ανανεώσεις,

$$V(s_t) \leftarrow V(s_t) + \alpha[G_t - V(s_t)], V(s_t) \leftarrow V(s_t) + \alpha[R_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \quad (2.29)$$

υπολογίζονται κάθε χρονική στιγμή **t**, κατά την οποία ο πράκτορας επισκέπτεται μια μη τερματική κατάσταση. Η συνάρτηση τιμών, όμως, αλλάζει μόνο μια φορά με το άθροισμα όλων των αλλαγών. Στην συνέχεια, όλη η διαθέσιμη εμπειρία επεξεργάζεται ξανά με μια καινούργια συνάρτηση τιμών

για την δημιουργία νέων αξιολογήσεων. Η διαδικασία αυτή, συνεχίζεται μέχρις ότου η συνάρτηση να συγκλίνει. Η ορολογία της διαδικασίας αυτής, ονομάζεται *ενημέρωση παρτίδας*. Υπό την διαδικασία αυτή, ο έλεγχος χρονικών διαφορών TD(0) συγκλίνει ντετερμινιστικά σε μία μόνο λύση ανεξάρτητη της παραμέτρου α, αρκεί να είναι ικανοποιητικά μικρή.

Για τον έλεγχο TD ακολουθείται η διαδικασία της γενικευμένης επανάληψης με βάση την πολιτική (GPI), όπως και στον έλεγχο Monte Carlo. Και στην συγκεκριμένη περίπτωση, εμφανίζεται η πρόκληση μεταξύ εξερεύνησης και εκμετάλλευσης της υπάρχουσας πολιτικής. Η λογική είναι όμοια και χωρίζεται ξανά σε δύο κατηγορίες.

On - Policy Έλεγχος TD: Sarsa

Το πρώτο βήμα, είναι η εκμάθηση μιας συνάρτησης πράξης - αξίας αντί για την συνάρτηση κατάστασης - αξίας V(s). Στην συγκεκριμένη περίπτωση για μια on - policy μέθοδο, πρέπει να προσεγγίσουμε την $Q^{\pi}(s, \alpha)$ για την συγκεκριμένη πολιτική **π** και για όλες τις καταστάσεις **s** και δράσεις **α**. Το θετικό είναι, πως η Q^{π} χρησιμοποιεί την ίδια TD μέθοδο, όπως περιγράφηκε παραπάνω για την εκμάθηση V^{π}.

Στην προηγούμενη ενότητα εξετάσαμε τις μεταβάσεις από μία κατάσταση σε μια άλλη και μάθαμε την αξία των καταστάσεων αυτών. Αλλά η σχέση μεταξύ των καταστάσεων και των ζευγών καταστάσεων - πράξεων, είναι συμμετρική. Τώρα σκεφτείτε τις μεταβάσεις από ένα ζεύγος κατάστασης - πράξης σε ένα άλλο και να πρέπει να μάθει την αξία του ζεύγους αυτού. Επίσημα αυτές οι περιπτώσεις είναι ίδιες: είναι και οι δύο αλυσίδες Markov με διαδικασία ανταμοιβής. Τα θεωρήματα που εξασφαλίζουν τη σύγκλιση των τιμών κατάστασης υπό Έλεγχος Χρονικών Διαφορών (0) ισχύουν επίσης για το αντίστοιχο αλγόριθμο για τιμές δράσης:

$$Q(S_t, a_t) \leftarrow Q(S_t, a_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, a_{t+1}) - Q(S_t, a_t)] \tag{2.30}$$

Η ανανέωση αυτή γίνεται μετά από κάθε μετάβαση σε μη τερματική κατάσταση. Αν η κατάσταση s είναι τερματική, τότε η Q θεωρείται ίση με μηδέν. Η μέθοδος αυτή, χρησιμοποιεί κάθε στοιχείο από σύνολο των εμπειριών $(s_t, \alpha_t, s_{t+1}, \alpha_{t,+1}$. Αυτός είναι ο λόγος που ο συγκεκριμένος αλγόριθμος πήρε το όνομα Sarsa. Στο Σχήμα 18 παρουσιάζεται αναλυτικά ο αλγόριθμος για TD(0) και TD(λ).

Off - Policy Έλεγχος TD: Q - learning

Μια από τις μεγαλύτερες ανακαλύψεις στην ενισχυτική μάθηση, αφορά έναν off - policy αλγόριθμο ελέγχου ο οποίος ονομάζεται Q - learing. Η πιο απλή του μορφή ονομάζεται Q - learning ενός βήματος και ορίζεται ως

$$Q(S_t, a_t) \leftarrow Q(S_t, a_t) + \alpha[R_{t+1} + \gamma \max_{\alpha'} Q(S_{t+1}, \alpha') - Q(S_t, a_t)] \tag{2.31}$$

Σε αυτή την περίπτωση, η εκπαιδευμένη συνάρτηση πράξης - αξίας Q, προσεγγίζει κατευθείαν την ιδανική συνάρτηση Q^*, ανεξαρτήτως της πολιτικής που ακολουθείται. Αυτό, απλοποιεί δραματικά την ανάλυση του αλγορίθμου. Η πολιτική αυτή, βέβαια, εξακολουθεί να επηρεάζει με τις αποφάσεις της το ζεύγος κατάστασης - πράξης που θα δεχθεί επίσκεψη και θα ανανεωθεί. Παρόλα αυτά, όλα τα προαπαιτούμενα για ορθή σύγκλιση, είναι όλα τα ζευγάρια να συνεχίζουν να ανανεώνονται.[38] Ο αλγόριθμος Q - learning παρουσιάζεται στο Σχήμα 19.

Αρχικοποίηση:
Q(s,α) ∈ ℝ , αυθαίρετο ∀ s ∈ S, α ∈ A(s) και Q(τελική κατάσταση,▪) = 0
Επανάλαβε για κάθε επεισόδιο:
Αρχικοποίηση των S
Επιλογή πράξης A σε μια κατάσταση S μέσω της ε - άπληστης πολιτικής από την Q
Επανάλαβε για κάθε βήμα το επεισοδίου:
Πραγματοποίηση πράξης A, παρατήρηση R, S'
Επιλογή A' από την S' χρησιμοποιώντας την ε – άπληστη πολιτική
$Q(s, \alpha) \leftarrow Q(s, \alpha) + \alpha[R + \gamma Q(s', \alpha') - Q(s, \alpha)]$
S ← S'
A ← A'
μέχρις ότου η κατάσταση S να είναι τερματική

(a) SARSA

Αρχικοποίηση:
Q(s,α) ∈ ℝ , αυθαίρετο ∀ s ∈ S, α ∈ A(s)
Επανάλαβε για κάθε επεισόδιο:
E(s,α) = 0 , ∀ s ∈ S, α ∈ A(s)
Επανάλαβε για κάθε βήμα το επεισοδίου:
Πραγματοποίηση πράξης A, παρατήρηση R, S'
Επιλογή A' από την S' χρησιμοποιώντας την ε – άπληστη πολιτική
$\delta \leftarrow R + \gamma Q(s', \alpha') - Q(s, a)$
$E(s,\alpha) \leftarrow E(s,\alpha) + 1$
Για κάθε s ∈ S, α ∈ A(s):
$Q(s, a) \leftarrow Q(s, a) + \alpha\delta E(s, \alpha)$
$E(s,\alpha) \leftarrow \gamma\lambda E(s,\alpha)$
S ← S'
A ← A'
μέχρις ότου η κατάσταση S να είναι τερματική

(b) SARSA-λ

Σχήμα 18: Αλγόριθμοι on-policy ελέγχου χρονικών διαφορών

Αρχικοποίηση:
Q(s,α) ∈ ℝ , αυθαίρετο ∀ s ∈ S, α ∈ A(s) και Q(τελική κατάσταση,▪) = 0
Επανάλαβε για κάθε επεισόδιο:
Αρχικοποίηση των S
Επανάλαβε για κάθε βήμα το επεισοδίου:
Επιλογή πράξης A σε μια κατάσταση S από την Q(ε - άπληστης πολιτικής)
Πραγματοποίηση πράξης A, παρατήρηση R, S'
$Q(s, \alpha) \leftarrow Q(s, \alpha) + \alpha[R + \gamma \max_{\alpha} Q(s', \alpha) - Q(s, \alpha)]$
S ← S'
μέχρις ότου η κατάσταση S να είναι τερματική

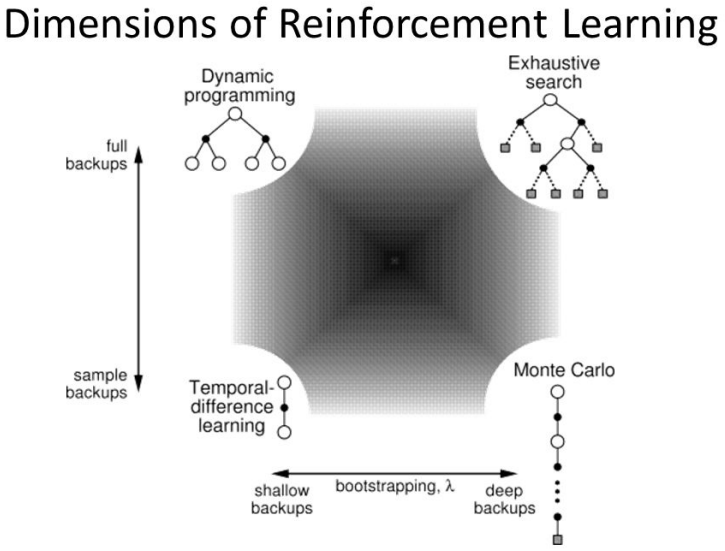
Σχήμα 19: Off-policy αλγόριθμος ελέγχου πολιτικής Q learning

Κεφάλαιο 3

Βαθιά Ενισχυτική Μάθηση

3.1 Μηχανισμοί προσέγγισης συναρτήσεων τιμών

Τα συστήματα ενισχυτικής μάθησης είναι καλό να έχουν δυνατότητες γενίκευσης (generalization), ειδικά σε περιπτώσεις που χρησιμοποιούνται για τη δημιουργία τεχνητής νοημοσύνης σε εφαρμογές μεγάλης κλίμακας. Για να επιτευχθεί αυτό, συνήθως χρησιμοποιούνται μέθοδοι επιβλεπόμενης μάθησης (supervised learning) για προσέγγιση συναρτήσεων (function approximation), θεωρώντας κάθε ανάστροφη ενημέρωση (backup) για την συνάρτηση αξίας, ως παράδειγμα εκπαίδευσης. Συγκεκριμένα, οι μέθοδοι βαθμωτής καθόδου κατά την κλίση (gradient descent) επιτρέπουν τη φυσική επέκταση με δυνατότητες προσέγγισης συναρτήσεων, των τεχνικών που συζητήθηκαν στις προηγούμενες ενότητες. [39]



Σχήμα 20: Ενοποιημένη επισκόπηση των μεθόδων Ενισχυτικής Μάθησης

Ειδικά για την περίπτωση των μεθόδων γραμμικής καθόδου κατά την κλίση (linear gradient descent), υπάρχει μεγάλο θεωρητικό ενδιαφέρον, ενώ αποδίδουν καλά και στην πράξη, εφόσον τροφοδοτούνται με τα κατάλληλα χαρακτηριστικά κατάστασης. Η επιλογή των κατάλληλων χαρακτηριστικών κατάστασης είναι κρίσιμης σημασίας και αποτελεί έναν σημαντικό τρόπο προσθήκης πρότερης γνώσης σε συστήματα ενισχυτικής μάθησης. Οι μέθοδοι γραμμικής καθόδου κατά την κλίση, περιλαμβάνουν μεταξύ άλλων τις συναρτήσεις ακτινικής βάσης (radial basis functions), την κωδικοποίηση πλακιδίων (tile coding) και την κωδικοποίηση Kanerva (Kanerva coding). Αρκετά διαδεδομένες είναι κι οι μέθοδοι ανάστροφης διάδοσης σφάλματος με χρήση νευρωνικών δικτύων, οι οποίες παρουσιάζουν πάρα πολύ καλές επιδόσεις σε συγκεκριμένες εφαρμογές όπως το TD-Gammon [40], ωστόσο παρουσιάζουν προβλήματα σε άλλα κλασικά πεδία δοκιμών όπως το αυτοκίνητο πλα-

γιάς (Mountain Car). Επιπλέον οι θεωρητικές εγγυήσεις που παρέχουν για σύγκλιση σε πολιτικές κοντά στην βέλτιστη, είναι πιο αδύναμες σε σχέση με αυτές των γραμμικών μεθόδων. Μάλιστα, παρουσιάζουν το φαινόμενο να «ξεχνούν» πράγματα που έχουν ήδη μάθει (unlearning past experience).

Στην παρούσα ανάλυση θα χρησιμοποιήσουμε τον αλγόριθμο DQN που αναπτύχθηκε από την DeepMind και αποτέλεσε σημαντικό βήμα για την χρήση νευρωνικών δικτύων στην ενισχυτική μάθηση.[41]

3.2 Μέθοδος DQN

Η εκμάθηση πρακτόρων, απευθείας από υψηλών διαστάσεων αισθητήριες εισόδους, όπως όραση και φωνή, αποτελεί μια από τις προκλήσεις της ενισχυτικής μάθησης. Αυτό συμβαίνει, διότι η ενισχυτική μάθηση παρουσιάζει αρκετές δυσκολίες από την σκοπιά της βαθιάς μάθησης. Καταρχήν, η εκμάθηση γίνεται μέσα από βαθμωτές επιβραβεύσεις οι οποίες είναι σποραδικές, θορυβώδεις και με καθυστέρηση. Επιπλέον, υπάρχει μεγάλη συσχέτιση των δεδομένων στην ενισχυτική μάθηση, καθώς τα γεγονότα που συμβαίνουν είναι διαδοχικά. Τέλος, η κατανομή των δεδομένων αλλάζει καθώς ο αλγόριθμος μαθαίνει νέες συμπεριφορές, το οποίο από την πλευρά της βαθιάς μάθησης αποτελεί πρόβλημα, καθώς υποθέτει σταθερή κατανομή.

Ο συγκεκριμένος αλγόριθμος, επιτυγχάνει να ξεπεράσει τις παραπάνω δυσκολίες και να εκπαιδευτεί με πολιτικές ελέγχου από ακατέργαστα δεδομένα με χρήση νευρωνικών δικτύων. Το δίκτυο εκπαιδεύτηκε με μια παραλλαγή της μεθόδου Q - learning με χρήση της στοχαστικής καθόδου κατά την κλίση μεθόδου για την ανανέωση των βαρών του νευρωνικού δικτύου. Επιπλέον για να αντιμετωπιστεί το πρόβλημα των συσχετιζόμενων δεδομένων, χρησιμοποιήθηκε ένας μηχανισμός αναπαραγωγής εμπειρίας [42], ο οποίος επιλέγει τυχαία δείγματα από παρελθοντικές μεταβάσεις.

Όπως σε κάθε πρόβλημα ενισχυτικής μάθησης, έχουμε έναν πράκτορα, ο οποίος αλληλεπιδρά με το περιβάλλον του μέσα από ένα σύνολο από δυνατές πράξεις $\mathcal{A}$. Ως στόχο, ορίζουμε την επιλογή δράσεων, τέτοιων ώστε να μεγιστοποιείται η συνολική μελλοντική ανταμοιβή. Υποθέτουμε, πως οι μελλοντικές ανταμοιβές μειώνονται με έναν εκπτωτικό όρο γ ανά χρονικό βήμα και συνεπώς η συνολική επιστροφή ορίζεται ως $G_t = \sum_{t'=t}^{T} \gamma^{t'-t} R_{t'}$, όπου T είναι ο αριθμός των χρονικών βημάτων κατά τον οποίο το επεισόδιο τερματίζει. Ορίζουμε ως βέλτιστη συνάρτηση πράξης - αξίας $Q^*(s, \alpha)$ ως την μέγιστη αναμενόμενη επιστροφή που μπορεί να επιτευχθεί, ακολουθώντας οποιαδήποτε στρατηγική.

$$Q^*(s, \alpha) = \max_{\pi} \mathbb{E}[G_t | s_t = s, \alpha_t = \alpha, \pi] \tag{3.1}$$

Η βέλτιστη συνάρτηση πράξης - αξίας, υπακούει την εξίσωση του Bellman. Η συγκεκριμένη εξίσωση μπορεί διαισθητικά να εξηγηθεί ως εξής: αν μια βέλτιστη τιμή $Q^*(s^{'}, \alpha^{'})$ για την κατάσταση $s^{'}$ στην επόμενη χρονική στιγμή, ήταν γνωστή για όλες τις πιθανές δράσεις $\alpha^{'}$, τότε η βέλτιστη στρατηγική είναι η επιλογή της πράξης $\alpha^{'}$, η οποία μεγιστοποιεί την αναμενόμενη τιμή της έκφρασης $R + \gamma Q^*(s^{'}, \alpha^{'})$.

$$Q^*(s, \alpha) = \mathbb{E}_{s' \sim \mathcal{E}}[R + \gamma \max_{\alpha'} Q^*(s^{'}, \alpha^{'}) | s, \alpha] \tag{3.2}$$

Η βασική ιδέα πίσω από πολλούς αλγορίθμους ενισχυτικής μάθησης είναι η προσέγγιση της συνάρτησης πράξης - αξίας χρησιμοποιώντας την εξίσωση Bellman ως μια επαναληπτική ανανέωση, $Q^*(s, \alpha) = \mathbb{E}[R + \gamma \max_{\alpha'} Q^*(s^{'}, \alpha^{'}) | s, \alpha]$. Τέτοιοι αλγόριθμοι συγκλίνουν στην βέλτιστη συνάρτηση Q.

$$Q_i \rightarrow Q^* \; \textit{καθώς} \; i \rightarrow \infty \tag{3.3}$$

Στην πράξη, η προσέγγιση αυτή δεν είναι καθόλου αποτελεσματική, καθώς η συνάρτηση πράξης - αξίας , υπολογίζεται ξεχωριστά για κάθε κατάσταση, χωρίς καμία γενίκευση. Αντιθέτως, είναι συνηθισμένη η χρήση προσεγγιστικών συναρτήσεων τιμών για την εκτίμηση της συνάρτησης Q.

$$Q(s, \alpha; \theta) \approx Q^*(s, \alpha) \tag{3.4}$$

Ως προσεγγιστική συνάρτηση τιμών ορίζουμε ένα νευρωνικό δίκτυο με βάρη θ. Το δίκτυο αυτό, το ονομάζουμε Q - δίκτυο. Στην προκείμενη περίπτωση, η εκπαίδευση μπορεί να επιτευχθεί με την ελαχιστοποίηση των loss functions $L_i(\theta_i)$, οι οποίες αλλάζουν με κάθε επανάληψη i.

$$L_i(\theta_i) = \mathbb{E}_{s,a\sim\rho(.)}[(y_i - Q(s,a;\theta))^2] \tag{3.5}$$

όπου $y_i = \mathbb{E}_{s'\sim\mathcal{E}}[R + \gamma \max_{a'} Q^*(s', a'; \theta_{i-1})|s, a]$ είναι ο στόχος της επανάληψης i και $\rho(s,a)$ είναι μια κατανομή πιθανότητας συναρτήσει των καταστάσεων και των πράξεων, την οποία ονομάζουμε κατανομή συμπεριφοράς. Οι παράμετροι από τις προηγούμενες επαναλήψεις, παραμένουν σταθερές καθώς βελτιστοποιούμε την loss function.

