Εθνικό Μετσόβιο Πολυτεχνείο

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΣΥΣΤΗΜΑΤΑ ΑΥΤΟΜΑΤΙΣΜΟΥ
ΚΑΤΕΥΘΥΝΣΗ ΣΥΣΤΗΜΑΤΑ ΚΑΤΑΣΚΕΥΩΝ ΚΑΙ ΠΑΡΑΓΩΓΗΣ.

Διπλωματική Εργασία

Του φοιτητή του διατμηματικού προγράμματος μεταπτυχιακών σπουδών Συστήματα Αυτοματισμού

Παπαχρήστου Νικόλαου

Αριθμός Μητρώου: 02120117

ΘΕΜΑ

Ανάπτυξη συστημάτων αυτομάτου ελέγχου συνδυάζοντας ελεγκτές PID και τεχνολογίες ενισχυτικής μάθησης

Επιβλέπων:

Χαράλαμπος Σαρίμβεης
Καθηγητής

Αριθμός Διπλωματικής Εργασίας:

Αθήνα, Νοέμβριος 2022

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η Διπλωματική Εργασία με θέμα

Ανάπτυξη συστημάτων αυτομάτου ελέγχου συνδυάζοντας ελεγκτές PID και τεχνολογίες ενισχυτικής μάθησης

Του φοιτητή του διατμηματικού προγράμματος μεταπτυχιακών σπουδών Συστήματα Αυτοματισμού

Παπαχρήστου Νικόλαου

Αριθμός Μητρώου: 02120117

Παρουσιάστηκε δημόσια και εξετάστηκε στις

……../……../……

Ο Επιβλέπων:
Χαράλαμπος Σαρίμβεης
Καθηγητής

ΘΕΜΑ: Ανάπτυξη συστημάτων αυτομάτου ελέγχου συνδυάζοντας ελεγκτές PID και τεχνολογίες ενισχυτικής μάθησης

Φοιτητής: Παπαχρήστου Νικόλαος
Επιβλέπων: Χαράλαμπος Σαρίμβεης

Περίληψη

Στην παρούσα διπλωματική εργασία μελετάται και αναλύεται η διαδικασία ελέγχου με τη μέθοδο της ενισχυτικής μάθησης (Reinforcement Learning-RL) σε συνδυασμό με PI ελεγκτές του συστήματος ηλεκτρικής κίνησης ενός ηλεκτροκίνητου αυτοκινήτου .Το συγκεκριμένο σύστημα αποτελείται από μια σύγχρονη μηχανή μόνιμου μαγνήτη ,έναν dc-ac μετατροπέα και μια πηγή ρεύματος . Σκοπός μας είναι να κατασκευάσουμε έναν ελεγκτικό μηχανισμό ,του οποίου η λειτουργία θα βασίζεται στον έλεγχο τόσο των στροφών του κινητήρα όσο και των ρευμάτων που τον διαπερνούν .Η διαδικασία ελέγχου περιλαμβάνει την εκπαίδευση του συστήματος ούτως ώστε η απόκριση του να είναι αποδοτική σε ένα ευρύ φάσμα επιλογής επιμέρους βαρών για τους ελεγκτές. Η επίδραση του RL ελεγκτή αφορά μόνο τον εξωτερικό βρόχο ελέγχου του συστήματος .Η ανάλυση του συστήματος και των επιμέρους χαρακτηριστικών του γίνεται αρχικά σε θεωρητικό επίπεδο και ακολούθως προσομοιώνεται στο γραφικό περιβάλλον Simulink της Matlab . Όπως θα δούμε τα αποτελέσματα της προσομοίωσης επιβεβαιώνουν όχι μόνο την αύξηση της απόδοσης του συστήματος με τη χρησιμοποίηση της ενισχυτικής μάθησης αλλά και την ευρωστία του ως προς μια πληθώρα επιλογών κερδών στον PI ελεγκτή του εξωτερικού βρόχου ελέγχου.

Abstract

In this diploma thesis , the control process of an electric car is studied and analyzed with the method of reinforcement leaning in combination with PI controllers. This specific consists of a permanent magnet synchronous motor, a dc-ac converter and a current source. Our goal is to formulate a control mechanism, whose operation will be based on the control of both the engine speed and the currents that penetrate it. The control process incudes the training of the system in order its response to be efficient in broad spectrum of weight choices for the controllers. The effect of the reinforcement learning controller concerns only the outer control loop of the system. Firstly the theoretical analysis of the system and its own specific characteristics are taking place and then they are simulated in the graphic environment of matlab simulink. As we will see, the results of the simulation confirm not only the increase in the performance of the system by using reinforcement learning but also its robustness in terms of a multitude of gain options in the PI controller of the outer control loop.