Σημειώνεται, ότι ο αλγόριθμος είναι model - free. Επιλύει ,δηλαδή, την πρόκληση της ενισχυτικής μάθησης απευθείας χρησιμοποιώντας δείγματα από το περιβάλλον $\mathcal{E}$, χωρίς να κατασκευάζει μια εκτίμηση αυτού. Επιπλέον, πρόκειται για έναν αλγόριθμο off-policy. Μαθαίνει με μια άπληστη στρατηγική $\alpha = \max_{\alpha} Q(s,a;\theta)$, ενώ ακολουθεί μια κατανομή συμπεριφοράς η οποία εξασφαλίζει ικανοποιητική εξερεύνηση του χώρου κατάστασης. Στην πράξη, η στρατηγική είναι ε-άπληστη και ακολουθεί την άπληστη στρατηγική με πιθανότητα $1 - \varepsilon$ και επιλέγει κάποια τυχαία πράξη με πιθανότητα ε.

Τέλος, όπως και στον αλγόριθμο TD-Gammon και στην τεχνική DQN χρησιμοποιούμε την τεχνική της *αναπαραγωγής εμπειριών*, κατά την οποία, αποθηκεύουμε τις εμπειρίες του πράκτορα, για κάθε χρονικό βήμα σε ένα σύνολο δεδομένων που το ορίζουμε ως $\mathcal{D}$. Κατά την διάρκεια εκτέλεσης του αλγορίθμου, πραγματοποιούμε ανανεώσεις των βαρών του δικτύου με χρήση τυχαίων δειγμάτων από το σύνολο $\mathcal{D}$ της μνήμης. Ο αλγόριθμος παρουσιάζεται αναλυτικά στο Σχήμα 21.

Η συγκεκριμένη προσέγγιση, έχει αρκετά πλεονεκτήματα σε σχέση με την κλασική μέθοδο του online Q-learning. Καταρχήν, κάθε βήμα των εμπειριών χρησιμοποιείται, ενδεχομένως, σε αρκετές ανανεώσεις των βαρών, με αποτέλεσμα να επιτρέπει μεγαλύτερη αποδοτικότητα των δεδομένων. Επιπλέον, η εκμάθηση απευθείας από συνεχόμενα δείγματα δεν είναι αποτελεσματική, εξαιτίας της ισχυρής συσχέτισης μεταξύ τους. Συνεπώς, με την χρήση της αναπαραγωγής εμπειριών επιτυγχάνεται τυχαία χρήση δειγμάτων των δεδομένων με αποτέλεσμα να μειώνεται δραστικά η συσχέτιση αυτών. Τέλος, η εκμάθηση on-policy έχει ως αποτέλεσμα, οι συγκεκριμένες παράμετροι εκείνης της χρονικής στιγμής, να αποφασίζουν τα επόμενα δείγματα δεδομένων με τα οποία θα εκπαιδευτούν μελλοντικά οι παράμετροι.

Initialize replay memory $\mathcal{D}$ to capacity N
Initialize action-value function Q with random weights
for episode $= 1, M$ **do**
  Initialise sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**
    With probability ϵ select a random action a_t
    otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ according to equation 3
  end for
end for

Σχήμα 21: Αλγόριθμος DQN

3.3 Αναζήτηση πολιτικής

Οι μέθοδοι αναζήτησης πολιτικής, δεν χρειάζεται να διατηρούν έναν αλγόριθμο συνάρτησης τιμών, αντιθέτως ερευνούν απευθείας μια βέλτιστη πολιτική π^*. Τυπικά, επιλέγεται μια παραμετροποιημένη πολιτική π_θ, της οποίας οι παράμετροι ανανεώνονται με τέτοιο τρόπο, ώστε να μεγιστοποιούν την αναμενόμενη επιστροφή $\mathbb{E}[G_t|\theta]$, χρησιμοποιώντας είτε gradient-based είτε gradient-free βελτιστοποιήσεις.[43] Νευρωνικά δίκτυα τα οποία κωδικοποιούν πολιτικές, εκπαιδεύτηκαν με επιτυχία τόσο σε gradient-free όσο και σε gradient-based μεθόδους.

Όταν σχεδιάζουμε πολιτικές απευθείας, είναι σύνηθες οι παράμετροι εξόδου να αποτελούν κατανομές πιθανότητας. Για συνεχές πεδίο δράσεων, αυτό μπορεί να σημαίνει την μέση τιμή και την τυπική απόκλιση μιας κανονικής κατανομής (Gaussian distribution), ενώ για διακριτές πράξεις μπορεί να αναπαριστά τις ατομικές πιθανότητες μιας κατανομής. Το αποτέλεσμα σε κάθε περίπτωση είναι μια στοχαστική πολιτική από την οποία μπορούμε άμεσα να λαμβάνουμε πράξεις. Για τις gradient-free μεθόδους, η εύρεση καλύτερης πολιτικής προϋποθέτει μια αναζήτηση ανάμεσα σε μοντέλα μιας ήδη ορισμένης κλάσης. Ένα κύριο πλεονέκτημα των gradient-free πολιτικών αναζήτησης, είναι πως μπορούν να βελτιστοποιήσουν μη παραγωγήσιμες συναρτήσεις.[44]. Στην συνέχεια θα αναλύσουμε εκτενέστερα τον τρόπο με τον οποίο μοντελοποιούμε τις πολιτικές που μόλις περιγράψαμε.

3.4 Policy Gradients

Όπως ήδη γνωρίζουμε, στόχος της Ενισχυτικής μάθησης είναι η μεγιστοποίηση μιας αναμενόμενης ανταμοιβής καθώς ακολουθούμε μια πολιτική π. Όπως αναφέραμε και στην προηγούμενη ενότητα, ορίζουμε ένα σύνολο παραμέτρων θ για να παραμετροποιήσουμε την πολιτική αυτή π_θ. Συνεπώς, μπορούμε να ορίσουμε ως στόχο την μεγιστοποίηση της συνάρτησης

$$\mathcal{J}(\theta) = \mathbb{E}[(\tau)] \tag{3.6}$$

Όπως και σε άλλα προβλήματα μηχανικής μάθησης, εάν μπορέσουμε να βρούμε τις παραμέτρους θ^* οι οποίες μεγιστοποιούν την συνάρτηση $\mathcal{J}$, τότε έχουμε λύσει το πρόβλημα μας. Μια συνηθισμένη προσέγγιση για την επίλυση του παρόντος έργου, είναι να χρησιμοποιήσουμε Gradient Ascent(or Descent). Στην gradient ascent, η ανανέωση των βαρών γίνεται ως εξής:

$$\theta_{t+1} = \theta_t + \alpha \nabla \mathcal{J}(\theta_t) \tag{3.7}$$

Το ερώτημα που τίθεται σε αυτό το σημείο, είναι το πως θα βρεθεί το ανάδελτα της συνάρτησης, καθώς το ολοκλήρωμα είναι υπολογιστικά ασύμφορο. Συνεπώς πρέπει να βρεθεί μια εναλλακτική. Πρώτο βήμα είναι να αναδιατυπώσουμε το ανάδελτα της $\mathcal{J}$

$$\nabla \mathbb{E}[r(\tau)] = \nabla \int \pi(\tau) r(\tau) d\tau = \int \nabla \pi(\tau) r(\tau) d\tau = \int \pi(\tau) \nabla \log \pi(\tau) r(\tau) d\tau \tag{3.8}$$

$$\nabla \mathbb{E}[r(\tau)] = \mathbb{E}[r(\tau) \nabla \log \pi(\tau)] \tag{3.9}$$

Policy Gradient Θεώρημα

Η παράγωγος της αναμενόμενης τιμής της συνολικής ανταμοιβής (επιστροφής) ισούται με την αναμενόμενη τιμή του γινομένου της ανταμοιβής επί την παράγωγο του λογαρίθμου της πολιτικής π

$$\nabla E[r(\tau)] = \mathbb{E}[G(\tau) \nabla \log \pi(\tau)] \tag{3.10}$$

Σε αυτό το σημείο θα αναλύσουμε τον ορισμό της πολιτικής π.Συγκεκριμένα

$$\pi_\theta(\tau) = \mathcal{P}(s_0) \prod_{t=1}^{T} \pi_\theta(\alpha_t|s_t) p(s_{t+1}, R_{t+1}|s_t, \alpha_t)$$

Για να κατανοήσουμε καλύτερα τον παραπάνω τύπο, θα εξηγήσουμε κάθε όρο ξεχωριστά. $\mathcal{P}$ αντιπροσωπεύει την κατανομή πιθανότητας εκκίνησης σε κάποια κατάσταση s_0. Στην συνέχεια, εφαρμόζουμε τον κανόνα του γινομένου για κάθε επόμενη πράξη, καθώς η επιλογή κάθε δράσης είναι ανεξάρτητη από την προηγούμενη. Αυτό προκύπτει από τις Μαρκοβιανές διαδικασίες αποφάσεων, καθώς κάθε κατάσταση περιέχει όλη την απαραίτητη πληροφορία για την επιλογή της επόμενης πράξης και είναι ανεξάρτητη του παρελθόντος. Για κάθε βήμα, λαμβάνεται από τον πράκτορα μια απόφαση δράσης χρησιμοποιώντας την πολιτική π_θ και με την αλληλεπίδραση με το περιβάλλον του, οδηγείται σε μια καινούργια κατάσταση. Οι όροι αυτοί πολλαπλασιάζονται για το χρονικό διάστημα T βημάτων, το οποίο αναπαριστά το μήκος της τροχιάς. Συνεπώς, για τον λογάριθμο της πολιτικής ισχύει

$$\log \pi_\theta(\tau) = \log \mathcal{P}(s_0) + \sum_{t=1}^{T} \log \pi_\theta(\alpha_t|s_t) + \sum_{t=1}^{T} \log p(s_{t+1}, R_{t+1}|s_t, \alpha_t) \Rightarrow \tag{3.11}$$

$$\nabla \log \pi_\theta(\tau) = \sum_{t=1}^{T} \nabla \log \pi_\theta(\alpha_t|s_t) \Rightarrow \tag{3.12}$$

$$\nabla \mathbb{E}_{\pi_\theta}[r(\tau)] = \mathbb{E}_{\pi_\theta}[r(\tau)(\sum_{t=1}^{T} \nabla \log \pi_\theta(\alpha_t|s_t))] \tag{3.13}$$

Το αποτέλεσμα που καταλήξαμε είναι ιδιαιτέρως σημαντικό, διότι δεν χρειάζεται να γνωρίζουμε την κατανομή των καταστάσεων $\mathcal{P}$ ούτε τις δυναμικές του περιβάλλοντος p, οι οποίες είναι ιδιαιτέρως δύσκολο να μοντελοποιηθούν. Όλοι οι αλγόριθμοι οι οποίοι χρησιμοποιούν το συγκεκριμένο αποτέλεσμα είναι Model-free αλγόριθμοι.

Η αναμενόμενη τιμή ή ισοδύναμα ο όρος του ολοκληρώματος εξακολουθεί να αποτελεί πρόβλημα που πρέπει να αντιμετωπιστεί. Μια απλή, αλλά αποτελεσματική, προσέγγιση είναι η μέθοδος Monte Carlo που έχουμε ήδη αναλύσει. Να προχωρήσουμε, με απλά λόγια, σε δειγματοληψία ενός μεγάλου αριθμού από τροχιές, και να υπολογίσουμε την μέση τιμή αυτών. Αυτή είναι μια προσέγγιση όμοια με την εκτίμηση ενός ολοκληρώματος συνεχούς χώρου μέσω ενός συνόλου από διακριτά σημεία του χώρου αυτού.

Τέλος, δεν έχουμε ασχοληθεί καθόλου με την ανταμοιβή $G(\tau$ της εκάστοτε τροχιάς. Παρόλο που η παράγωγος της παραμετροποιημένης πολιτικής δεν εξαρτάται από την ανταμοιβή, ο όρος αυτός αυξάνει την διασπορά σε μια δειγματοληψία Monte Carlo. Συγκεκριμένα υπάρχουν T σε αριθμό πηγές που δημιουργούν αύξηση στην διασπορά με την κάθε μία να συνεισφέρει κατά R_t. Μια άλλη προσέγγιση είναι η χρήση της επιστροφής G_t καθώς, από την σκοπιά της βελτιστοποίησης του στόχου της ενισχυτικής μάθησης, ανταμοιβές του παρελθόντος δεν συνεισφέρουν σε κάποιο επίπεδο. Επομένως, αν αντικαταστήσουμε τον όρο $r(\tau)$ με μια εκπτωτική επιστροφή G_t, τότε οδηγούμαστε στην κλασικό Policy Gradient αλγόριθμο που ονομάζεται REINFORCE.

3.5 Μέθοδος REINFORCE

Όπως αναφέρθηκε και στην προηγούμενη ενότητα, ο αλγόριθμος REINFORCE, υπολογίζει το policy gradient ως εξής:

$$\nabla \mathbb{E}_{\pi_\theta}[r(\tau)] = \mathbb{E}_{\pi_\theta}[\sum_{t=1}^{T} G_t \nabla \log \pi_\theta(\alpha_t|s_t)] \tag{3.14}$$

Ακόμη και τώρα δεν έχει λυθεί το πρόβλημα της διασποράς στα δείγματα των τροχιών. Ένας τρόπος να καταλάβουμε διαφορετικά το πρόβλημα, είναι να φανταστούμε τον στόχο της ενισχυτικής μάθησης ως την *Μέγιστη Πιθανότητας Εκτίμησης*. Στην στατιστική, η μέγιστη πιθανότητα εκτίμησης είναι μια τεχνική προσέγγισης των παραμέτρων μια κατανομής πιθανότητας με στόχο να μοντελοποιεί όσο καλύτερα γίνεται τα δεδομένα που παρατηρούμε.[45] Στην συγκεκριμένη τεχνική, δεν έχει σημασία πόσο κακές είναι οι αρχικές εκτιμήσεις, στο όριο των δεδομένων, το μοντέλο θα συγκλίνει στις πραγματικές παραμέτρους. Παρόλα αυτά, στην περίπτωση που τα δείγματα των δεδομένων

έχουν μεγάλη διασπορά, η σταθεροποίηση των παραμέτρων του μοντέλου εξακολουθεί να είναι αρκετά δύσκολη. Στο πλαίσιο μας, κάθε διακύμανση της τροχιάς μπορεί να προκαλέσει μια μη βέλτιστη μετατόπιση στην κατανομή πιθανότητας της πολιτικής. Το πρόβλημα αυτό επιδεινώνεται από την κλίμακα των ανταμοιβών.[46]

Συνεπώς, προσπαθούμε να βελτιστοποιήσουμε τη διαφορά μεταξύ των ανταμοιβών, εισάγοντας μια άλλη μεταβλητή που ονομάζεται πρότυπη τιμή b. Για να διατηρηθεί αμερόληπτη η εκτίμηση κλίσης, η πρότυπη τιμή είναι ανεξάρτητη από τις παραμέτρους της πολιτικής.

REINFORCE με πρότυπη τιμή

$$\nabla \mathbb{E}_{\pi_\theta}[r(\tau)] = \mathbb{E}_{\pi_\theta}[\sum_{t=1}^{T}(G_t - b)\nabla \log \pi_\theta(a_t|s_t)] \tag{3.15}$$

Συμπερασματικά, για να κατανοήσουμε το λόγο που χρησιμεύει η πρότυπη τιμή, πρέπει αρχικά να παρατηρήσουμε πως η παράγωγος παραμένει ίδια με τον επιπλέον όρο. Επιπλέον, η χρήση της πρότυπης τιμής, τόσο στην θεωρία όσο και στην πράξη, μειώνει την διασπορά, ενώ διατηρεί την παράγωγο ανεπηρέαστη.[47] Ο αλγόριθμος REINFORCE παρουσιάζεται στο Σχήμα 22.

Initialize policy parameter θ, baseline b
for iteration=$1, 2, \ldots$ **do**
  Collect a set of trajectories by executing the current policy
  At each timestep in each trajectory, compute
   the *return* $R_t = \sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'}$, and
   the *advantage estimate* $\hat{A}_t = R_t - b(s_t)$.
  Re-fit the baseline, by minimizing $\|b(s_t) - R_t\|^2$,
   summed over all trajectories and timesteps.
  Update the policy, using a policy gradient estimate $\hat{g}$,
   which is a sum of terms $\nabla_\theta \log \pi(a_t \mid s_t, \theta)\hat{A}_t$
end for

Σχήμα 22: Αλγόριθμος REINFORCE

3.6 Μέθοδοι Δράστη - Κριτή

Στις προηγούμενες ενότητες αναλύθηκαν δύο βασικές μέθοδοι ενισχυτικής μάθησης:

- **Βασιζόμενες στην Αξία**. Προσπαθούν να βρουν ή να προσεγγίσουν τη βέλτιστη συνάρτηση τιμών, η οποία είναι μια αντιστοίχηση μεταξύ μιας πράξης και μιας αξίας. Όσο μεγαλύτερη είναι η αξία, τόσο καλύτερη είναι η δράση. Οι μέθοδοι Q-learning και DQN, είναι μέθοδοι αυτής της κατηγορίας.