Πρόλογος

Η ενισχυτική μάθηση αποτελεί ένα ιδιαίτερα ενδιαφέρον πεδίο μελέτης και εφαρμογής της τεχνητής νοημοσύνης. Η χρήση της ενισχυτικής μάθησης στις μέρες μας γίνεται ολοένα και πιο συχνή και αφορά κυρίως έναν ευέλικτο ελεγκτικό μηχανισμό με εφαρμογή σε πολλαπλού τύπου εφαρμογές και συστήματα. Στη παρούσα διπλωματική εργασία γίνεται όχι μόνο θεωρητική προσέγγιση της ενισχυτικής μάθησης αλλά και ανάλυση-σχεδιασμός του ελεγκτικού μηχανισμού που θα διέπει την εύρυθμη λειτουργία του κινητήρα ενός ηλεκτρικού οχήματος, βασισμένος σε αυτή.

Στο **Κεφάλαιο 1**,γίνεται μια ευρεία αναφορά στην ενισχυτική μάθηση, σε επιμέρους εφαρμογές της και στη διαδικασία λήψης απόφασης Marcov.Στη συνέχεια παρουσιάζονται οι συναρτήσεις και οι αλγόριθμοι βάσει των οποίων θεμελιώνεται η θεωρητική προσέγγισή της.

Στο **Κεφάλαιο 2**,γίνεται μια ιστορική αναφορά στα νευρωνικά δίκτυα και παρουσιάζεται η μορφή τους τόσο σε μονοεπίπεδο όσο και σε πολυεπίπεδο πλαίσιο. Έπειτα ακολουθεί η ανάλυση της διαδικασίας backpropagation.

Στο **Κεφάλαιο 3** ,σε αρχικό επίπεδο δίνονται ιστορικά στοιχεία για τον αλγόριθμο ενίσχυσης(reinforce) , ο οποίος αποτελεί βάση όλων των μεταγενέστερων αλγορίθμων ενισχυτικής μάθησης. Στη συνέχεια αναλύεται τι είναι η πολιτική, οι συναρτήσεις στόχου και η πολιτική κλίσης(gradient descent).Τέλος παρουσιάζεται ο συγκεκριμένος αλγόριθμος.

Στο **Κεφάλαιο 4**, παρουσιάζεται η δομή, η λειτουργία και η χρησιμότητα των αλγόριθμων πράκτορα-κριτή (Actor-Critic).Επίσης αναλύεται η συνάρτηση πλεονεκτήματος (advantage function) και ο τρόπος εκπαίδευσής της.

Στο **Κεφάλαιο 5**, παρουσιάζεται το ολοκληρωμένο μοντέλο της μηχανής στο σύγχρονα στρεφόμενο *d-q* πλαίσιο αναφοράς, καθώς και ο σχεδιασμός ενός αποδοτικού συστήματος PI ελέγχου .Ο έλεγχος ακολουθεί τη διαδικασία της γραμμικοποίησης του συστήματός μας και εξάγεται το τελικό μαθηματικό μοντέλο του ελεγκτή.

Στο **Κεφάλαιο 6** ,παρατίθενται η δομική σύσταση του RL ελεγκτή και τα αποτελέσματα της προσομοίωσης του συστήματος στο περιβάλλον του Simulink στη matlab μέσω γραφικών απεικονίσεων . Εν συνεχεία ακολουθούν επιμέρους σχολιασμοί των αποτελεσμάτων αυτών.

Στο **Κεφάλαιο 7** ,εξάγονται τα βασικά συμπεράσματα για τη λειτουργικότητα και την ευρωστία του ελεγκτικού μηχανισμού που χρησιμοποιούμε και προοικονομούμε την μελλοντική χρησιμότητα τέτοιου είδους ελεγκτών .

Ευχαριστίες

Στο σημείο αυτό, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Χαράλαμπο Σαρίμβεη για την εμπιστοσύνη που μου έδειξε με το να αναλάβω αυτό το ενδιαφέρον και αξιόλογο θέμα, αλλά και τη συνολική ως τώρα συνεργασία μας .Επίσης, νομίζω πως το μεγαλύτερο ευχαριστώ το οφείλω στην οικογένειά μου ,στους ανθρώπους εκείνους που με την οικονομική και ψυχολογική υποστήριξή τους όλα αυτά τα χρόνια με βοήθησαν να ολοκληρώσω τις σπουδές μου, πραγματοποιώντας με αυτόν τον τρόπο ένα από τα μεγαλύτερα όνειρά μου και εκπληρώνοντας μια από τις πιο μεγάλες φιλοδοξίες μου.

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1ο

Εισαγωγή στην Ενισχυτική Μάθηση

1.1 Εισαγωγή

Η ιδέα ότι μαθαίνουμε αλληλεπιδρώντας με το περιβάλλον μας είναι πιθανώς η πρώτη που μας έρχεται στο μυαλό όταν σκεφτόμαστε τη φύση της μάθησης. Όταν ένα βρέφος παίζει, κουνάει τα χέρια του ή κοιτάζει, δεν έχει ξεκάθαρο δάσκαλο, αλλά αυτό έχει άμεση αισθητικοκινητική σύνδεση με το περιβάλλον του. Η σύνδεση παράγει πληθώρα πληροφοριών σχετικά με την αιτία και το αποτέλεσμα, τις συνέπειες των πράξεων και για το τι πρέπει να γίνει για να επιτευχθούν οι στόχοι. Σε όλη μας τη ζωή, τέτοιες αλληλεπιδράσεις αποτελούν αναμφίβολα μια σημαντική πηγή γνώσης για το περιβάλλον και τον εαυτό μας. Είτε μαθαίνουμε να οδηγούμε ένα αυτοκίνητο ή να κάνουμε μια συζήτηση, γνωρίζουμε πολύ καλά πώς το περιβάλλον μας ανταποκρίνεται σε αυτό που κάνουμε και επιδιώκουμε να επηρεάσουμε ότι συμβαίνει μέσω της συμπεριφορά μας. Η μάθηση από την αλληλεπίδραση είναι μια θεμελιώδης ιδέα για σχεδόν όλες τις θεωρίες μάθησης και νοημοσύνης.

Σε αυτή τη διπλωματική εργασία διερευνάται μια υπολογιστική προσέγγιση για το σχεδιασμό συστημάτων αυτομάτου ελέγχου με βάση τη μάθηση από την αλληλεπίδραση. Δηλαδή υιοθετείται η προοπτική της επίλυσης του προβλήματος βασισμένη στον κλάδο της τεχνητής νοημοσύνης που ονομάζεται ενισχυτική μάθηση, που επικεντρώνεται στη σταδιακή εκπαίδευση του συστήματος ελέγχου , αλληλεπιδρώντας με το περιβάλλον. [1].

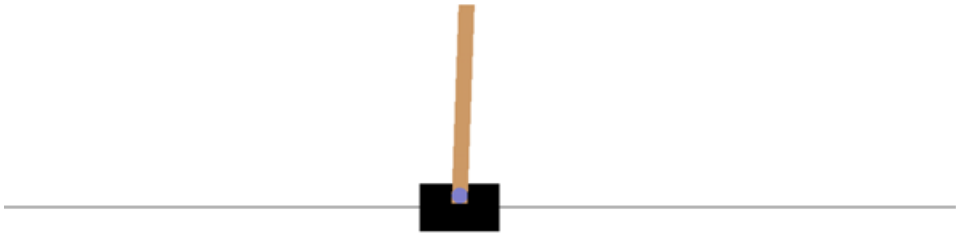
1.2 Ενισχυτική Μάθηση

Η ενισχυτική μάθηση (reinforcement leaning-RL) ασχολείται με την επίλυση διαδοχικής λήψης αποφάσεων προβλημάτων. Πολλά προβλήματα στον πραγματικό κόσμο όπως τα βιντεοπαιχνίδια, τα αθλήματα, η οδήγηση, η βελτιστοποίηση αποθέματος, ο ρομποτικός έλεγχος μπορούν να λυθούν με αυτόν τον τρόπο. Αυτά είναι πράγματα που κάνουν και οι άνθρωποι και οι μηχανές . Όταν λύνουμε αυτά τα προβλήματα, έχουμε έναν σκοπό ή στόχο όπως να κερδίσουμε ένα παιχνίδι, να φτάσουμε με ασφάλεια στον προορισμό μας ή να ελαχιστοποιήσουμε το κόστος κατασκευής προϊόντων. Εμείς αναλαμβάνουμε ενέργειες και λαμβάνουμε σχόλια από το περιβάλλον σχετικά με το πόσο

κοντά είμαστε στην επίτευξη του σκοπού, την τρέχουσα βαθμολογία, την απόσταση από τον προορισμό μας ή την τιμή ανά μονάδα. Η επίτευξη του στόχου που έχουμε θέσει, συνήθως περιλαμβάνει τη λήψη πολλών ενεργειών στη σειρά, με κάθε ενέργεια να επηρεάζει το περιβάλλον γύρω μας. Παρατηρούμε αυτές τις αλλαγές στο περιβάλλον καθώς και την ανατροφοδότηση που λαμβάνουμε πριν αποφασιστεί η επόμενη ενέργεια που θα εφαρμόσουμε. Ως παράδειγμα, περιγράφεται το εξής σενάριο: ένας φίλος μας προκαλεί να ισορροπήσουμε μια ράβδο στο χέρι μας για όσο το δυνατόν περισσότερο χρόνο. Αν δεν έχουμε προσπαθήσει ποτέ στο παρελθόν να ισορροπήσουμε μια ράβδο, οι αρχικές προσπάθειες δεν θα είναι πολύ επιτυχημένες. Στις πρώτες αυτές όμως προσπάθειες, αποκτούμε μια αίσθηση της ισορροπίας της ράβδου μέσω δοκιμής και λάθους ,ενόσω αυτό συνεχίζει να πέφτει.