- **Βασιζόμενες στην Πολιτική**. Πρόκειται για μεθόδους που προσπαθούν να ανακαλύψουν την βέλτιστη πολιτική απευθείας. Μια τέτοια μέθοδος είναι η μέθοδος REINFORCE.

Κάθε μέθοδος έχει τα πλεονεκτήματά της. Για παράδειγμα, οι βασιζόμενες στην πολιτική μέθοδοι είναι καλύτερες για συνεχή και στοχαστικά περιβάλλοντα, έχουν ταχύτερη σύγκλιση, ενώ οι μέθοδοι με βάση την αξία είναι πιο αποτελεσματικές και σταθερές στο δείγμα.

Όταν αυτές οι δύο αλγοριθμικές οικογένειες εγκαθίστανται στην επιστημονική κοινότητα, το επόμενο προφανές βήμα είναι να προσπαθήσουμε να τις συγχωνεύσουμε. Και έτσι γεννήθηκαν οι αλγόριθμοι Δράστη - Κριτή. Οι μέθοδοι αυτές επιδιώκουν να επωφεληθούν από όλα τα πλεονεκτήματα

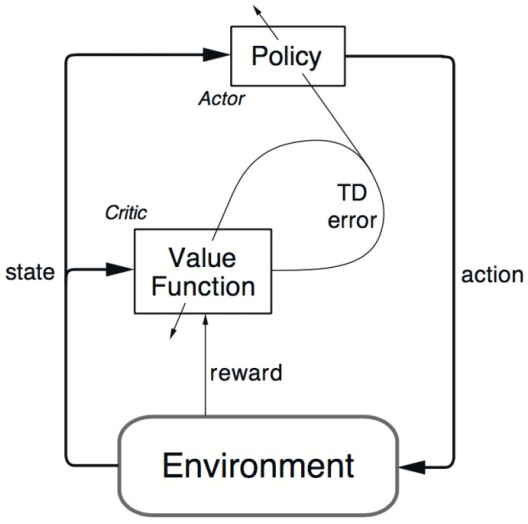
τόσο από τις βασιζόμενες στην αξία όσο και τις βασιζόμενες στην πολιτική μεθόδους, ενώ εξαλείφουν όλα τα μειονεκτήματά τους. Το ερώτημα είναι πώς επιτυγχάνεται αυτό;

Η κεντρική ιδέα είναι ότι διαχωρίζουμε τον αλγόριθμο σε δύο πράκτορες. Έναν για τον υπολογισμό δράσεων βασιζόμενο στην κατάσταση, και έναν άλλο ο οποίος παράγει τις τιμές Q της εκάστοτε πράξης.

Ο Δράστης παίρνει ως είσοδο την κατάσταση και εξάγει την καλύτερη δράση. Βασικά ελέγχει τον τρόπο συμπεριφοράς του πράκτορα, μαθαίνοντας τη βέλτιστη πολιτική (μέθοδος βασισμένη στην πολιτική). Ο κριτής, από την άλλη πλευρά, αξιολογεί μια πράξη με τον υπολογισμό της συνάρτησης αξιολόγησης (μέθοδος βασιζόμενη στην αξία). Αυτά τα δύο μοντέλα συμμετέχουν σε ένα παιχνίδι όπου και οι δύο γίνονται καλύτεροι στο δικό τους ρόλο καθώς εκπαιδεύονται. Το αποτέλεσμα είναι ότι η συνολική αρχιτεκτονική θα μάθει να παίζει πιο αποτελεσματικά το παιχνίδι από ό,τι οι δύο μέθοδοι ξεχωριστά. Η δομή του αλγορίθμου Δράστη - Κριτή φαίνεται στο Σχήμα 23.

Ο Δράστης μπορεί να είναι μια προσεγγιστική συνάρτηση όπως ένα νευρωνικό δίκτυο και στόχος του είναι να παράγει την καλύτερη πράξη για μια δεδομένη κατάσταση. Φυσικά, μπορεί να είναι ένα πλήρως συνδεδεμένο νευρωνικό δίκτυο ή ένα συνελικτικό ή οτιδήποτε άλλο. Ο Κριτής είναι μια άλλη προσεγγιστική συνάρτηση, η οποία λαμβάνει ως είσοδο το περιβάλλον και τη δράση του Δράστη, τα επεξεργάζεται και εξάγει μια αξία (τιμή Q) για το δεδομένο ζεύγος. Υπενθυμίζουμε σε αυτό το σημείο, ότι η τιμή Q είναι ουσιαστικά η μέγιστη μελλοντική ανταμοιβή.

Η εκπαίδευση των δύο δικτύων εκτελείται χωριστά και χρησιμοποιεί gradient ascent (για να βρεθεί το μέγιστο και όχι το ελάχιστο) ώστε να ενημερώσουν και τα δύο, τα βάρη τους. Καθώς ο χρόνος περνά, ο Δράστης μαθαίνει να παράγει όλο και καλύτερες ενέργειες (αρχίζει να μαθαίνει την πολιτική) και ο Κριτής συνεχώς βελτιώνει την αξιολόγηση αυτών των δράσεων. Ο Κρίτης, εν προκειμένω, συμμετέχει στην εκπαίδευση και του δράστη, αντικαθιστώντας την συνάρτηση επιστροφής G_t. Ο Δράστης, δηλαδή, δεν εκπαιδεύεται ανά επεισόδιο, έχοντας ανάγκη την συνολική αξία του επεισοδίου. Αντιθέτως, βασίζεται στην εκτίμηση του Κριτή. Είναι σημαντικό να γίνει κατανοητό ότι η ενημέρωση των βαρών συμβαίνει σε κάθε βήμα (Learning TD) και όχι στο τέλος του επεισοδίου, σε αντίθεση με τις μεθόδους policy gradients.



Σχήμα 23: Δομή Αλγορίθμου A2C

Κεφάλαιο 4

Υλοποίηση και Αξιολόγηση Πειραμάτων

4.1 Εισαγωγή

Στην παρούσα διπλωματική θα μελετήσουμε την συμπεριφορά αλγορίθμων βαθιάς ενισχυτικής μάθησης για προβλήματα πλοήγησης σε περιβάλλοντα συνεχούς χώρου και χρόνου. Συγκεκριμένα, θεωρούμε αυτόνομο όχημα το οποίο πρέπει να εκπαιδευτεί ώστε να ολοκληρώνει επιτυχώς μια κίνηση με σκοπό την μεταφορά του, σε έναν επιθυμητό τελικό χωρικό στόχο. Για την υλοποίηση της ανωτέρω πρόκλησης, θα χρησιμοποιήσουμε τις μεθόδους βαθιάς ενισχυτικής μάθησης *DQN*, *REINFORCE* και *A2C (Δράστη - Κριτή)*, και η μοντελοποίηση όλου του προβλήματος έγινε με χρήση της γλώσσας προγραμματισμού *python*.

Είναι σημαντικό να αναφερθεί η λογική '*από άκρη σε άκρη*' που ακολουθήθηκε στην παρούσα εργασία. Ο όρος αυτός σημαίνει, πως ο τρόπος προσέγγισης του προβλήματος αφορά στην ενιαία επεξεργασία των δεδομένων εισόδου για την δημιουργία μιας ενοποιημένης επίλυσης του προβλήματος και την παραγωγή ενός διανύσματος ταχύτητας στην έξοδο, σε πραγματικό χρόνο.

Στο κεφάλαιο αυτό παρουσιάζεται, αρχικά, ο τρόπος με τον οποίο προσημειώθηκε το περιβάλλον που ενεργεί ο πράκτορας καθώς και οι παραδοχές που υιοθετήθηκαν. Στην συνέχεια αναλύεται η σχεδίαση των πρακτόρων που εκπαιδεύτηκαν και τέλος παρατίθενται τα αποτελέσματα των πειραμάτων που πραγματοποιήθηκαν.

4.2 Μοντελοποίηση του Προβλήματος

4.2.1 Αυτόνομη Πλοήγηση

Στην παρούσα διπλωματική μελετάμε το πρόβλημα της αυτόνομης πλοήγησης από την σκοπιά της ενισχυτικής μάθησης. Σε αυτή την περίπτωση θεωρούμε ότι υπάρχουν ορισμένοι βασικοί άξονες κατά την εκπαίδευση του πράκτορα, που αυξάνουν την πολυπλοκότητα και την γενικευμένη λειτουργία του αλγορίθμου. Οι άξονες αυτοί παρουσιάζονται στην συνέχεια:

Μοντελοποίηση κίνησης

Με τον όρο μοντελοποίηση κίνησης εννοούμε, τις αποφάσεις που μπορεί να λάβει το αυτόνομο όχημα και τι αποτελέσματα αυτές προκαλούν στο περιβάλλον. Αυτό με πιο απλά λόγια, σημαίνει εάν το όχημα αποφασίζει την μελλοντική του μετατόπιση, την μελλοντική του ταχύτητα, την μελλοντική του επιτάχυνση ή οποιαδήποτε άλλο τρόπο μετακίνησής του στο επίπεδο, μπορούμε να μοντελοποιήσουμε. Από την σκοπιά της ρομποτικής, κάθε ρομποτικό όχημα έχει ενσωματωμένους ελεγκτές θέσης ταχύτητας και δύναμης(επιτάχυνσης). Υπάρχει, δηλαδή, η δυνατότητα να λαμβάνει μια μετατόπιση, μια ταχύτητα ή μια δύναμη-επιτάχυνση και να την υλοποιεί με εξαιρετική ακρίβεια και σε πολύ μικρό χρονικό διάστημα. Προφανώς με δεδομένους τους φυσικούς περιορισμούς (αδράνειας, τριβής κτλ).

Στο συγκεκριμένο πείραμα, ως επιλογές δράσεων του αυτόνομου οχήματος θεωρούμε τις αποφάσεις ταχύτητας. Όσον αφορά τις επιλογές ταχυτήτων θεωρήσαμε δύο περιπτώσεις. Η πρώτη αφορά ένα διάνυσμα ταχυτήτων με μια γραμμική ταχύτητα με προσανατολισμό που αλλάζει ως προς το σχετικό σύστημα αξόνων του οχήματος ώστε να το μετακινεί "ευθεία" σύμφωνα με το δικό του σύστημα

και μια γωνιακή ταχύτητα περιστροφής με κατεύθυνση το διάνυσμα που είναι κάθετο στο επίπεδο κίνησης (Σχήμα). Η δεύτερη μοντελοποίηση που χρησιμοποιήθηκε, αφορά την ύπαρξη μόνο γραμμικών ταχυτήτων ως προς τους δύο άξονες του συστήματος του επιπέδου (συστήματος αναφοράς) χωρίς να αλλάζει ο προσανατολισμός του οχήματος. (Σχήμα).

Μια επιπλέον παραδοχή της μοντελοποίησης, είναι η δυνατότητα του οχήματος να λαμβάνει ακαριαία την επιθυμητή ταχύτητα, ενώ στην πραγματικότητα ο ελεγκτής (PID συνήθως) χρειάζεται ένα χρονικό διάστημα 'ανάβασης' (Σχήμα). Ο λόγος που επιλέχθηκε αυτή η απλοποίηση είναι επειδή το πρόβλημα εστιάζει στην αλγοριθμική προσέγγιση του προβλήματος παρά στην αντιμετώπιση σφαλμάτων λόγω φυσικών παραγόντων. Εκείνη η σκοπιά του προβλήματος αφορά την βελτιστοποίηση των τεχνικών από την πλευρά του αυτομάτου ελέγχου.

Θέση Εκκίνησης - Στόχου

Μια εξαιρετικά σημαντική παράμετρος κατά την εκπαίδευση ενός πράκτορα, είναι η θέση από την οποία εκκινεί καθώς και η θέση του επιθυμητού στόχου. Επιπλέον, μείζονος σημασίας είναι εάν οι θέσεις αυτές είναι σταθερές ή αλλάζουν από επεισόδιο σε επεισόδιο. Η αλλαγή αυτή προσθέτει πολυπλοκότητα στο πρόβλημα καθώς ο πράκτορας θα πρέπει να κάνει μια γενίκευση και να προσαρμόζεται σε κάθε συνδυασμό εκκίνησης - στόχου.

Τοπολογία Χάρτη - Στατικά Εμπόδια

Η πλοήγηση σε χώρο με εμπόδια, αποτελεί μια από τις μεγαλύτερες προκλήσεις προς επίλυση. Ανάλογα με τον τρόπο που έχουν δομηθεί τα εμπόδια στον χάρτη δημιουργούν σημαντικές μη γραμμικότητες και συνεπώς τοπικά ελάχιστα (ή μέγιστα) καθιστώντας την σύγκλιση ενός πράκτορα στο επιθυμητό αποτέλεσμα εξαιρετικά δύσκολο.

Δυναμικά Εμπόδια

Ως δυναμικά εμπόδια, ορίζονται τα εμπόδια που αλλάζουν την θέση τους στον χώρο κατά την διάρκεια εκτέλεσης του πειράματος. Χωρίζονται σε δύο κατηγορίες. Η πρώτη αφορά την εκτέλεση μια περιοδικής κίνησης ή έστω μιας ομοιόμορφης κίνησης που καθιστά εφικτή και αξιόπιστη την πρόβλεψη μελλοντικής θέσης. Η δεύτερη κατηγορία αφορά την ύπαρξη δυναμικών εμπορίων που εκτελούν τυχαίες και απρόβλεπτες κινήσεις.

Μια συγκεκριμένη περίπτωση της δεύτερης κατηγορίας, αφορά την ύπαρξη πολλών πρακτόρων να ενεργούν στο ίδιο περιβάλλον. Τέτοιες λογικές πλέον αφορούν μελέτη πολυπρακτορικών συστημάτων, με τα οποία δεν θα ασχοληθούμε στην παρούσα διπλωματική.

Οδομετρία

Με τον όρο οδομετρία, περιγράφονται συνολικά όλα τα στιγμιαία μεγέθη θέσης, ταχύτητας, επιτάχυνσης κ.α. ενός οχήματος. Γνώση της δοσομετρίας του, λοιπόν, σημαίνει επακριβής γνώση των παραπάνω στοιχείων κάθε στιγμή. Στα συγκεκριμένα πειράματα, θεωρούμε δεδομένη, από τον πράκτορα, την γνώση της θέσης του και της ταχύτητας του.

4.2.2 Προσομοίωση του Περιβάλλοντος

Το περιβάλλον που εκτελούνται τα πειράματα είναι συνεχούς χώρου. Αυτό σημαίνει ότι το πρόβλημα δεν μπορεί να μοντελοποιηθεί με κάποιο πλέγμα δυνατών θέσεων. Αντίθετα το αυτόνομο όχημα έχει την δυνατότητα να βρεθεί σε οποιαδήποτε θέση του πεδίου δράσης του και πρέπει να είναι σε θέση να πράξει ορθά. Αναλυτικότερα το περιβάλλον προσομοίωσης μοντελοποιήθηκε ως ένα ορθογώνιο επίπεδο, διαστάσεων 7×8 τετραγωνικών μέτρων.

Όσον αφορά το περιβάλλον προσομοίωσης, δημιουργήθηκε με την απλή λογική της αποθήκευσης της θέσης και της ταχύτητας όλων των στοιχείων που ήταν απαραίτητα για την πλήρη γνώση

του χώρου. Δηλαδή, η θέση του οχήματος και των εμποδίων κάθε στιγμή και οι διαστάσεις τους, η ταχύτητα του οχήματος, η θέση του επιθυμητού στόχου κτλ.

Κάθε βήμα κίνησης θεωρούμε ότι είναι $100ms$ και σε αυτό το διάστημα υποθέτουμε πως η ταχύτητα είναι ομαλή. Χρησιμοποιήθηκαν και οι δύο μοντελοποιήσεις ταχύτητας που αναφέρθηκαν στην προηγούμενη παράγραφο με τους κάτωθι τύπους ομαλής κίνησης(Σχήμα):

$$\textit{Τύπου 1:}\Delta x = v_x \times \Delta t, \Delta y = v_y \times \Delta t \tag{4.1}$$

$$\textit{Τύπου 2:}\Delta x = v_r \times \cos\theta \times \Delta t, \Delta y = v_r \times \sin\theta \times \Delta t, \theta = \omega \times \Delta t \tag{4.2}$$

4.2.3 Μοντελοποίηση των Ανταμοιβών

Ένα πρωταρχικό κομμάτι της μελέτης αφορά τον τρόπο με τον οποίο γίνεται η επιλογή των ανταμοιβών. Είναι σαφές πως η σύγκρουση με κάποιο εμπόδιο θα πρέπει να τιμωρείται με κάποια αρνητική τιμή, ενώ η πρόσβαση στον επιθυμητό στόχο να επιβραβεύεται με μια μη αρνητική. Η δυσκολία έγκειται στην επιλογή των στιγμιαίων ανταμοιβών που δεν αποτελούν τερματικές καταστάσεις.

Για προβλήματα διακριτού χώρου η σταθερή '*χρονική*' τιμωρία, δηλαδή μια σταθερή αρνητική τιμή σε κάθε βήμα, αποτελεί ικανοποιητική προσέγγιση για την επίλυση του προβλήματος. Στην συγκεκριμένη περίπτωση, όμως, που το περιβάλλον είναι συνεχούς χώρου, η τυχαία άφιξη του πράκτορα στον τελικό στόχο, θα είναι εξαιρετικά σπάνια, λόγω του τεράστιου εύρους επιλογών. Συνεπώς, η προσέγγιση αυτή αποτυγχάνει καθώς λαμβάνει, κατά πλειοψηφία, σε κάθε επεισόδιο την ίδια αρνητική αξία ανεξαρτήτως των κινήσεων που επιλέγει.