Αυτά τα λάθη μας επιτρέπουν να συλλέξουμε σημαντικές πληροφορίες και να αποκτήσουμε κάποια διαίσθηση σχετικά με το πώς να ισορροπήσουμε τη ράβδο, πού είναι το κέντρο βάρους της, πόσο γρήγορα και με ποια γωνία γέρνει, κ.λπ. Χρησιμοποιούμε αυτές τις πληροφορίες για να κάνουμε διορθώσεις στις κινήσεις μας, και σταδιακά μπορούμε να ισορροπήσουμε τη ράβδο για περισσότερο χρόνο.

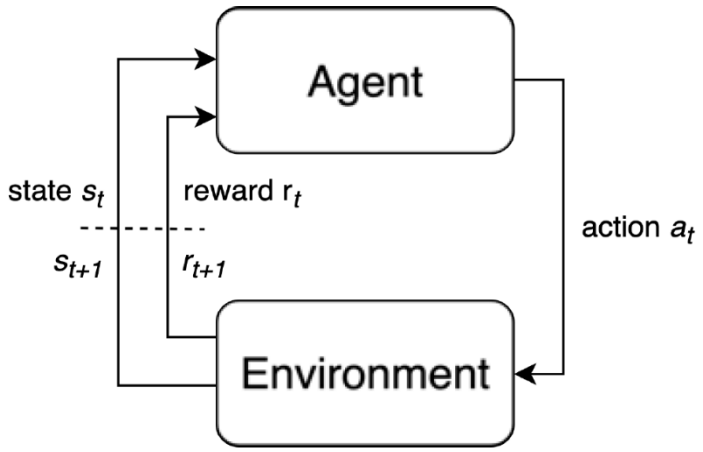
Αυτή η διαδικασία δείχνει με απλοποιημένο τρόπο τις αρχές στις οποίες βασίζεται η ενισχυτική μάθηση. Πιο συγκεκριμένα, αποτελούμε τον "agent"(πράκτορας), και η ράβδος είναι το «περιβάλλον». Το αντίστοιχο σύστημα που μπορούμε να εκπαιδεύσουμε με ενισχυτική μάθηση είναι το CartPole, που φαίνεται στο σχήμα 1.1. Ένας πράκτορας ελέγχει ένα βαγόνι που κινείται κατά μήκος ενός άξονα προκειμένου να εξισορροπήσει έναν πόλο όρθιο για δεδομένο χρονικό διάστημα.



Σχήμα1.1Το CartPole-v0 είναι ένα απλό περιβάλλον παιχνιδιών. Στόχος είναι η εξισορρόπηση ενός πόλου για 200 χρονικά βήματα ελέγχοντας την κίνηση αριστερά-δεξιά ενός βαγονιού.

Η ενισχυτική μάθηση μελετά προβλήματα αυτής της μορφής και μεθόδους με τις οποίες τεχνητοί πράκτορες μαθαίνουν να τα λύνουν. Πρόκειται για ένα πεδίο της τεχνητής νοημοσύνης που βασίζεται στη θεωρία του βέλτιστου ελέγχου και τις διαδικασίες λήψης αποφάσεων Markov (Markov Decision Processes ,MDPs). Το πρόβλημα μελετήθηκε αρχικά από τον Richard Bellman στη δεκαετία του 1950 στο πλαίσιο του δυναμικού προγραμματισμού [2]. Θα δούμε ξανά αυτό το όνομα όταν μελετήσουμε μια διάσημη εξίσωση στην ενισχυτική μάθηση - την εξίσωση Bellman. Τα προβλήματα RL μπορούν να εκφραστούν ως ένα σύστημα που αποτελείται από έναν πράκτορα και ένα περιβάλλον.

Το περιβάλλον παράγει πληροφορίες που περιγράφουν την κατάσταση του συστήματος. Αυτό είναι γνωστό ως περιβάλλον. Ο πράκτορας αλληλεπιδρά με το περιβάλλον παρατηρώντας την κατάσταση για να επιλέξει μια ενέργεια. Το περιβάλλον δεχόμενο τη δράση μεταβαίνει στην επόμενη κατάσταση , ενώ επίσης επιστρέφεται μια ανταμοιβή(reward) στον πράκτορα. Όταν ο κύκλος του (περιβάλλον ⇒δράση ⇒ ανταμοιβή) ολοκληρώνεται, λέμε ότι έχει ολοκληρωθεί ένα βήμα. Ο κύκλος επαναλαμβάνεται μέχρι να ολοκληρωθεί ένα επεισόδιο, για παράδειγμα μια πεπερασμένη σειρά βημάτων. Αυτή ολόκληρη η διαδικασία περιγράφεται από το διάγραμμα βρόχου ελέγχου στο σχήμα 1.2.



Σχήμα 1.2.Βρόχος της ενισχυτικής μάθησης.