Συμπερασματικά, είναι ύψιστης σημασίας, η δημιουργία τέτοιων μη τερματικών ανταμοιβών, οι οποίες να εμπεριέχουν μιας μορφής πληροφορία για την πρόοδο του πράκτορα προς τον στόχο. Στην προσπάθεια αυτή, πειραματιστήκαμε με τις εξής βασικές ανταμοιβές:

- **Σχετική Απόσταση από τον Στόχο**. Δηλαδή, πόσο απέχει το όχημα από τον στόχο. Η προσέγγιση αυτή έγινε με δύο τρόπους:

$$\textit{Νόρμα τύπου 1:Ανταμοιβή} = -(|\Delta x| + |\Delta y|) \tag{4.3}$$

$$\textit{Νόρμα τύπου 2:Ανταμοιβή} = -\sqrt{\Delta x^2 + \Delta y^2} \tag{4.4}$$

- **Σχετική Γωνία**. Στην περίπτωση της μοντελοποίησης της ταχύτητας *τύπου 2* (όπου αλλάζει ο προσανατολισμός του οχήματος), θεωρούμε ως σημαντική ανταμοιβή την ελάχιστη απόκλιση μεταξύ της σχετικής γωνίας του πράκτορα με το σταθερό σύστημα αναφοράς και της γωνίας του ευθύγραμμου τμήματος πράκτορα - στόχου. Με απλά λόγια επιβραβεύεται η προσπάθεια του πράκτορα να 'κοιτάει' συνεχώς τον στόχο.

- **Μεταβολή της Σχετικής Απόστασης από τον Στόχο**. Σε αυτή την περίπτωση, δεν ενδιαφερόμαστε για την απόσταση του οχήματος από τον στόχο, αλλά αν αυτή αυξήθηκε ή μειώθηκε στην τελευταία μετατόπιση. Η ιδέα είναι πως δεν μας αφορά πόσο απέχουμε από τον στόχο, αλλά το αν μειώνουμε ή αυξάνουμε την απόσταση αυτή, οπότε και επιβραβεύουμε ή τιμωρούμε τον πράκτορα αντίστοιχα.

$$\textit{Ανταμοιβή} = -((\sqrt{\Delta x^2 + \Delta y^2})_{\textit{τωρινή τιμή}} - (\sqrt{\Delta x^2 + \Delta y^2})_{\textit{προηγούμενη τιμή}}) \tag{4.5}$$

- Συνδυασμός των τεχνικών μείωσης της απόστασης (1 και 3) και της γωνίας με βάρη κανονικοποίησης, ανάλογα με την παράμετρο που θέλουμε να δώσουμε έμφαση.

4.2.4 Σχεδίαση των Πρακτόρων

Για την επίλυση του προβλήματος σχεδιάστηκαν, τρεις πράκτορες. Κάθε ένας υλοποιεί και έναν διαφορετικό αλγόριθμο βαθιάς ενισχυτικής μάθησης. Σε όλες τις περιπτώσεις, θεωρήθηκε πως οι πράξεις που μπορεί να πραγματοποιήσει ο πράκτορας είναι διακριτές, και για τους δύο τρόπους μοντελοποίησης της ταχύτητας. Οι δυνατές πράξεις για κάθε τρόπο κίνησης παρουσιάζονται αναλυτικά στον Πίνακα 1. Επιπλέον, θεωρήθηκε πως όλοι οι πράκτορες προσομοιώνουν οχήματα τα οποία καταλαμβάνουν κυκλικό χώρο και οι διαστάσεις τους είναι ακτίνας $R = 0.3m$.

Πίνακας 1: Σύνολο των πράξεων του πράκτορα για τις δύο μοντελοποιήσεις ταχυτήτων.

Σύνολο Διακριτών Πράξεων			
Τύπου 1		Τύπου 2	
υ_x	υ_y	υ_r	ω
1.0	0.0	1.0	0.0
-1.0	0.0	0.0	0.8
0.0	1.0	0.0	-0.8
0.0	-1.0	-	-

4.2.4.1 Πράκτορας DQN

Αρχικά, ο αλγόριθμος DQN σχεδιάστηκε με ένα νευρωνικό δίκτυο, δύο κρυφών επιπέδων οι νευρώνες των οποίων αλλάζουν αναλόγως του πειράματος. Το δίκτυο προσεγγίζει την συνάρτηση Q, η οποία αξιολογεί την επιλογή κάθε μιας από τις δυνατές δράσεις, για την κατάσταση στην οποία βρίσκεται εκείνη την στιγμή ο πράκτορας.

Επιπλέον, χρησιμοποιήθηκε η τεχνική της αναπαραγωγής εμπειριών. Ο πράκτορας, δηλαδή, αποθηκεύει παρελθοντικά στιγμιότυπα καταστάσεων (συγκεκριμένα τα 10.000 πιο πρόσφατα) και εφαρμόζει εκπαίδευση στο νευρωνικό δίκτυο, από ένα δείγμα 200 τυχαίων δειγμάτων των στιγμιοτύπων αυτών, κάθε φορά.

Τέλος, μία ακόμα σημαντική παράμετρος της σχεδίασης του αλγορίθμου, είναι ο τρόπος με τον οποίο λαμβάνει αποφάσεις. Για να επιτευχθεί ισορροπία μεταξύ '*εκμετάλλευσης*' της πολιτικής και '*εξερεύνησης*', χρησιμοποιούμε την ε - άπληστη τεχνική για την επιλογή των πράξεων. Αναλυτικότερα, η ιδέα της συγκεκριμένης τεχνικής αφορά τον ορισμό μιας παραμέτρου ε, η οποία έχει τον ρόλο της πιθανότητας επιλογής μιας πράξης τυχαία. Αρχικά, η τιμή της είναι ίση με ένα, και σε κάθε εποχή εκπαίδευσης μειώνουμε την τιμή αυτή κατά ένα μέγεθος ελάττωσης ε - *decay*. Συνεπώς κατά την εκκίνηση του αλγορίθμου, η επιλογή των πράξεων γίνεται τυχαία, και σταδιακά διαλέγει όλο και περισσότερο με βάση την πολιτική του.

4.2.4.2 Πράκτορας REINFORCE

Ο αλγόριθμος *REINFORCE* μοντελοποιήθηκε με ένα νευρωνικό δύο κρυφών επιπέδων ξανά, το μέγεθος των οποίων τροποποιήθηκε κατάλληλα για κάθε πείραμα. Το νευρωνικό δίκτυο, αυτό, λαμβάνει ως είσοδο την κατάσταση του περιβάλλοντος (στιγμιαία θέση του πράκτορα, θέση επιθυμητού στόχου κτλ), και δίνει ως έξοδο μια κατανομή πιθανότητας για την επιλογή μιας από τις δυνατές πράξεις του πράκτορα.

Με το πέρας κάθε επεισοδίου, γινόταν αποθήκευση όλων των διαδοχικών καταστάσεων, πράξεων και ανταμοιβών του πράκτορα ως ιστορικό. Με την χρήση του ιστορικού αυτού κατά την εκπαίδευση, πραγματοποιείται ανανέωση των βαρών του νευρωνικού δικτύου ανά έναν συγκεκριμένο αριθμό επεισοδίων. Το ιστορικό αυτό στην συνέχεια άδειαζε, ώστε να αποθηκεύσει τα επόμενα δεδομένα. Ο λόγος που επιλέχθηκε ο αριθμός των επεισοδίων να είναι μεγαλύτερος του 1, είναι η αύξηση των δειγμάτων που λαμβάνονται πριν την αλλαγή του δικτύου. Με αυτό τον τρόπο, βελτιώνεται η αξιολόγηση των προσωρινών βαρών του νευρωνικού πριν αναπροσαρμοστούν εκ νέου.

Τέλος, χρησιμοποιήθηκε η τεχνική της πρότυπης τιμή για να αποσυμπλέξουμε τα πειράματα από το πρόβλημα της μεγάλης διακύμανσης των ανταμοιβών. Συγκεκριμένα έγινε κανονικοποίηση σε κάθε μια από τις εκπτωτικές(μειωμένες) ανταμοιβές, ως προς την μέση τιμή και τυπική απόκλιση αυτών.

4.2.4.3 Πράκτορας A2C

Στον συγκεκριμένο αλγόριθμο εκπαιδεύονται δύο νευρωνικά δίκτυα. Ένα για τον Δράστη και ένα για τον Κριτή. Και τα δύο δίκτυα, αυτά, είναι δύο κρυφών επιπέδων και 12 νευρώνων έκαστο. Στο επίπεδο εξόδου, ο Δράστης δίνει μια κατανομή πιθανότητας όλων των δυνατών πράξεων (όπως στο policy gradient), ενώ ο Κριτής, δίνει μια εκτίμηση αξίας της κατάστασης στην οποία βρίσκεται κάθε στιγμή.

Για την εκπαίδευση και των δύο δικτύων υλοποιήθηκαν δύο προσέγγισης. Η πρώτη, αναπροσαρμόζει τα βάρη των δικτύων ανά κάποιο συγκεκριμένο αριθμό επεισοδίων με την χρήση του ιστορικού που αναφέραμε και στον αλγόριθμο Policy Gradient. Η δεύτερη τεχνική χρησιμοποιεί την αναπαραγωγή εμπειριών για την δημιουργία του δείγματος των δεδομένων που θα χρησιμοποιηθούν για την εκπαίδευση σε κάθε εποχή, όπως στον αλγόριθμο DQN. Εν τέλει, όπως θα φανεί και από τα αποτελέσματα στην συνέχεια, η τεχνική της αναπαραγωγής εμπειριών κατάφερε να δώσει αποτελέσματα.

4.3 Πειραματικά Αποτελέσματα

4.3.1 Προσέγγιση των Πειραμάτων

Στην παρούσα ενότητα θα εξηγήσουμε συνοπτικά την συλλογιστική που ακολουθήθηκε για την πορεία εκτέλεσης των πειραμάτων. Και για τους τρεις αλγορίθμους η σειρά των πειραμάτων ήταν όμοια, με την διαφοροποίηση να εμπίπτει στην αποτελεσματικότητα τους ως προς κάθε περίπτωση. Στο συγκεκριμένο σημείο, θα γίνει αναφορά στις παραμέτρους ως προς τις οποίες κατηγοριοποιήθηκε η πειραματική διαδικασία. Δόθηκε έμφαση στην μοντελοποίηση των ταχυτήτων, καθώς και στις θέσεις εκκίνησης και θέσεις στόχου. Πιο συγκεκριμένα, εάν αυτές θα είναι συγκεκριμένες και συνεχώς ίδιες (σταθερές), ή εάν θα είναι τυχαίες και θα αλλάζουν από επεισόδιο σε επεισόδιο. Τέλος, εξαιρετικά σημαντική παράμετρος είναι η μορφή του χάρτη. Αν, δηλαδή, είναι άδειος ή εάν υπάρχουν εμπόδια.

Καταρχάς, για κάθε διαφορετική παράμετρο του πειράματος έγινε προσπάθεια να επιλυθεί και με τις δύο μοντελοποιήσεις ταχύτητας που αναφέραμε στα προηγούμενα κεφάλαια. Στην συνέχεια για κάθε περίπτωση, προσπαθήσαμε να επιλύσουμε το πιο απλό πρόβλημα, που είναι η πλοήγηση σε άδειο χάρτη, σταθερής αρχικής θέσης και σταθερού τελικού στόχου. Επόμενη πρόκληση ήταν η δυνατότητα γενίκευσης σε άδειο χάρτη, για τυχαία αρχική θέση ή τυχαίο τελικό στόχο και στην συνέχεια και τα δύο ταυτόχρονα. Τέλος, για την περίπτωση αρχικής και τελικής θέσης σταθερής έγινε μελέτη σε χάρτη με ύπαρξη εμποδίων.

Εν κατακλείδι, μετά από εκτενή μελέτη και πειραματισμό με τα διαφορετικά μοντέλα, επετεύχθη με επιτυχία η σύγκλιση πολλών εξ αυτών στον επιθυμητό στόχο. Στο σημείο αυτό, αξίζει να αναφερθεί πως η συνθήκη τερματισμού, κατά την εκπαίδευση των αλγορίθμων είναι, ο πράκτορας να έχει επιτύχει τον σκοπό του στα 100 ή 200 πιο πρόσφατα επεισόδια. Στον Πίνακα 2 παρουσιάζονται αναλυτικά όλες οι διαφορετικές παράμετροι που τροποποιήθηκαν κατά την εκτέλεση των αλγορίθμων, μαζί με την περιγραφή αυτών. Στα επόμενα κεφάλαια, θα αναλυθούν τα αποτελέσματα των πειραμάτων και τα ορθώς εκπαιδευμένα μοντέλα, για κάθε αλγόριθμο ξεχωριστά.

4.3.2 Πράκτορας DQN

Σε πρώτη φάση θα μελετηθεί, όπως ήδη αναφέραμε, το πρόβλημα σταθερής αρχικής θέσης και σταθερού στόχου, σε άδειο χάρτη. Στο συγκεκριμένο πείραμα και με τις δύο μοντελοποιήσεις ταχυτήτων, βρέθηκε η βέλτιστη λύση του προβλήματος. Σε κάθε ένα από τα επόμενα πειράματα χρησιμο-

Πίνακας 2: Παράμετροι των πειραμάτων

Παράμετροι	Περιγραφή	DQN	REINFORCE	A2C
reward_type	Ο τύπος των ανταμοιβών για τα μη τερματικά βήματα	✓	✓	✓
collision_reward	Ανταμοιβής σύγκρουσης του πράκτορα με εμπόδιο	✓	✓	✓
goal_reward	Ανταμοιβή άφιξης του πράκτορα στον επιθυμητό στόχο	✓	✓	✓
episode_steps	Μέγιστο πλήθος βημάτων ανά επεισόδιο	✓	✓	✓
actor_learning_rate	Η ταχύτητα εκμάθησης του δράστη		✓	✓
critic_learning_rate	Η ταχύτητα εκμάθησης του κριτή	✓		✓
epsilon	Παράμετρος ε της ε - άπληστης τεχνικής	✓		
epsilon_decay	Ρυθμός μείωσης της τιμής ε	✓		
epsilon_min	Ελάχιστη τιμή της ε	✓		
gamma	Εκπτωτικός όρος για μελλοντικές ανταμοιβές	✓	✓	✓
neural_network_actor	Νευρωνικό Δίκτυο του Δράστη		✓	✓
neural_network_critic	Νευρωνικό Δίκτυο του Κριτή	✓		✓

ποιήθηκε νευρωνικό δίκτυο με 12 νευρώνες σε κάθε ένα από τα δύο κρυφά επίπεδα, με συναρτήσεις ενεργοποίησης τις συναρτήσεις ReLU, ενώ στο επίπεδο εξόδου η συνάρτηση ενεργοποίησης είναι γραμμική. Στην συνέχεια παρουσιάζονται αναλυτικά τα αποτελέσματα για κάθε περίπτωση.

4.3.2.1 Σταθερή θέση Εκκίνησης και Στόχου

Μοντελοποίηση ταχύτητας Τύπου 1

Πείραμα 1

Οι τιμές των παραμέτρων που χρησιμοποιήθηκαν κατά την εκπαίδευση του αλγορίθμου, φαίνονται στον Πίνακα 3. Στο συγκεκριμένο πείραμα, ο πράκτορας οδηγήθηκε σε βέλτιστη επίλυση του προβλήματος για το εύρος δράσεων που είχε στην διάθεση του. Η αρχική θέση του οχήματος είναι στο σημείο $(1.0, 1.0)$ και η θέση του στόχου στο σημείο $(4.0, 4.0)$ Συγκεκριμένα η τροχιά του παρουσιάζεται στο Σχήμα 24.

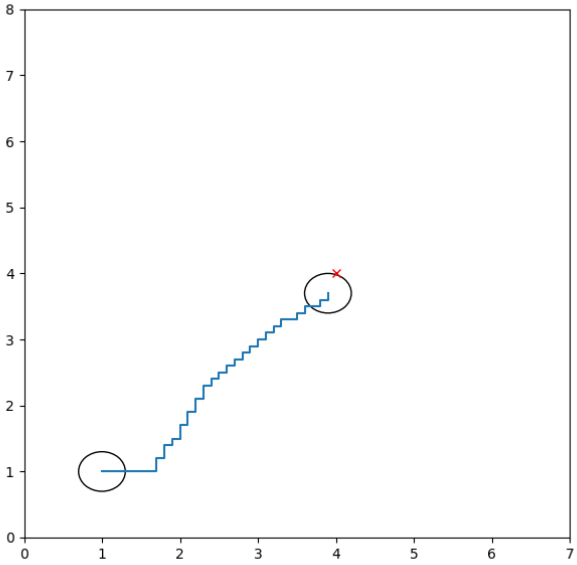
Πίνακας 3: Παράμετροι Πειράματος 1

Παράμετροι	DQN
reward_type	Σχετική Απόσταση από τον Στόχο
goal_reward	0
episode_steps	500
critic_learning_rate	0.001
epsilon_decay	0.999
epsilon_min	0.01
gamma	0.95

Πείραμα 2

Όπως και στο προηγούμενο πείραμα και ο πράκτορας του Πειράματος 2 συγκλίνει στην βέλτιστη διαδρομή. Στην συγκεκριμένη περίπτωση, η θέση του τελικού στόχου είναι στο σημείο $(5.0, 5.0)$. Σε αυτό το σημείο να τονίσουμε πως για τις επιλογές ταχυτήτων *'τύπου 1'*, κάθε τροχιά η οποία βρίσκεται εντός του ορθογωνίου παραλληλογράμμου μεταξύ της αρχικής θέσης και του επιθυμητού στόχου και συνεχώς πλησιάζει τον στόχο, είναι βέλτιστη. Οι παράμετροι και η τροχιά του εκπαιδευμένου αλγορίθμου παρουσιάζονται στον Πίνακα 4 και στο Σχήμα 25, αντίστοιχα.