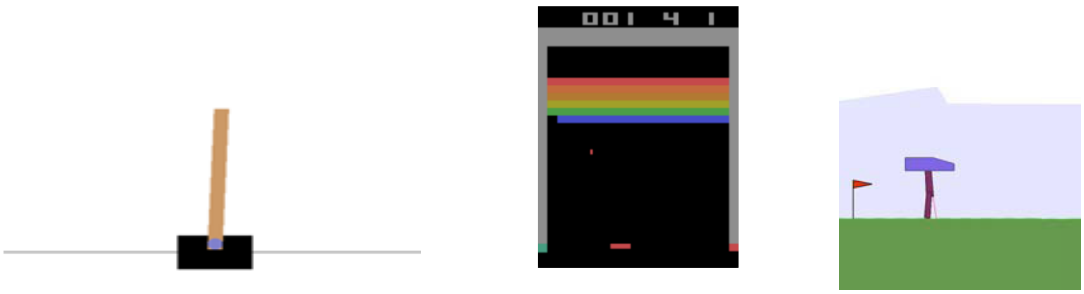
Ονομάζουμε πολιτική τη συνάρτηση επιλογής δράσης ενός πράκτορα. Τυπικά, μια πολιτική είναι μια συνάρτηση που συσχετίζει τις καταστάσεις με τις δράσεις. Μια ενέργεια θα επηρεάσει το περιβάλλον, άρα την κατάσταση που παρατηρεί ο πράκτορας στο επόμενο βήμα. Η αλληλεπίδραση μεταξύ ενός πράκτορα και του περιβάλλοντος συνεχίζεται στο χρόνο - επομένως μπορεί να θεωρηθεί ως μια διαδοχική διαδικασία λήψης

αποφάσεων. Τα προβλήματα ενισχυτικής μάθησης έχουν ως στόχο τον προσδιορισμό της πολιτικής που θα μεγιστοποιήσει τις ανταμοιβές που λαμβάνονται από το περιβάλλον. Ο πράκτορας μαθαίνει να το κάνει αυτό αλληλεπιδρώντας με το περιβάλλον σε μια διαδικασία δοκιμής και σφάλματος και χρησιμοποιεί τα σήματα ανταμοιβής που λαμβάνει για να ενισχύσει τις καλές ενέργειες.

Ο πράκτορας και το περιβάλλον ορίζονται ως αμοιβαία αποκλειόμενα, έτσι ώστε τα όρια μεταξύ της αλλαγής της κατάστασης, της δράσης και της ανταμοιβής να είναι ξεκάθαρα. Για παράδειγμα, όταν οδηγούμε ποδήλατο, μπορούμε να έχουμε πολλαπλούς αλλά εξίσου έγκυρους ορισμούς ενός πράκτορα και ενός περιβάλλοντος. Αν εμείς θεωρούμε ότι ολόκληρο το σώμα μας είναι ο πράκτορας που παρατηρεί το περιβάλλον και παράγει τις μυϊκές κινήσεις ως δράσεις, τότε το περιβάλλον είναι το ποδήλατο και ο δρόμος. Αν εμείς θεωρήσουμε τις ψυχικές μας διαδικασίες ως τον πράκτορα, τότε το περιβάλλον είναι το φυσικό μας σώμα, το ποδήλατο και ο δρόμος, με ενέργειες να είναι τα νευρικά σήματα που αποστέλλονται από τον εγκέφαλό μας στους μύες και καταστάσεις να είναι τα σήματα που αποστέλλονται πίσω στον εγκέφαλό μας.

Ουσιαστικά, ένα ενισχυτικό σύστημα μάθησης είναι ένας βρόχος ελέγχου ανάδρασης όπου ένας πράκτορας και ένα περιβάλλον αλληλεπιδρούν και ανταλλάσσουν σήματα, ενώ ο πράκτορας προσπαθεί να μεγιστοποιήσει το στόχο. Τα σήματα που ανταλλάσσονται συμβολίζονται ως (s_t, a_t, r_t), τα οποία αντιπροσωπεύουν την κατάσταση, τη δράση και την ανταμοιβή, αντίστοιχα, και t υποδηλώνει το χρονικό βήμα στο οποίο συνέβησαν αυτά τα σήματα. Η (s_t, a_t, r_t) πλειάδα ονομάζεται εμπειρία. Ο βρόχος ελέγχου μπορεί να επαναληφθεί χωρίς χρονικό περιορισμό ή να τερματιστεί φτάνοντας είτε μια τερματική κατάσταση είτε σε ένα μέγιστο χρονικό βήμα $t = T$. Ο χρονικός ορίζοντας από $t = 0$ έως όταν το περιβάλλον τερματίζεται ονομάζεται επεισόδιο. Μια τροχιά είναι μια ακολουθία εμπειριών σε ένα επεισόδιο,$\tau = (s_0, a_0, r_0)$, (s_1, a_1, r_1),….Ένας πράκτορας συνήθως χρειάζεται πολλά επεισόδια για να μάθει μια καλή πολιτική, που κυμαίνονται από εκατοντάδες έως εκατομμύρια ανάλογα με την πολυπλοκότητα του προβλήματος.

Ας δούμε τα τρία παραδείγματα περιβαλλόντων ενισχυτικής μάθησης, που εμφανίζονται στο Σχήμα 1.3 και πώς ορίζονται οι καταστάσεις, οι ενέργειες και οι ανταμοιβές. Όλα τα περιβάλλοντα διατίθεται μέσω του OpenAI Gym [3] το οποίο είναι μια βιβλιοθήκη ανοιχτού κώδικα που παρέχει ένα τυποποιημένο σύνολο περιβαλλόντων.



Σχήμα 1.3 Τρία παραδείγματα περιβαλλόντων με διαφορετικές καταστάσεις, ενέργειες και ανταμοιβές(rewards). Αυτά τα περιβάλλοντα είναι διαθέσιμα στο OpenAI.

Το CartPole (σχήμα 1.3α) είναι ένα από τα απλούστερα περιβάλλοντα ενισχυτικής μάθησης, το οποίο περιγράφηκε πρώτα από τους Barto, Sutton και Anderson [4] το 1983. Σε αυτό το περιβάλλον, ένας πόλος συνδέεται με ένα αμαξίδιο που μπορεί να μετακινηθεί κατά μήκος μιας τροχιάς χωρίς τριβή. Τα κύρια χαρακτηριστικά του περιβάλλοντος συνοψίζονται παρακάτω:

1. **Στόχος**: Κρατήστε τον πόλο όρθιο για 200 χρονικά βήματα.
2**. Κατάσταση**: Ένας πίνακας μήκους 4 που αντιπροσωπεύει: [θέση αμαξιδίου, ταχύτητα αμαξιδίου, γωνία πόλων, γωνιακή ταχύτητα πόλου]. Για παράδειγμα, [-0,034, 0,032, -0,031, 0,036].
3. **Ενέργεια**: Ένας ακέραιος αριθμός, είτε 0 για να μετακινήσετε το αμαξίδιο σε μια σταθερή απόσταση προς τα αριστερά, είτε 1 για να μετακινήστε το αμαξίδιο σε μια σταθερή απόσταση προς τα δεξιά.
4. **Ανταμοιβή:** +1 για κάθε φορά που ο πόλος παραμένει όρθιος.
5. **Τερματισμός**: Όταν ο πόλος πέσει πάνω (12 μοίρες μεγαλύτερος από από κάθετο), ή όταν το αμαξίδιο μετακινηθεί έξω από την οθόνη ή όταν το μέγιστο χρονικό βήμα 200 έχει ολοκληρωθεί.

Το Atari Breakout (Εικόνα 1.3b) είναι ένα ρετρό παιχνίδι arcade που αποτελείται από μια μπάλα, μια σανίδα που βρίσκεται στο κάτω μέρος και ελέγχεται από έναν πράκτορα και τούβλα. Ο στόχος είναι να πετύχουμε και να καταστρέψουμε όλα τα τούβλα αναπηδώντας την μπάλα από τη σανίδα. Ένας παίκτης ξεκινά με πέντε ζωές παιχνιδιού και μια ζωή χάνεται κάθε φορά που η μπάλα πέφτει κάτω από την οθόνη.

1. **Στόχος**: Μεγιστοποιήστε το σκορ του παιχνιδιού.
2. **Κατάσταση**: Μια ψηφιακή εικόνα RGB με ανάλυση 160 x 210 pixels - δηλαδή, αυτό που βλέπουμε στην οθόνη του παιχνιδιού.
3. **Δράση**: Ένας ακέραιος από το σύνολο {0, 1, 2, 3} που αντιστοιχεί στον ελεγκτή παιχνιδιών ενέργειες {όχι-δράση, εκτόξευση της μπάλας, κίνηση δεξιά, κίνηση αριστερά}.
4. **Ανταμοιβή:** Η διαφορά βαθμολογίας παιχνιδιού μεταξύ διαδοχικών καταστάσεων.
5. **Τερματισμός:** Όταν χάνονται όλες οι ζωές του παιχνιδιού.

Το BipedalWalker (Σχήμα 1.3γ) είναι ένα πρόβλημα συνεχούς ελέγχου όπου ένας πράκτορας χρησιμοποιεί έναν αισθητήρα lidar του ρομπότ για να αισθανθεί το περιβάλλον του και να περπατήσει προς τα δεξιά χωρίς να πέσει.