Μοντελοποίηση ταχύτητας Τύπου 2



Σχήμα 24: Τροχιά αυτόνομου οχήματος για το Πείραμα 1

Πίνακας 4: Παράμετροι Πειράματος 2

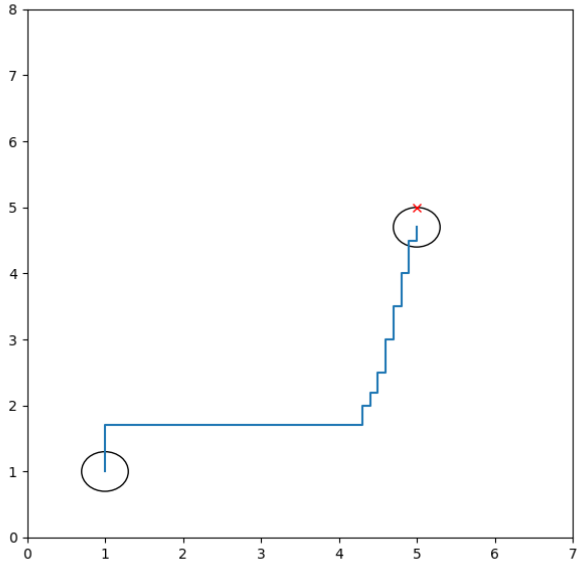
Παράμετροι	**DQN**
reward_type	Σχετική Απόσταση από τον Στόχο
goal_reward	0
episode_steps	200
critic_learning_rate	0.001
epsilon_decay	0.99
epsilon_min	0.01
gamma	0.95

Πείραμα 3

Στο συγκεκριμένο πείραμα επετεύχθη η βελτιστοποίηση της τροχιάς και για την μοντελοποίηση ταχυτήτων *'τύπου 2'*, όπως φαίνεται στο Σχήμα 26. Αυτό που δεν μπορεί να παρατηρηθεί στο σχήμα, είναι η περιστροφή του πράκτορα. Η διαδικασία που ακολουθεί είναι, αρχικά, η στροφή μέχρι να 'κοιτάει' τον στόχο και στην συνέχεια κινείται ευθεία προς αυτόν. Οι παράμετροι παρουσιάζονται στον Πίνακα 5. Ο τελικός στόχος είναι στο σημείο $(4.0, 4.0)$.

Πείραμα 4

Τελευταίο πείραμα για την περίπτωση της σταθερής θέσης και στόχου, είναι όμοιο με το Πείραμα 3, με μια βασική διαφορά στις δράσεις περιστροφής. Συγκεκριμένα, αντί για τις δύο δράσεις $[0.0, \pm 0.8]$, περιστρέφεται και μετατοπίζεται παράλληλα $[0.5, \pm 0.5]$. Οι υπόλοιπες παράμετροι φαίνονται στον Πίνακα 6. Όπως φαίνεται στο Σχήμα 27 το συγκεκριμένο μοντέλο λειτουργεί σχεδόν βέλτιστα, καθώς δεν βρίσκει μετωπικά τον στόχο και συνεπώς διανύει μεγαλύτερη απόσταση από αυτή που θα χρειαζόταν.



Σχήμα 25: Τροχιά αυτόνομου οχήματος για το Πείραμα 2

Πίνακας 5: Παράμετροι Πειράματος 3

Παράμετροι	**DQN**
reward_type	Σχετική Γωνία
goal_reward	0
episode_steps	200
critic_learning_rate	0.001
epsilon_decay	0.999
epsilon_min	0.01
gamma	0.95

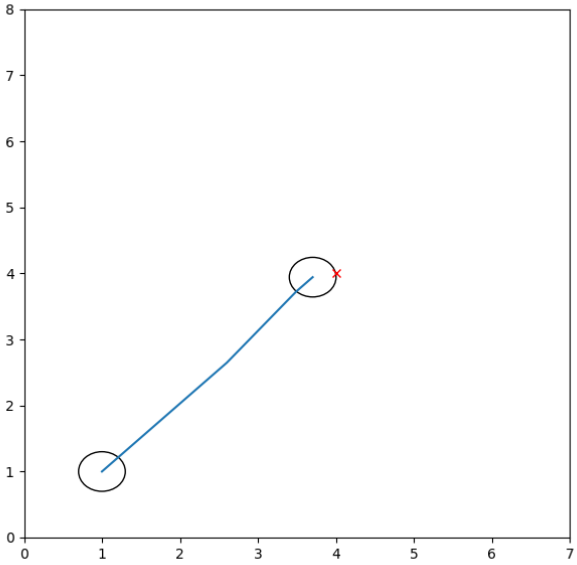
4.3.2.2 Τυχαία θέση Εκκίνησης και Στόχου

Στην συνέχεια έγινε προσπάθεια να επιλυθεί το πρόβλημα άδειου χάρτη, αλλά τυχαίας αρχικής θέσης και τυχαίου τελικού στόχου. Σε αυτό το πρόβλημα η μοντελοποίηση ταχύτητας ’*τύπου 2*’ απέτυχε κατά την εκπαίδευση και δεν έδωσε αξιόπιστα αποτελέσματα. Αντίθετα, η προσέγγιση ταχύτητας ’*τύπου 1*’, ανταποκρίθηκε πολύ καλύτερα.

Κατά την αντιμετώπιση της συγκεκριμένης πρόκλησης με χρήση της τεχνικής *DQN*, παρουσιάστηκε σε όλα τα Πειράματα που εκπαιδεύτηκαν επιτυχώς, το εξής φαινόμενο. Λόγω της άπληστης πολιτικής επιλογής κινήσεων, σε ορισμένες περιπτώσεις ο πράκτορας εγκλωβιζόταν σε ’*κυκλικές*’ τροχιές. Διάλεγε, δηλαδή, δράσεις που τον οδηγούσαν σε θέσεις που είχε ήδη βρεθεί.

Πείραμα 5

Στην συνέχεια, παρατίθεται το πείραμα τυχαίας αρχικής θέσης και τυχαίου τελικού στόχου, που έδωσε τα καλύτερα αποτελέσματα. Οι παράμετροι του *Πειράματος 5* παρουσιάζονται στον Πίνακα 7. Η αξιολόγηση του αλγορίθμου έγινε σε ένα δείγμα 1000 περιπτώσεων τυχαίων αρχικών δεδομένων. Τα αποτελέσματα ήταν 83.5% επιτυχίες ενώ για τις αποτυχίες (16.5%), ευθύνεται πάντα το φαινόμενο ’*κυκλικών τροχιών*’ που αναφέρθηκε στην προηγούμενη παράγραφο. Στο Σχήμα 28, παρουσιάζονται ορισμένες τροχιές επιτυχημένων επεισοδίων, ενώ στο Σχήμα 29 φαίνεται το φαινόμενο ’*κυκλικών τροχιών*’.



Σχήμα 26: Τροχιά αυτόνομου οχήματος για το Πείραμα 3

Πίνακας 6: Παράμετροι Πειράματος 4

Παράμετροι	**DQN**
reward_type	Σχετική Γωνία
goal_reward	0
episode_steps	200
critic_learning_rate	0.001
epsilon_decay	0.999
epsilon_min	0.01
gamma	0.95

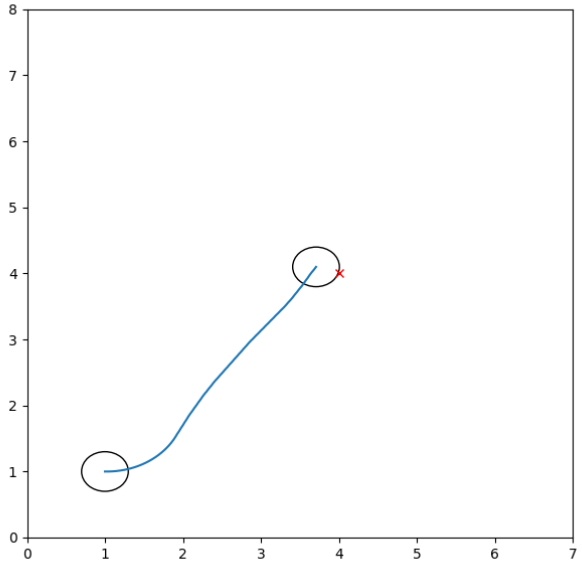
Πείραμα 6

Παρόλο που το συγκεκριμένο πείραμα δεν δίνει καλύτερα αποτελέσματα από το *Πείραμα 5*, αξίζει να γίνει αναφορά λόγω της μοντελοποίησης της ανταμοιβής με την ιδέα της *Μεταβολής της Σχετικής Απόστασης από τον Στόχο*. Ο παράμετροι παρουσιάζονται στον Πίνακα 8.Μετά από την αξιολόγηση 1000 τυχαίων δειγμάτων, όπως και προηγουμένως, το ποσοστό επιτυχιών ήταν 73.5%.

Ο λόγος που γίνεται αναφορά στο συγκεκριμένο πείραμα, αφορά την φύση των ανταμοιβών αυτών. Πρόκειται για ανταμοιβές που παίρνουν και θετικές τιμές. Συνεπώς χωρίς την κατάλληλη θετική επιβράβευση για την άφιξη στον στόχο, υπάρχει κίνδυνος, ο πράκτορας να επιλέξει να κινείται προς τον στόχο καθ' όλη την διάρκεια του επεισοδίου χωρίς να τον φτάνει, με στόχο να συλλέξει όσες περισσότερες θετικές ανταμοιβές μπορεί. Στο συγκεκριμένο πείραμα, κατά την εκπαίδευση, ο πράκτορας συνέκλινε στον επιθυμητό στόχο. Παρόλα αυτά, η συγκεκριμένη μοντελοποίηση των ανταμοιβών δεν ενδείκνυται για προβλήματα ελαχιστοποίησης απόστασης ή χρόνου.

4.3.3 Πράκτορας Policy Gradient

Σε κάθε ένα από τα επόμενα πειράματα χρησιμοποιήθηκε νευρωνικό δίκτυο με 12 νευρώνες σε κάθε ένα από τα δύο κρυφά επίπεδα, με συναρτήσεις ενεργοποίησης τις συναρτήσεις ReLU, ενώ στο επίπεδο εξόδου η συνάρτηση ενεργοποίησης είναι η softmax. Η σειρά εκτέλεσης των πειραμάτων εί-



Σχήμα 27: Τροχιά αυτόνομου οχήματος για το Πείραμα 4

Πίνακας 7: Παράμετροι Πειράματος 5

Παράμετροι	**DQN**
reward_type	Σχετική Απόσταση από τον Στόχο
goal_reward	0
episode_steps	200
critic_learning_rate	0.001
epsilon_decay	0.99
epsilon_min	0.01
gamma	0.95

ναι η ίδια με αυτή για τον αλγόριθμο DQN. Αρχικά, μελετήθηκε το πρόβλημα σταθερής αρχικής θέσης και σταθερού στόχου, σε άδειο χάρτη. Συγκεκριμένα μελετήθηκε η συμπεριφορά του προβλήματος και για τις δύο μοντελοποιήσεις ταχυτήτων για διαφορετικούς τύπους μη τερματικών ανταμοιβών. Στην συνέχεια παρουσιάζονται αναλυτικά τα αποτελέσματα του πρώτου προβλήματος, για κάθε περίπτωση.

4.3.3.1 Σταθερή θέση Εκκίνησης και Στόχου

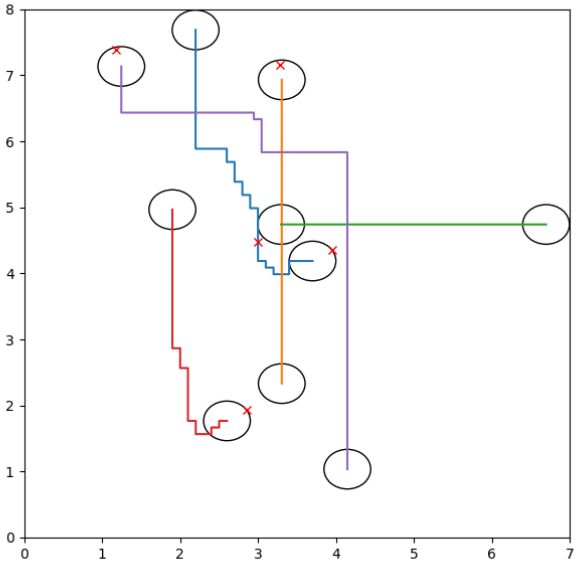
Μοντελοποίηση ταχύτητας Τύπου 1

Πείραμα 7

Στον Πίνακα 9 παρουσιάζονται οι παράμετροι εκπαίδευσης του πράκτορα. Στο συγκεκριμένο πείραμα επετεύχθη η προσέγγιση του στόχου $[6.0, 6.0]$ από την θέση $[1.0, 1.0]$ με βέλτιστη τροχιά (Σχήμα 30).

Πείραμα 8

Στον Πίνακα 10 παρουσιάζονται οι παράμετροι εκπαίδευσης του αλγορίθμου. Στο συγκεκριμένο πείραμα επετεύχθη η προσέγγιση του στόχου $[4.0, 4.0]$ από την θέση $[1.0, 1.0]$ με βέλτιστη τροχιά



Σχήμα 28: Επιτυχημένες τροχιές αυτόνομου οχήματος για το Πείραμα 5

Πίνακας 8: Παράμετροι Πειράματος 6

Παράμετροι	**DQN**
reward_type	Μεταβολή της Σχετικής Απόστασης από τον Στόχο
goal_reward	50
episode_steps	200
critic_learning_rate	0.001
epsilon_decay	0.995
epsilon_min	0.01
gamma	0.95

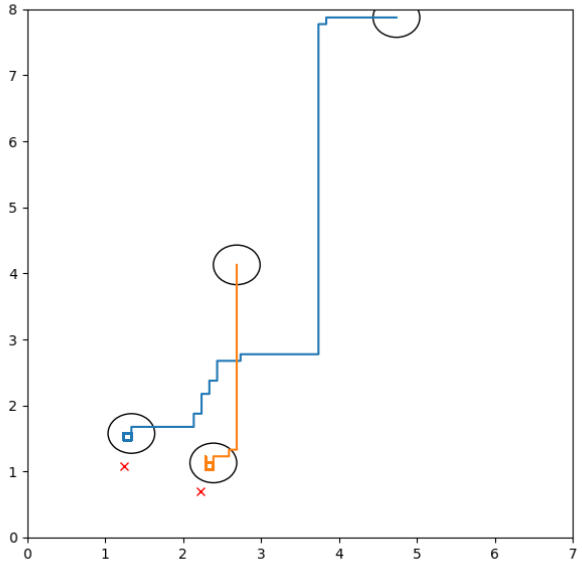
(Σχήμα 31). Η ανταμοιβή σε αυτή την περίπτωση είναι και πάλι η σχετική απόσταση από τον στόχο, αλλά αυτή την φορά μοντελοποιημένη με την ευκλείδεια νόρμα, σε αντίθεση με τις υπόλοιπες περιπτώσεις που είναι με νόρμα τύπου 1.

Μοντελοποίηση ταχύτητας Τύπου 2

Πείραμα 9

Οι παράμετροι του πειράματος παρουσιάζονται στον Πίνακα 11. Στο συγκεκριμένο πείραμα επιλέχθηκε η ανταμοιβή της μεταβολής της σχετικής θέσης από τον στόχο. Η τροχιά στην οποία συνέκλινε ο πράκτορας παρουσιάζεται στο Σχήμα 32. Εκ πρώτης όψεως, δείχνει σαν να έχει εκπαιδευτεί λανθασμένα. Το παράδοξο όμως είναι ότι όσες φορές και αν επαναληφθεί το πείραμα, η τροχιά θα είναι πάντα η ίδια.

Αν αναλυθεί εκτενώς η μορφή της ανταμοιβής, τότε το αποτέλεσμα είναι απολύτως κατανοητό. Όπως αναφέραμε και στην αντίστοιχη μοντελοποίηση ανταμοιβής για τον αλγόριθμο DQN (Πείραμα 6) δεν είναι ορθό να χρησιμοποιηθούν θετικές ανταμοιβές στην προσπάθεια επίλυσης προβλήματος ελαχιστοποίησης βημάτων για την επίτευξη ενός τελικού σκοπού. Το συγκεκριμένο Πείραμα, αποτελεί ξεκάθαρη απόδειξη του προηγούμενου ισχυρισμού. Όπως παρατηρούμε στην τροχιά, το όχημα κινείται συνεχώς προς τον στόχο. Μειώνει δηλαδή συνεχώς την μεταξύ τους απόσταση. Συνεπώς, αν



Σχήμα 29: Αποτυχημένες κυκλικές τροχιές αυτόνομου οχήματος για το Πείραμα 5

Πίνακας 9: Παράμετροι Πειράματος 7

Παράμετροι	**REINFORCE**
reward_type	Σχετική Απόσταση από τον Στόχο
goal_reward	50
episode_steps	200
actor_learning_rate	0.001
episodes_per_update	3
gamma	0.95

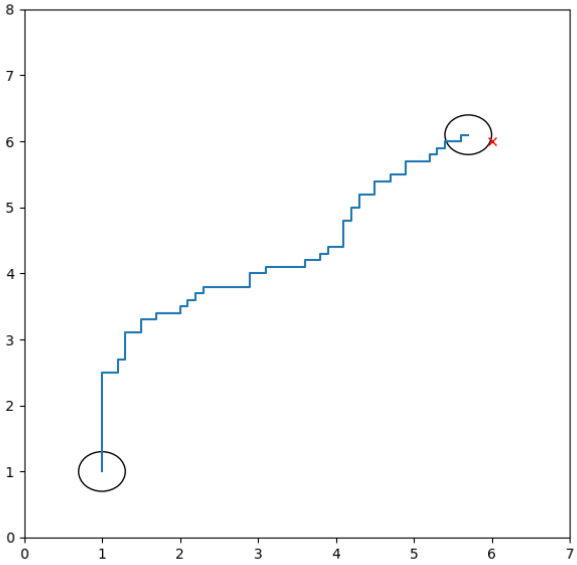
ο πράκτορας ελαχιστοποιούσε την τροχιά θα λάμβανε μικρότερο αριθμό από θετικές ανταμοιβές πριν λάβει την τελική θετική ανταμοιβή άφιξης. Είναι, πλέον, εμφανές πως η τροχιά που ακολουθεί του προσφέρει μεγαλύτερη συνολική επιστροφή από την επιλογή της μικρότερης τροχιάς.

Ένα ακόμα ενδιαφέρον στοιχείο, αφορά το σημείο κατά το οποίο αποφάσισε να αλλάξει πορεία προς τα πάνω. Αν υποθέσουμε πως ο πράκτορας κινείται συνεχώς προς τα δεξιά, τότε σύμφωνα με την συγκεκριμένη μοντελοποίηση ανταμοιβών, θα λάμβανε όλο και μικρότερες θετικές ανταμοιβές, μέχρι να φτάσει στο ύψος του επιθυμητού στόχου, όπου από εκεί και πέρα θα άρχιζε να απομακρύνεται και συνεπώς να λαμβάνει αρνητικές ανταμοιβές. Επομένως, ο πράκτορας πλησιάζει αρκετά στην νοητή αυτή ευθεία μεταξύ της θέσης του και του στόχου, λαμβάνει όσο το δυνατόν περισσότερες θετικές ανταμοιβές, πριν αρχίσει να κινείται προς τον στόχο.

Συμπερασματικά, το συγκεκριμένο πείραμα είναι ενδεικτικό της σημασίας που έχουν οι ανταμοιβές στα αποτελέσματα των Πειραμάτων. Ενώ, δηλαδή, η τροχιά για το πρόβλημα που θέλουμε να επιλύσουμε είναι πολύ μακριά από την βέλτιστη, ο πράκτορας έχει βρει την βέλτιστη επιλογή για τις ανταμοιβές που του δόθηκαν.

Πείραμα 10

Οι παράμετροι του πειράματος παρουσιάζονται στον Πίνακα 12. Πρόκειται για ανταμοιβή Σχετικής γωνίας και όπως φαίνεται και στο Σχήμα 33 συνέκλινε από την θέση $[1.0, 1.0]$ στον τελικό στόχο $[4.0, 4.0]$ με βέλτιστο τρόπο.



Σχήμα 30: Τροχιά αυτόνομου οχήματος για το Πείραμα 7

Πίνακας 10: Παράμετροι Πειράματος 8

Παράμετροι	**REINFORCE**
reward_type	Ευκλείδεια Απόσταση
goal_reward	200
episode_steps	200
actor_learning_rate	0.001
episodes_per_update	3
gamma	0.95

4.3.3.2 Τυχαία θέση Εκκίνησης και Στόχου

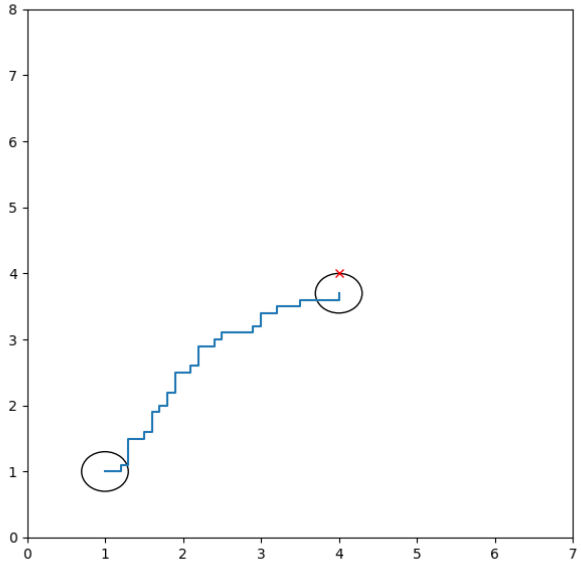
Στην συνέχεια, αποτελέσματα έδωσαν μόνο τα πειράματα με την μοντελοποίηση ταχυτήτων τύπου 1. Το πρώτο αφορά σταθερό τελικό στόχο και τυχαία αρχική θέση, ενώ στο δεύτερο πείραμα, τόσο η αρχική θέση όσο και ο επιθυμητός στόχος είναι τυχαία.

Πείραμα 11

Στον Πίνακα 13 παρουσιάζονται αναλυτικά οι παράμετροι του πειράματος. Πρόκειται για πείραμα τυχαίας αρχικής θέσης του πράκτορα αλλά σταθερή τελικής θέσης $[4.0, 4.0]$. Στο Σχήμα 34 παρουσιάζονται οι πορείες ορισμένων τυχαίων επεισοδίων. Η τροχιά του κάθε επεισοδίου απεικονίζεται με διαφορετικό χρώμα στο σχήμα. Όσον αφορά τον κόκκινο κύκλο γύρω από το σημείο του τελικού στόχου, δεν δείχνει το όχημα. Αντιθέτως δείχνει το εύρος χώρου γύρω από το οποίο, αν φτάσει το κέντρο του πράκτορα, τότε έχει πετύχει τον σκοπό του. Το ποσοστό επιτυχίας είναι εξαιρετικό 97.3%, αλλά η τροχιά απέχει κατά πολύ της βέλτιστης και αυτό οφείλεται στην μεγάλη στοχαστικότητα των δράσεων σε αρκετές από τις καταστάσεις. Το φαινόμενο αυτό θα εξηγηθεί στο Κεφάλαιο 5.

Πείραμα 12

Το τελευταίο επιτυχημένο πείραμα αφορά το πρόβλημα τυχαίας αρχικής θέσης και τυχαίου τελικού στόχου. Οι παράμετροι του πειράματος παρουσιάζονται στον Πίνακα 14. Όσον αφορά τα αποτε-



Σχήμα 31: Τροχιά αυτόνομου οχήματος για το Πείραμα 8

Πίνακας 11: Παράμετροι Πειράματος 9

Παράμετροι	**REINFORCE**
reward_type	Μεταβολή της Σχετικής Απόστασης από τον Στόχο
goal_reward	0
episode_steps	200
actor_learning_rate	0.0001
episodes_per_update	3
gamma	0.95

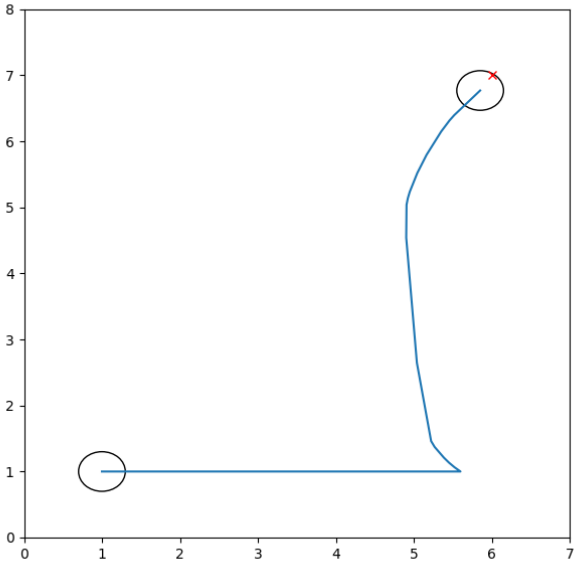
λέσματα των πειραμάτων, ο αλγόριθμος REINFORCE απέδωσε πολύ μεγαλύτερα ποσοστά επιτυχημένων προσπαθειών. Συγκεκριμένα, σε έναν αριθμό 1000 επεισοδίων αξιολόγησης του αλγορίθμου, προέκυψε πως το 98.1% πέτυχαν την αποστολή τους. Στο Σχήμα 35 παρουσιάζονται ενδεικτικά ορισμένες τυχαίες τροχιές από την διαδικασία επαλήθευσης.

Κοιτώντας τις τροχιές αυτές συμπεραίνει κανείς απευθείας, πως παρόλο που όλες συγκλίνουν στον επιθυμητό στόχο, καμία δεν ακολουθεί μια τροχιά κοντά στην βέλτιστη. Ακόμα χειρότερα φαίνεται πως σε πολλές περιπτώσεις ακολουθούν πολύ ανορθόδοξες πορείες στην προσπάθειά τους. Το φαινόμενο αυτό ερμηνεύεται στο Κεφάλαιο 5.

4.3.4 Πράκτορας Δράστη - Κριτή

Ο πράκτορας A2C αποτελείται από δύο νευρωνικά δίκτυα ανεξάρτητα, ένα για τον δράστη και ένα για τον κριτή. Το νευρωνικό δίκτυο του δράστη είναι ίδιας αρχιτεκτονικής με αυτό του REINFORCE ενώ του κριτή ίδιο με του DQN, με μόνη διαφορά πως έχει μια τιμή εξόδου. Για την περίπτωση του Δράστη - Κριτή ακολουθήθηκε η ίδια πορεία εκτέλεσης πειραμάτων, από σταθερές θέσεις σε τυχαίες. Για το πρόβλημα σταθερού σημείου εκκίνησης και σταθερού επιθυμητού στόχου, το μοντέλο A2C συνέκλινε στην βέλτιστης λύση, όπως και τα προηγούμενα. Συνεπώς, δεν υπάρχει ουσιαστικός λόγος να παρουσιαστούν τα πειράματα. Η παρουσίαση των αποτελεσμάτων ξεκινά από το πρόβλημα τυχαίας αρχικής και επιθυμητής θέσης για την μοντελοποίηση ταχυτήτων τύπου 1.

Στην συνέχεια, επειδή στο γενικευμένο πρόβλημα αρχικής - τελικής θέσης, ο πράκτορας A2C



Σχήμα 32: Τροχιά αυτόνομου οχήματος για το Πείραμα 9

Πίνακας 12: Παράμετροι Πειράματος 10

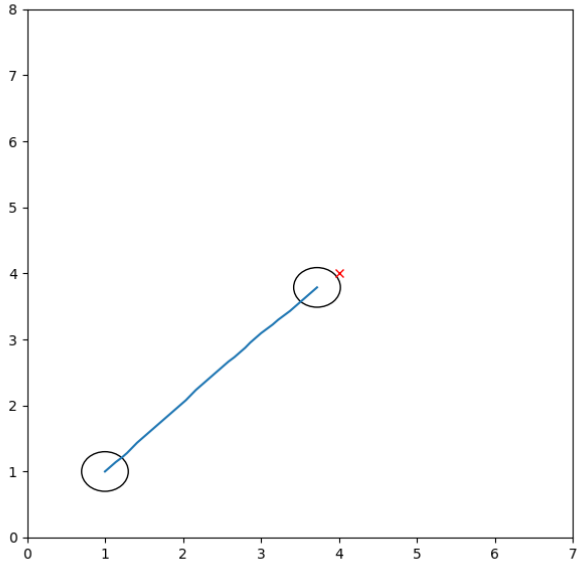
Παράμετροι	**REINFORCE**
reward_type	Σχετική Γωνία
goal_reward	50
episode_steps	200
actor_learning_rate	0.0001
episodes_per_update	3
gamma	0.95

έδωσε τα καλύτερα αποτελέσματα συγκριτικά με τα προηγούμενα δύο μοντέλα, έγινε μελέτη ακόμα πιο πολύπλοκων προβλημάτων.

Συγκεκριμένα, η πρώτη επιτυχημένη προσπάθεια, αφορά τον τρόπο με τον οποίο μετακινείται το όχημα. Όπως έχουμε ήδη αναφέρει, η προσέγγιση που ακολουθήσαμε, είναι ο πράκτορας να επιλέγει μια ταχύτητα και αυτή να υλοποιείται ακαριαία. Στην πραγματικότητα αυτό δεν συμβαίνει. Για παράδειγμα, εάν ο πράκτορας κινείται με μια ταχύτητα $0.1m/s$ και στην συνέχεια αποφασίσει να κινηθεί με την ίδια ταχύτητα αλλά προς την αντίθετη κατεύθυνση, τότε χρειάζεται έναν χρόνο να επιβραδύνει και στην συνέχει να επιταχύνει ξανά. Η ουσία της συγκεκριμένης απόκλισης από την πραγματικότητα είναι, πως αν ο πράκτορας επιλέξει μια ταχύτητα για ένα συγκεκριμένο χρονικό διάστημα, δεν είναι δεδομένο που ακριβώς θα βρεθεί. Σύμφωνα με την μέχρι τώρα μοντελοποίηση, είναι ασφαλές να θεωρήσει κατά την εκπαίδευση, βέβαιη την σύνδεση ταχύτητας και μελλοντικής θέσης που θα βρεθεί, πριν ακόμα πραγματοποιηθεί.

Επειδή, δεν υπάρχει η δυνατότητα να μοντελοποιηθεί στο περιβάλλον που σχεδιάσαμε, η δυναμική σχέση που θα πραγματοποιούσε αξιόπιστα την κίνηση, ακολουθήθηκε η εξής προσέγγιση. Εφόσον εστιάζουμε την ουσία στην στοχαστικότητα της μελλοντικής θέσης για μια επιλογή ταχύτητας, δώσαμε μια τυχαία στοχαστική τιμή στην υλοποίηση της μετατόπισης. Συγκεκριμένα, θεωρήσαμε πως για μια συγκεκριμένη ταχύτητα, η μετατόπιση θα βρίσκεται μεταξύ ενός εύρους τιμών $\pm 10\%$ γύρω από την ντετερμινιστική μετατόπιση της ευθύγραμμης ομαλής κίνησης. Δηλαδή,

$$\Delta x = (v + e) * \Delta t, \text{ όπου } e \in [-10\%, 10\%] \times \textit{Εύρος ταχυτήτων}$$



Σχήμα 33: Τροχιά αυτόνομου οχήματος για το Πείραμα 10

Πίνακας 13: Παράμετροι Πειράματος 11

Παράμετροι	**REINFORCE**
reward_type	Σχετική Απόσταση από τον Στόχο
goal_reward	50
episode_steps	200
actor_learning_rate	0.0 01
episodes_per_update	3
gamma	0.95

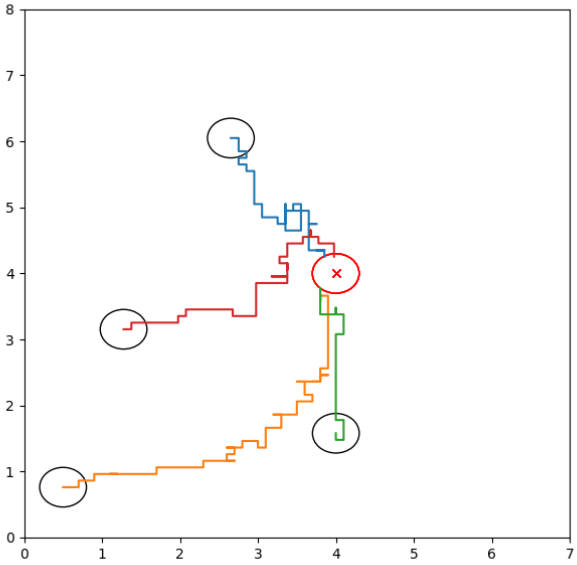
Τέλος, μια ακόμα αύξηση της πολυπλοκότητας του προβλήματος είναι η δυνατότητα κίνησης(μοντελοποίηση ταχύτητας τύπου 2), όχι μόνο προς τις τέσσερις βασικές κατευθύνσεις ξεχωριστά, αλλά και με συνδυασμό αυτών. Και σε αυτή την περίπτωση, ο πράκτορας ανταποκρίθηκε ικανοποιητικά. Στον Πίνακα 15 φαίνονται οι παράμετροι για όλα τα κάτωθι πειράματα.

Πείραμα 13

Πρόκειται για το πρόβλημα τυχαίας εκκίνησης και τυχαίου στόχου. Ο συγκεκριμένος αλγόριθμος έδωσε εξαιρετικά αποτελέσματα και συγκεκριμένα κατά την επαλήθευση σε 1000 τυχαία επεισόδια, είχε 98.5% επιτυχία. Ορισμένες από τις τροχιές των επεισοδίων της επαλήθευσης παρουσιάζονται στο Σχήμα 36. Είναι εμφανές πως οι τροχιές είναι αισθητά καλύτερες από αυτές του αλγορίθμου REINFORCE.

Πείραμα 14

Στο συγκεκριμένο πείραμα, όπως ήδη αναφέραμε, έχουμε προσθέσει μια στοχαστικότητα στην ταχύτητα. Η στοχαστικότητα αυτή συνεισφέρει ταυτόχρονα και στις δύο κατευθύνσεις. Δηλαδή, εάν ο πράκτορας κινείται προς την κατεύθυνση του $x^{'}x$, πέραν της αλλοίωσης αυτής της ταχύτητας, θα προκληθεί και μια μικρή μετατόπιση ως προς τον άξονα $y^{'}y$. Το μοντέλο απέδωσε ικανοποιητικά και σε πείραμα αξιολόγησης 1000 επεισοδίων, είχε 98.7% επιτυχία. Στο Σχήμα 37 παρουσιάζονται ενδεικτικά, ορισμένες τροχιές του οχήματος κατά την αξιολόγησή του.