1. **Στόχος**: Περπατήστε προς τα δεξιά χωρίς να πέσετε.
2. **Κατάσταση**: Μια συστοιχία μήκους 24 που αντιπροσωπεύει: [γωνία κύτους, γωνιακή ταχύτητα κύτους, ταχύτητα x, ταχύτητα y, γωνία άρθρωσης ισχίου 1, ταχύτητα άρθρωσης ισχίου 1, γόνατο 1 γωνία άρθρωσης, γόνατο 1 ταχύτητα άρθρωσης, πόδι 1 επαφή εδάφους, ισχίο 2 γωνία άρθρωσης, ισχίο 2 ταχύτητα άρθρωσης, γόνατο 2 άρθρωση γωνία, γόνατο 2 ταχύτητα άρθρωσης, πόδι 2 επαφή με το έδαφος,..., 10 μετρήσεις lidar]. Για παράδειγμα [2.745e-03, 1.180e-05, -1.539e-03, -1.600e-02,…, 7.091e-01, 8.859e-01, 1.000e+00, 1.000e+00].
3. **Δράση**: Ένα διάνυσμα τεσσάρων αριθμών κινητής υποδιαστολής στο διάστημα [-1.0, 1.0] το οποίο αντιπροσωπεύει: [ροπή και ταχύτητα ισχίου 1, ροπή και ταχύτητα γόνατος 1, ροπή ισχίου 2 και ταχύτητα, 2 ροπή και ταχύτητα γόνατος]. Για παράδειγμα, [0,097, 0,430, 0,205, 0,089].
4. **Ανταμοιβή(reward)**: Ανταμοιβή για τη μετάβαση προς τα δεξιά, έως και +300, -100 εάν πέσει το ρομπότ. Επιπλέον, υπάρχει μια μικρή αρνητική ανταμοιβή (κόστος κίνησης) σε κάθε χρονικό βήμα, ανάλογα με την απόλυτη ροπή που εφαρμόζεται.
5. **Τερματισμός**: Όταν το σώμα του ρομπότ αγγίζει το έδαφος ή φτάνει στο στόχο στη δεξιά πλευρά ή μετά το μέγιστο χρονικό βήμα 1600.

Αυτά τα περιβάλλοντα δείχνουν μερικές από τις διαφορετικές μορφές που μπορούν να πάρουν τα περιβάλλοντα και οι δράσεις. Στο CartPole και το BipedalWalker, οι καταστάσεις είναι διανύσματα που περιγράφουν ιδιότητες όπως οι θέσεις και οι ταχύτητες. Στο Atari Breakout, η κατάσταση είναι μια εικόνα από το παιχνίδι στην οθόνη. Στο CartPole και στο Atari Breakout, οι ενέργειες είναι μονοί, διακριτοί ακέραιοι, ενώ στο BipedalWalker, μια ενέργεια είναι ένα συνεχές διάνυσμα τεσσάρων αριθμών κινητής

υποδιαστολής. Οι ανταμοιβές είναι πάντα κλιμακωτά μεγέθη, αλλά το εύρος τους ποικίλλει από εφαρμογή σε εφαρμογή.

Έχοντας δει μερικά παραδείγματα, ας περιγράψουμε τώρα επίσημα τις καταστάσεις, τις ενέργειες και τις ανταμοιβές.

s_t $\mathcal{E}$ S είναι η κατάσταση, S είναι ο χώρος κατάστασης. (1.1)
a_t $\mathcal{E}$ A είναι η δράση, A είναι ο χώρος δράσης. (1.2)
$r_t = R(s_t, a_t, s_{t+1})$ είναι η ανταμοιβή, R είναι η συνάρτηση ανταμοιβής. (1.3)

Ο χώρος κατάστασης S είναι το σύνολο όλων των πιθανών καταστάσεων σε ένα περιβάλλον. Ανάλογα με το περιβάλλον, μπορεί να οριστεί με πολλούς διαφορετικούς τρόπους, όπως ακέραιους αριθμούς, πραγματικούς αριθμούς, διανύσματα, πίνακες, δομημένα ή μη δομημένα δεδομένα. Ομοίως, ο χώρος δράσης A είναι το σύνολο όλων των πιθανών ενεργειών που ορίζονται από ένα περιβάλλον. Μπορεί επίσης να λάβει πολλές μορφές, αλλά συνήθως ορίζεται είτε ως βαθμωτό πεδίο είτε ως διάνυσμα. Η συνάρτηση ανταμοιβής $R(s_t, a_t, s_{t+1})$ αποδίδει ένα θετικό, αρνητικό ή μηδενικό βαθμωτό μέγεθος σε κάθε μετάβαση (s_t, a_t, s_{t+1}). Ο χώρος κατάστασης, ο χώρος δράσης και η συνάρτηση ανταμοιβής καθορίζονται από το περιβάλλον. Μαζί, ορίζουν τις (s, a, r) πλειάδες που είναι η βασική ενότητα πληροφοριών που περιγράφει ένα σύστημα ενισχυτικής μάθησης.