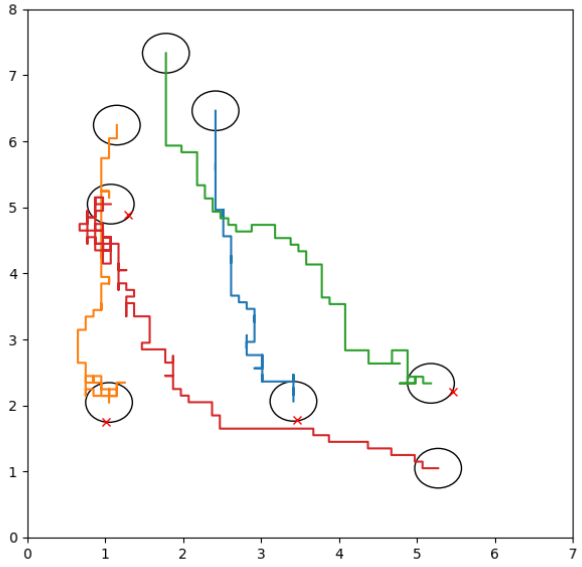
Σχήμα 34: Τροχιές αυτόνομου οχήματος για το Πείραμα 11

Πίνακας 14: Παράμετροι Πειράματος 12

Παράμετροι	**REINFORCE**
reward_type	Σχετική Απόσταση από τον Στόχο
goal_reward	50
episode_steps	200
actor_learning_rate	0.0 01
episodes_per_update	3
gamma	0.95

Πείραμα 15

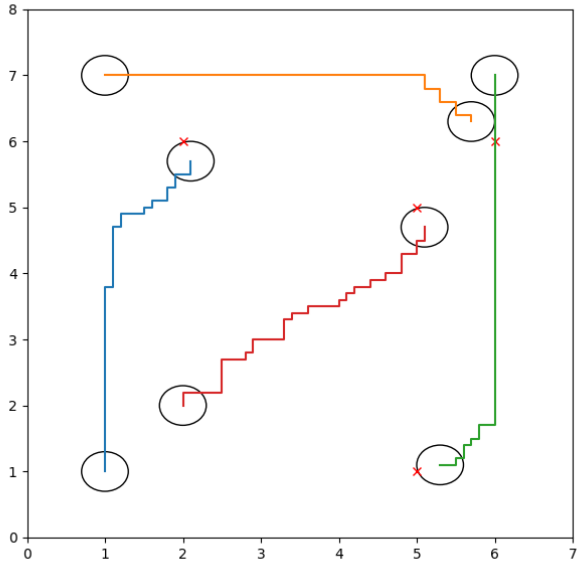
Το Πείραμα 15, αποτελεί το τελευταίο και πιο πολύπλοκο πείραμα που εκτελέστηκε με επιτυχία. Συγκεκριμένα, πρόκειται για ένα πρόβλημα τυχαίας εκκίνησης και τυχαίου επιθυμητού στόχου, με στοχαστικό σφάλμα στην ταχύτητα, και μεγαλύτερη ελευθερία κινήσεων. Συγκεκριμένα, ο πράκτορας πλέον αντί για τέσσερις επιλογές ταχυτήτων έχει οκτώ. Μπορεί, δηλαδή, να κινηθεί και διαγωνίως στην κατεύθυνση των διχοτόμων των τεσσάρων βασικών ταχυτήτων και με μέτρο ίσο με αυτών. Στο Σχήμα 38 παρουσιάζονται ορισμένες τροχιές του αλγορίθμου. Ο αλγόριθμος έδωσε ποσοστό επιτυχίας ίσο με 97.3%.



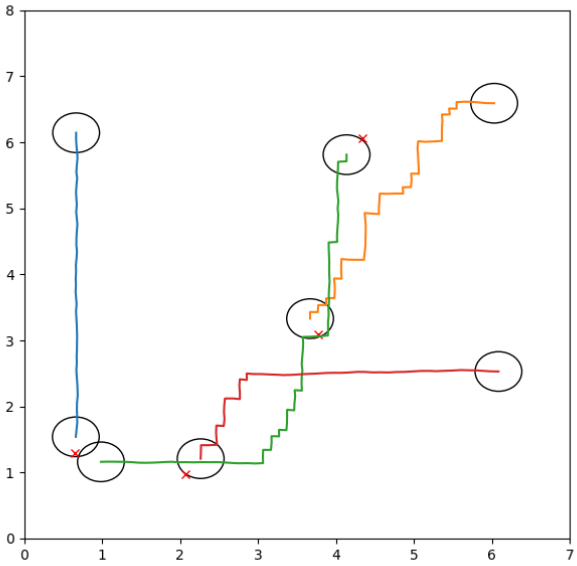
Σχήμα 35: Τροχιές αυτόνομου οχήματος για το Πείραμα 12

Πίνακας 15: Παράμετροι Πειραμάτων A2C

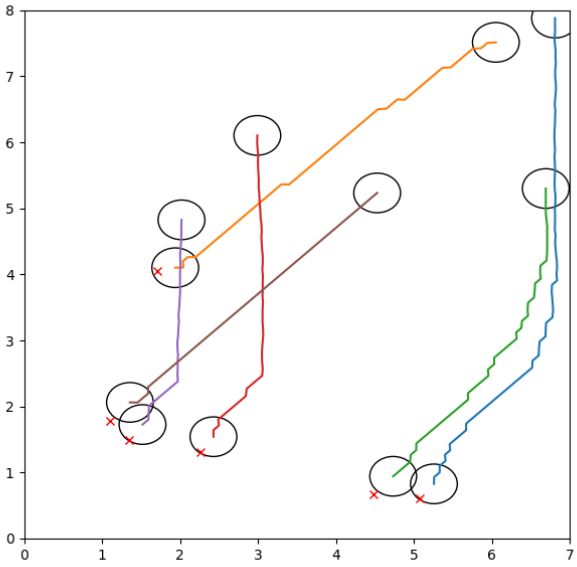
Παράμετροι	**A2C**
reward_type	Σχετική Απόσταση από τον Στόχο
goal_reward	50
episode_steps	200
actor_learning_rate	0.0001
critic_learning_rate	0.001
gamma	0.95



Σχήμα 36: Τροχιές αυτόνομου οχήματος για το Πείραμα 13



Σχήμα 37: Τροχιές αυτόνομου οχήματος για το Πείραμα 14



Σχήμα 38: Τροχιές αυτόνομου οχήματος για το Πείραμα 15

Κεφάλαιο 5

Συμπεράσματα

Συνοψίζοντας, στην συγκεκριμένη εργασία σχεδιάστηκαν τρεις διαφορετικοί αλγόριθμοι (DQN ,REINFORCE, A2C) οι οποίοι κατάφεραν να επιλύσουν μέχρι και το πρόβλημα τυχαίας αρχικής και τελικής θέσης, σε άδειο χάρτη. Στην συνέχεια, για τον αλγόριθμο A2C, ο οποίος όπως ήταν αναμενόμενο έδωσε τα βέλτιστα αποτελέσματα στο προηγούμενο πρόβλημα, μελετήθηκαν ορισμένα πιο δύσκολα προβλήματα. Ορισμένα από αυτά, εκπαιδεύτηκαν με επιτυχία ενώ κάποια άλλα απέτυχαν. Τα πετυχημένα πειράματα παρουσιάστηκαν αναλυτικά στο Κεφάλαιο 4.

Σε αυτό το κεφάλαιο θα συγκριθούν, αρχικά, τα αποτελέσματα των τριών αλγορίθμων για το γενικό πρόβλημα σε χάρτη χωρίς εμπόδια. Στην συνέχεια, θα αναλυθούν τα μη επιτυχώς εκπαιδευμένα μοντέλα και οι λόγοι που εκτιμούμε ότι συνέβη αυτό. Τέλος, θα γίνει αναφορά σε ιδέες και κατευθύνσεις εξέλιξης των πειραμάτων και μελλοντικής έρευνας.

5.1 Απόδοση Αλγορίθμων

Το κυρίως πρόβλημα που μελετήθηκε στην εργασία αυτή, ήταν η πλοήγηση ενός οχήματος σε ένα επίπεδο συνεχούς χώρου χωρίς εμπόδια. Κατά την πλοήγηση του, θα έπρεπε να είναι σε θέση να προσεγγίσει κάποιον τελικό στόχο (οπουδήποτε και αν βρισκόταν στο επίπεδο), γνωρίζοντας την θέση του, και ανεξαρτήτως που θα βρισκόταν κατά την εκκίνηση του επεισοδίου.

Για την επίλυση της πρόκλησης αυτής σχεδιάστηκαν τρεις ευφυείς πράκτορες βασιζόμενοι στις τεχνικές DQN, REINFORCE και A2C, αντίστοιχα. Όλοι τους κατά την εκπαίδευση, συνέκλιναν στην συνθήκη τερματισμού. Στην συνέχεια, όμως, έγινε επαλήθευση των εκπαιδευμένων μοντέλων και παρατηρήθηκε πως είχαν διαφορετική συμπεριφορά. Η αξιολόγηση έγινε σε ένα δείγμα 1000 επεισοδίων όπου η κατηγοριοποίηση της τροχιάς έγινε ως εξής. Εάν το όχημα φτάνει στον τελικό στόχο με τον ελάχιστο σχεδόν αριθμό βημάτων (με τον όρο '*ελάχιστο σχεδόν*', εννοούμε απόκλιση $\pm 5\%$ του ελάχιστου αριθμού βημάτων), τότε θεωρούμε πως ο πράκτορας πέτυχε τον στόχο του με βέλτιστο τρόπο(*optimal*). Η δεύτερη κατηγορία, είναι η περίπτωση όπου ο πράκτορας πέτυχε τον στόχο αλλά όχι με βέλτιστο τρόπο(*success*). Τέλος, η τελευταία κατηγορία αφορά την περίπτωση που ο πράκτορας απέτυχε να φτάσει στον στόχο(*failure*). Τα αποτελέσματα παρουσιάζονται στο Σχήμα 39

Στα αποτελέσματα του αλγορίθμου DQN το 45.6% είναι βέλτιστες τροχιές, το 36.9% επιτυχίες και το 16.5% αποτυχίες. Παρατηρείται μεγάλο ποσοστό αποτυχιών δεδομένου πως κατά την εκπαίδευση είχε 200 συνεχόμενες επιτυχίες. Ο λόγος που το μοντέλο έχει τόσο μεγάλο ποσοστό αποτυχιών, είναι το φαινόμενο *'Των κυκλικών Τροχιών'*, κατά τις οποίες ο πράκτορας επισκέπτεται μια θέση που έχει βρεθεί και στο παρελθόν. Σε αυτή την περίπτωση, λόγω της ντετερμινιστικής άπληστης πολιτικής που ακολουθεί, δεν μπορεί να ξεφύγει από την επαναλαμβανόμενη τροχιά.

Ένα εύλογο ερώτημα είναι, πώς κατά την εκπαίδευση έφτασε σε 200 συνεχόμενες επιτυχίες χωρίς να πέσει στο φαινόμενο των *'Των κυκλικών Τροχιών'*. Ο λόγος που δεν πέφτει σε κυκλικές τροχιές κατά την εκπαίδευση είναι διότι η ανανέωση των βαρών του νευρωνικού δικτύου γίνεται μετά από κάθε βήμα κίνησης. Συνεπώς, όταν πλέον ο πράκτορας θα έχει αποκτήσει την γενική εικόνα ότι πρέπει να κινηθεί προς τον στόχο, τότε σε περίπτωση που πέσει σε ατέρμονη επανάληψη, μετά από έναν αριθμό κινήσεων θα έχουν αλλάξει τα βάρη με τέτοιον τρόπο, όπου θα διαλέξει σε μια κατάσταση διαφορετική πράξη.

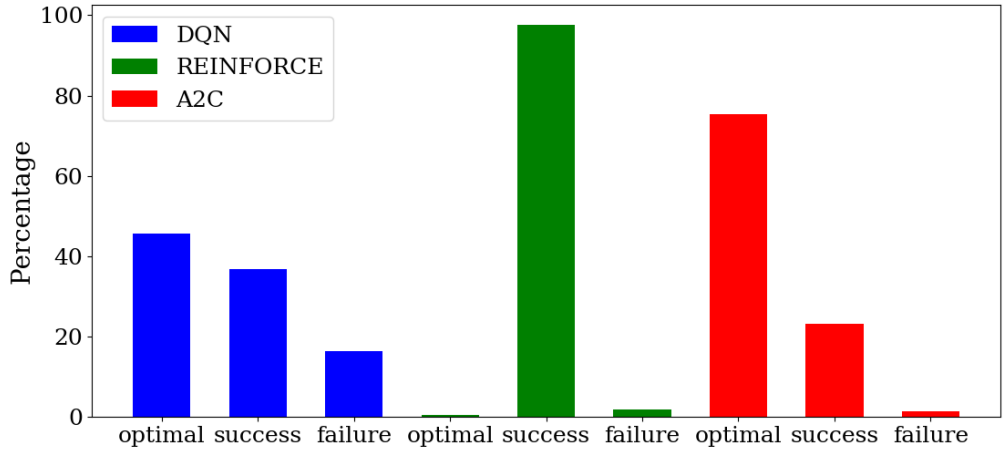
Στον αλγόριθμο REINFORCE είναι εμφανές πως έχει εξαιρετικά ποσοστά επιτυχίας (97.6%), αλλά μηδαμινά ποσοστά βέλτιστων διαδρομών (0.5%). Μια ερμηνεία μπορεί να είναι πως τα 200 συνεχόμενα επιτυχημένα επεισόδια, δεν είναι αρκετά για να εκπαιδευτεί το μοντέλο. Στην πραγματικότητα, όμως, η αύξηση των συνεχόμενων επιτυχιών δεν βελτιώνει το αποτέλεσμα. Συνεπώς, είναι διαφορετικός ο λόγος που δεν συγκλίνει σε βέλτιστη πολιτική.

Αρχικά, ο λόγος που ο πράκτορας δεν φτάνει με βέλτιστο τρόπο στον στόχο είναι πως σε κάθε κατάσταση, οι πιθανότητες επιλογής των πράξεων δεν είναι κατανεμημένες ούτως ώστε μια μόνο δράση να είναι κοντά στην μονάδα και οι υπόλοιπες κοντά στο μηδέν. Συνεπώς, είναι αρκετά πιθανό ακόμα και στην ίδια κατάσταση ο πράκτορας να λαμβάνει συχνά διαφορετικές δράσεις. Στην συνέχεια, θα δοθεί μια εξήγηση αυτού του φαινομένου.

Λόγω της εκπαίδευσης για ολόκληρα επεισόδια, υπάρχει μεγάλη συσχέτιση στα δεδομένα εκπαίδευσης ανά εποχή. Επομένως είναι εξαιρετικά δύσκολο για τον πράκτορα να αποσυμπλέξει την συνεισφορά κάθε κίνησης με την ανταμοιβή της. Αντίθετα αναπροσαρμόζει τα βάρη του νευρωνικού δικτύου για την συνολική αξία όλου του επεισοδίου. Συνεπώς, είναι πιθανό δύο διαφορετικά επεισόδια, να αλλάζουν τα βάρη με τέτοιο τρόπο που το ένα να αναιρεί την αύξηση της πιθανότητας για μια πράξη σε κάποια κατάσταση ή αντίστοιχα να αυξάνει την πιθανότητα μια άλλης δράσης.

Συμπερασματικά, για να επιτευχθεί η σύγκλιση κοντά στην μονάδα της πιθανότητας μιας δράσης για κάθε κατάσταση, θα έπρεπε για πολύ μεγάλο αριθμό αρχικών και τελικών θέσεων να υπάρξουν πολλαπλά επεισόδια κατά την εκπαίδευση. Αυτό είναι ασύμφορο σε περιβάλλον συνεχούς χώρου έως και αδύνατο. Σε κάθε περίπτωση αποκλίνει από την στρατηγική επίλυσης τέτοιων προβλημάτων αποτελεσματικά.

Εν κατακλείδι, ο πράκτορας A2C είναι φανερό πως παρουσιάζει τα καλύτερα αποτελέσματα, συγκριτικά με τα άλλα δύο μοντέλα. Συγκεκριμένα το ποσοστό βέλτιστων διαδρομών είναι 75.3% ενώ το ποσοστό απλών επιτυχιών είναι 23.3%. Επίσης είναι σημαντικό να σημειωθεί πως σε αντίθεση με τον αλγόριθμο REINFORCE, οι περιπτώσεις για τις οποίες ο A2C πράκτορας δεν συγκλίνει με βέλτιστο τρόπο δεν διαφέρει και πολύ από την πορεία με τα ελάχιστα βήματα.

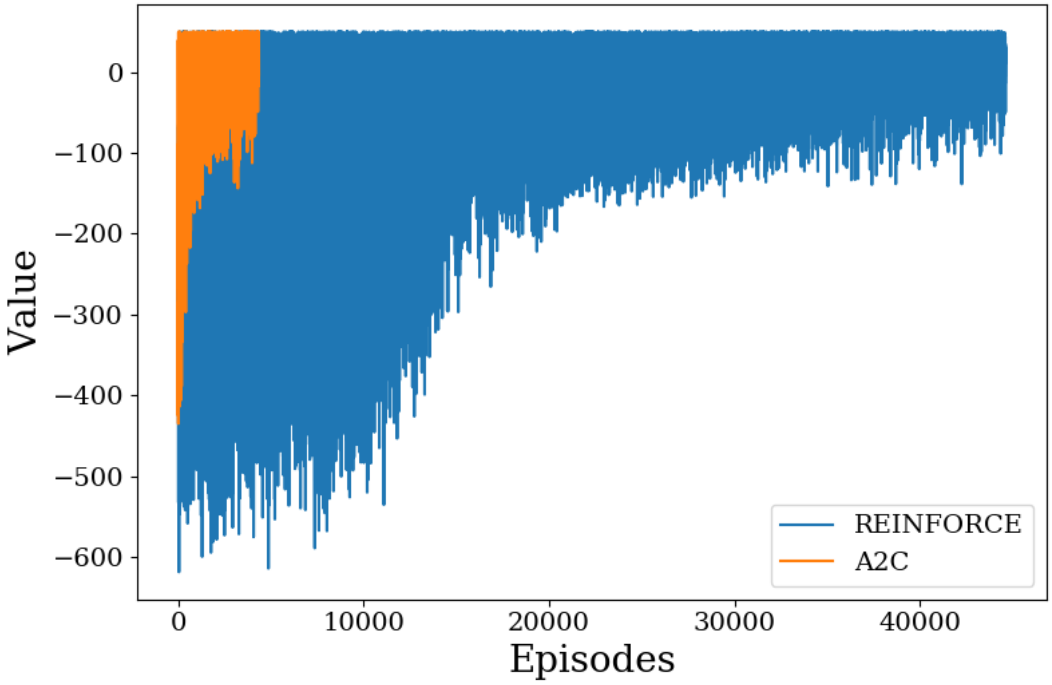


Σχήμα 39: Στατιστικά απόδοσης των μοντέλων

Μια ακόμα παράμετρος που καθιστά τον αλγόριθμο Δράστη - Κριτή εξαιρετικά αποδοτική προσέγγιση του προβλήματος είναι η ταχύτητα σύγκλισης. Συγκεκριμένα, θα συγκριθεί η ταχύτητα των δύο πρακτόρων (REINFORCE, A2C) να εκπαιδευτούν στο ίδιο πρόβλημα με την ίδια συνθήκη τερματισμού. Στο Σχήμα 40 παρουσιάζουμε την συνάρτηση της συνολικής αξίας ανά επεισόδιο καθ' όλη την διάρκεια της εκπαίδευσης.

Από το διάγραμμα μπορεί να παρατηρήσει κανείς εύκολα την κατά πολύ μεγαλύτερη ταχύτητα σύγκλισης σε σχέση με τον αλγόριθμο REINFORCE. Συγκεκριμένα, ο πράκτορας A2C χρειάστηκε

4353 επεισόδια, σε αντίθεση με τον πράκτορα REINFORCE που χρειάστηκε 44618. Επιπλέον, από το διάγραμμα, μπορούμε να δούμε την σταδιακή βελτίωση της συνολικής αξίας του κάθε επεισοδίου, καθώς προχωρά η εκπαίδευση.



Σχήμα 40: Γραφικές Παραστάσεις Αξίας - Επεισοδίου

5.2 Μοντελοποίηση πρακτόρων με ταχύτητα Τύπου 2

Κατά την διαδικασία των πειραμάτων, έγιναν δεκάδες προσπάθειες να επιλυθεί το πρόβλημα τυχαίας αρχικής και τελικής θέσης με μοντελοποίηση ταχυτήτων τύπου 2. Κανένα πείραμα, όμως, δεν συνέκλινε κατά την εκπαίδευση και για τον λόγο αυτό δεν παρουσιάστηκαν αποτελέσματα στο Κεφάλαιο 4, ακόμα και αποτυχημένα, καθώς θεωρήθηκε πως δεν έχουν κάποια αξία. Οι προσπάθειες έγιναν ως προς τον πειραματισμό των ανταμοιβών.

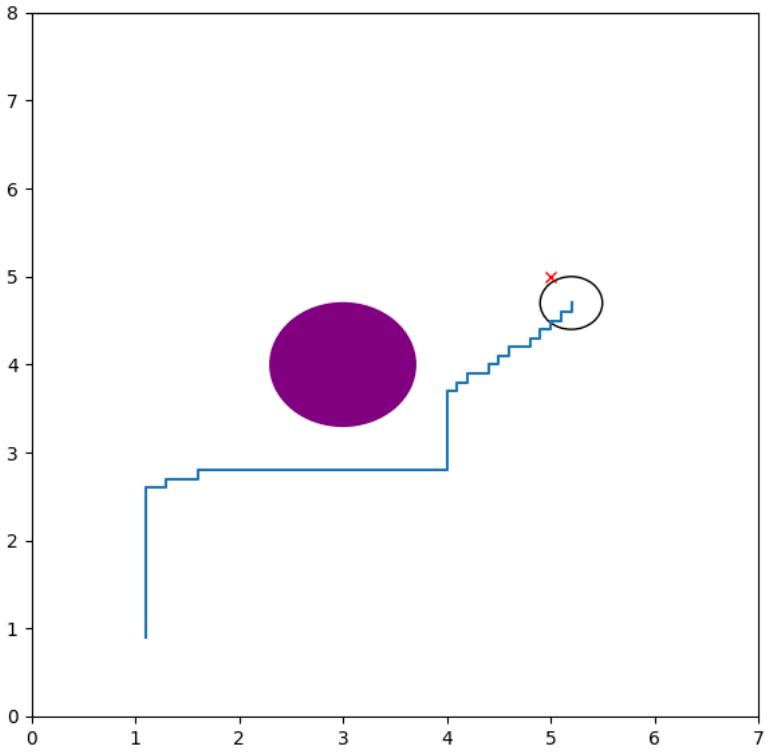
Αρχικά, έγινε προσπάθεια να χρησιμοποιηθεί η μοντελοποίηση ανταμοιβών *'Σχετικής απόστασης από τον στόχο'*. Εκ πρώτης ανάλυσης φαίνεται αρκετή για να επιλύσει το πρόβλημα. Η αστοχία όμως ήταν στο γεγονός πως η περιστροφή του οχήματος χωρίς να αλλάζει θέση δεν άλλαζε την ανταμοιβή. Συνεπώς δεν άλλαζε η αξία της επιλογής της περιστροφής σε καμία κατάσταση. Αυτό βέβαια δεν θα έπρεπε να αποτελεί αναγκαστικά πρόβλημα σύγκλισης καθώς δεν είναι ανάγκη μια πράξη να έχει άμεση ανταμοιβή για να θεωρηθεί ότι θα αυξήσει την συνολική αξία. Συμπερασματικά, όμως, με την συγκεκριμένη μοντελοποίηση ταχυτήτων δεν επιτεύχθηκε σύγκλιση.

Στην συνέχεια, δοκιμάστηκε η μοντελοποίηση της *'Σχετικής Γωνίας'*. Η συγκεκριμένη επιλογή κατάφερε να επιλύσει προβλήματα σταθερής αρχικής και τελικής θέσης, αλλά σε κάθε άλλη περίπτωση απέτυχε να δώσει αποτελέσματα. Αυτό ερμηνεύεται καθώς, χρειάζεται πολλές επιτυχίες για να αρχίσει να δημιουργεί την σύνδεση της θέσης του στόχου με την τιμή που λαμβάνει στην είσοδο. Διαφορετικά, επειδή η αλλαγή θέσης δεν συμβάλει στην ανταμοιβή, η επιλογή την γραμμικής ταχύτητας θα γινόταν λιγότερο συχνά. Στην περίπτωση της σταθερής αρχικής και τελικής θέσης, ο αριθμός των *'τυχαίων'* επιτυχιών ήταν αρκετός ώστε να συγκλίνει στην βέλτιστη λύση. Στην γενική περίπτωση, όμως, όχι.

Τέλος, δοκιμάστηκε η μοντελοποίηση ανταμοιβών, η οποία συνδυάζει την *'Σχετική απόσταση από τον στόχο'* και την *'Σχετική Γωνία'*. Σε αυτή την περίπτωση, παρόλο που αρχικά προβάλλεται ως ικανή να αντιμετωπίσει το πρόβλημα, κατά την διαδικασία των πειραμάτων, οδηγούσε τον πράκτορα σε τροχιές που δεν είχαν κάποιο νόημα στην επίτευξη του στόχου.

5.3 Η αποτυχία υλοποίησης της αποφυγής εμποδίων

Για το πρόβλημα σε χάρτη με εμπόδια, επίσης, δεν επετεύχθη η σύγκλιση κάποιου μοντέλου ή αν συνέκλινε, αυτό οφείλεται σε σύμπτωση λόγω της πολύ αραιής κατανομής των εμποδίων (Σχήμα 41). Σε αυτή την περίπτωση, η τροχιά που ακολουθεί το όχημα είναι σχεδόν βέλτιστη. Αυτό σημαίνει πως πιθανώς να επέλεγε την ίδια διαδρομή και χωρίς την ύπαρξη του εμποδίου. Με απλά λόγια, αν τα εμπόδια που υπάρχουν στον χάρτη δεν αναγκάζουν τον πράκτορα να αποκλίνει από την ιδανική πορεία του, τότε δεν μπορούμε να είμαστε βέβαιοι πως το μοντέλο έχει εκπαιδευτεί προσπαθώντας να τα αποφύγει ενώ προσεγγίζει τον στόχο, ταυτόχρονα.



Σχήμα 41: Τροχιά πράκτορα DQN, με ύπαρξη εμποδίου

Σε κάθε περίπτωση θεωρούμε πως ο λόγος που δεν μπόρεσαν οι αλγόριθμοι να δώσουν αποτελέσματα οφείλεται στην φύση του προβλήματος συνεχούς χώρου. Σε ένα διακριτοποιημένο περιβάλλον, με έλλειψη γνώσης του χάρτη, μετά από πολλές προσπάθειες θα ευρισκόταν κάποιο μονοπάτι προσέγγισης στόχου. Στο παρόν πρόβλημα είδαμε πως αυτό είναι αδύνατο.

Συμπερασματικά, θεωρείται αναμενόμενη η αποτυχία μοντελοποίησης των εμποδίων χωρίς καμία μεταβλητή εισόδου η οποία να δίνει πληροφορία για την θέση των εμποδίων στον πράκτορα σχετικά με την θέση στην οποία βρίσκεται. Συνεπώς, θα έπρεπε στην είσοδο του νευρωνικού δικτύου, να υπάρχει μια ένδειξη αισθητήρα για την σχετική θέση των εμποδίων ως προς το όχημα. Για παράδειγμα, αισθητήρα laser scanner, εικόνα ή βίντεο.

5.4 Κατεύθυνση Μελλοντικής Έρευνας

Όπως έχει ήδη αναφερθεί, το πρόβλημα μοντελοποιήθηκε με την εξής λογική. Η πιο απλή περίπτωση να αφορά την πλοήγηση σε χώρο χωρίς εμπόδια, με αρχική θέση και τελικό στόχο σταθερά. Αντίθετα, η πιο γενική περίπτωση να αφορά, την πλοήγηση σε άγνωστο χάρτη με εμπόδια στατικά και δυναμικά, τυχαίας αρχικής θέσης και τυχαίου επιθυμητού στόχου. Συνεπώς επέκταση της παρούσας εργασίας αποτελεί η έρευνα και δημιουργία μοντέλων σταδιακά προς την γενικότερη κατεύθυνση. Συγκεκριμένα, θα δοθούν τρεις βασικές κατευθύνσεις επέκτασης των μοντέλων.

Πρώτη κατεύθυνση που προτείνεται, αφορά την μελέτη και επίλυση των ίδιων προβλημάτων με την μοντελοποίηση ταχυτήτων *τύπου 2*. Στην συγκεκριμένη εργασία, έγιναν πολλές προσπάθειες αλλά καμία δεν απέδωσε. Επίσης, πρόκειται για την περίπτωση που είναι περισσότερο *'θολή'* από τις υπόλοιπες ως προς τον τρόπο με τον οποίο θα προσπεραστεί η δυσκολία της. Η μελέτη και αντιμετώπιση αυτού του προβλήματος παρουσιάζει ιδιαίτερο ενδιαφέρον καθώς πραγματικά ρομποτικά οχήματα κινούνται με την λογική *'τύπου 1'* πολύ πιο συχνά από την *'τύπου 2'*. Συνεπώς, μετά την εκπαίδευση σε περιβάλλον προσομοίωσης, υπάρχει η δυνατότητα να εφαρμοσθεί και σε πραγματική διάταξη.

Επόμενο πεδίο μελέτης που παρουσιάζει ενδιαφέρον, είναι η αντιμετώπιση σε χάρτη με εμπόδια. Σε αυτή την περίπτωση, θεωρούμε, όπως ήδη αναφέρθηκε, πως η αντιμετώπιση μόνο μέσα από τις αρνητικές ανταμοιβές κατά την σύγκρουση δεν αποτελεί καλή προσέγγιση σε προβλήματα συνεχούς κίνησης. Αντίθετα, το πρόβλημα εστιάζεται στην πληροφορία εισόδου και όχι στις παραμέτρους εκπαίδευσης. Επομένως, προτείνεται η προσπάθεια επίλυσης προβλημάτων σε χώρο με εμπόδια, με χρήση αισθητήρων που να δίνουν μια εικόνα στο όχημα για τον προσανατολισμό των εμποδίων στον χώρο, σχετικά με την θέση που βρίσκεται εκείνη την στιγμή. Με αυτές τις τιμές στην είσοδο, πιστεύουμε πως το μοντέλο θα είναι σε θέση να συνδέσει με πιο γενικό τρόπο την εξάρτηση της δράσης που πρέπει να επιλέξει με την αποφυγή συγκρούσεων, σαν πιο γενικό κανόνα.

Τέλος, μια ακόμα προσέγγιση και ίσως η πιο ενδιαφέρουσα είναι η μοντελοποίηση πολλών αυτόνομων πρακτόρων στον χώρο με διαφορετικούς στόχους ο κάθε ένας. Πλέον, δηλαδή πρόκειται για πολυπρακτορικά συστήματα στα οποία οι πράκτορες κινούνται ταυτόχρονα. Σε ένα ακόμα πιο εξελιγμένο επίπεδο θα μπορούσαν πολλά οχήματα να συνυπάρχουν σε χώρο με εμπόδια, επιλύοντας με αυτό τον τρόπο την πιο γενική περίπτωση του προβλήματος.

Βιβλιογραφία

[1] D. A. Pomerleau, "Efficient training of artificial neural networks for autonomous navigation," *Neural Computation*, vol. 3, no. 1, pp. 88–97, 1991.

[2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[3] I. Engedy and G. Horváth, "Artificial neural network based mobile robot navigation," in *2009 IEEE International Symposium on Intelligent Signal Processing*, IEEE, 2009, pp. 241–246.

[4] L. Sun, Y. Luo, X. Ding, and L. Wu, "Path planning and obstacle avoidance for mobile robots in a dynamic environment," *The Open Automation and Control Systems Journal*, vol. 6, pp. 77–83, 2014.

[5] S. Sugathan, B. V. Sowmya Shree, M. R. Warrier, and C. M. Vidhyapathi, "Collision avoidance using neural networks," in *Materials Science and Engineering Conference Series*, ser. Materials Science and Engineering Conference Series, vol. 263, Nov. 2017.

[6] K.-H. Chi and M.-F. R. Lee, "Obstacle avoidance in mobile robot using neural network," in *2011 International Conference on Consumer Electronics, Communications and Networks (CECNet)*, IEEE, 2011, pp. 5082–5085.

[7] T. A. Zimmerman, "Neural network based obstacle avoidance using simulated sensor data," in *ASEE 2014 Zone I Conference*, 2014.

[8] K. Macek, I. PetroviC, and N. Peric, "A reinforcement learning approach to obstacle avoidance of mobile robots," in *7th International Workshop on Advanced Motion Control. Proceedings (Cat. No. 02TH8623)*, IEEE, 2002, pp. 462–466.

[9] E. S. Low, P. Ong, and K. C. Cheah, "Solving the optimal path planning of a mobile robot using improved Q-learning," *Robotics and Autonomous Systems*, vol. 115, pp. 143–161, 2019.

[10] M. A. K. Jaradat, M. Al-Rousan, and L. Quadan, "Reinforcement based mobile robot navigation in dynamic environment," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 1, pp. 135–149, 2011.

[11] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 3052–3059.

[12] S. Lange, M. Riedmiller, and A. Voigtländer, "Autonomous reinforcement learning on raw visual input data in a real world application," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2012, pp. 1–8.

[13] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 31–36.

[14] L. Xie, S. Wang, A. Markham, and N. Trigoni, "Towards monocular vision based obstacle avoidance through deep reinforcement learning," *arXiv preprint arXiv:1706.09829*, 2017.

[15] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2017, pp. 3357–3364.

[16] J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard, "Deep reinforcement learning with successor features for navigation across similar environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 2371–2378.

[17] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, *et al.*, "Learning to navigate in complex environments," *arXiv preprint arXiv:1611.03673*, 2016.

[18] P. Long, T. Fanl, X. Liao, W. Liu, H. Zhang, and J. Pan, "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 6252–6259.

[19] F. Chen, S. Bai, T. Shan, and B. Englot, "Self-Learning Exploration and Mapping for Mobile Robots via Deep Reinforcement Learning," in *AIAA Scitech 2019 Forum*, 2019, p. 0396.

[20] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep Reinforcement Learning Robot for Search and Rescue Applications: Exploration in Unknown Cluttered Environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 610–617, 2019.

[21] L. Tai and M. Liu, "Towards cognitive exploration through deep reinforcement learning for mobile robots," *arXiv preprint arXiv:1610.01733*, 2016.

[22] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[23] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[24] E. Alpaydin, *Introduction to machine learning*. MIT press, 2009.

[25] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[27] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[28] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[29] S. Katagiri, *Handbook of neural networks for speech processing*. Artech House Boston, 2000, vol. 171.

[30] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[31] A. Margatina, "Transfer learning and attention-based conditioning methods for natural language processing," 2019.

[32] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[33] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[34] P. A. Gagniuc, *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons, 2017.

[35] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*, 2. Athena scientific Belmont, MA, 1995, vol. 1.

[36] R. Bellman *et al.*, "The theory of dynamic programming," *Bulletin of the American Mathematical Society*, vol. 60, no. 6, pp. 503–515, 1954.

[37] C. Dann, G. Neumann, and J. Peters, "Policy evaluation with temporal differences: A survey and comparison," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 809–883, 2014.

[38] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[39] N. Tziortziotis, "Machine learning for intelligent agents," PhD thesis, Aristotle University of Thessaloniki, 2015.

[40] G. Tesauro, "Temporal difference learning and TD-Gammon," *Communications of the ACM*, vol. 38, no. 3, pp. 58–68, 1995.

[41] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[42] L.-J. Lin, "Reinforcement learning for robots using neural networks," Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, Tech. Rep., 1993.

[43] M. P. Deisenroth, G. Neumann, J. Peters, *et al.*, "A survey on policy search for robotics," *Foundations and Trends® in Robotics*, vol. 2, no. 1–2, pp. 1–142, 2013.

[44] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[45] R. J. Rossi, *Mathematical Statistics: An Introduction to Likelihood Based Inference*. John Wiley & Sons, 2018.

[46] J. Peters and S. Schaal, "Policy gradient methods for robotics," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2006, pp. 2219–2225.

[47] D. Aberdeen, "POMDPs and policy gradients," *Proceedings of the Machine Learning Summer School (MLSS), Canberra, Australia*, 2006.