1.3 Ενισχυτική Μάθηση ως MDP(Markov Decision Process)

Τώρα, ας σκεφτούμε πώς ένα περιβάλλον μεταβαίνει από τη μία κατάσταση στην επόμενη χρησιμοποιώντας αυτό που είναι γνωστό ως συνάρτηση μετάβασης(transition function). Στην ενισχυτική μάθηση, μια συνάρτηση μετάβασης διατυπώνεται ως μαρκοβιανή διαδικασία λήψης αποφάσεων (MDP), η οποία είναι ένα μαθηματικό πλαίσιο που μοντελοποιεί τη διαδικασία διαδοχικής λήψης αποφάσεων. Για να κατανοήσουμε γιατί οι συναρτήσεις μετάβασης αναπαρίστανται ως μαρκοβιανή διαδικασία λήψης αποφάσεων εξετάζουμε μια γενική διατύπωση που παρουσιάζεται στην εξίσωση 1.4.

$$S_{t+1} \sim P\left(S_{t+1} \middle| (S_0, a_0), (S_1, a_1), \ldots (S_t, a_t)\right) \quad (1.4)$$

Η εξίσωση 1.4 λέει ότι στο χρονικό βήμα t, η επόμενη κατάσταση S_{t+1} λαμβάνεται από μια πιθανότητα διανομής P , η οποία εξαρτάται από ολόκληρο το ιστορικό. Η πιθανότητα ενός περιβάλλοντος να μεταβεί από μια κατάσταση $S_{t,}$ σε S_{t+1} εξαρτάται από όλες τις προηγούμενες καταστάσεις S και ενέργειες A που έχουν συμβεί μέχρι στιγμής σε ένα επεισόδιο. Είναι δύσκολο να μοντελοποιήσουμε μια συνάρτηση μετάβασης σε αυτή τη μορφή, ειδικά αν τα επεισόδια διαρκούν για πολλά χρονικά βήματα. Οποιαδήποτε συνάρτηση μετάβασης που σχεδιάζουμε θα πρέπει να είναι σε θέση να εξηγήσει έναν τεράστιο συνδυασμό επιπτώσεων που συνέβησαν σε οποιοδήποτε σημείο στο παρελθόν. Επιπλέον, αυτή η φόρμουλα καθιστά την συνάρτηση παραγωγής δράσης ενός πράκτορα, δηλαδή την πολιτική του, σημαντικά πιο περίπλοκη. Δεδομένου ότι ολόκληρο το ιστορικό των περιβαλλόντων και των ενεργειών είναι σημαντικό για την κατανόηση του τρόπου με τον οποίο μια ενέργεια μπορεί να αλλάξει τη μελλοντική κατάσταση του περιβάλλοντος, ένας πράκτορας θα πρέπει να λάβει υπόψη του όλες αυτές τις πληροφορίες όταν αποφασίζει πώς να ενεργήσει.

Για να κάνουμε τη μετάβαση στο περιβάλλον πιο πρακτική, τη μετατρέπουμε σε MDP προσθέτοντας την υπόθεση ότι η μετάβαση στην επόμενη κατάσταση S_{t+1} εξαρτάται μόνο από την προηγούμενη κατάσταση $S_{t,}$ και δράση a_t. Αυτό είναι γνωστό ως η ιδιοκτησία Markov. Με αυτήν την υπόθεση, η νέα συνάρτηση μετάβασης γίνεται η ακόλουθη:

$$S_{t+1} \sim P\left(S_{t+1} \middle| S_{t,} a_t\right) \tag{1.5}$$

Η εξίσωση 1.5 λέει ότι η επόμενη κατάσταση S_{t+1} λαμβάνεται από μια κατανομή πιθανότητας $P\left(S_{t+1} \middle| S_{t,} a_t\right)$. Αυτή αποτελεί μια απλούστερη μορφή της αρχικής συνάρτησης μετάβασης. Η ιδιότητα Markov υποδηλώνει ότι η τρέχουσα κατάσταση και η ενέργεια στο χρονικό βήμα t περιέχουν επαρκή πληροφορίες για τον πλήρη προσδιορισμό της πιθανότητας μετάβασης για την επόμενη κατάσταση σε $t + 1$. Παρά την απλότητα αυτής της διατύπωσης, εξακολουθεί να είναι αρκετά βαθυστόχαστη. Πολλές διαδικασίες μπορεί να εκφραστούν σε αυτή τη μορφή, συμπεριλαμβανομένων των παιχνιδιών, του ρομποτικού ελέγχου και του σχεδιασμού. Αυτό συμβαίνει επειδή μια κατάσταση μπορεί να οριστεί ώστε να περιλαμβάνει τις απαραίτητες πληροφορίες που απαιτούνται για τη δημιουργία της συνάρτησης μετάβασης Markov.

Για παράδειγμα, ας σκεφτούμε την ακολουθία Fibonacci που περιγράφεται από τον τύπο $S_{t+1} = S_t + S_{t-1}$, όπου κάθε όρος S_t θεωρείται κατάσταση. Για να κάνουμε τη συνάρτηση Markov, επαναπροσδιορίζουμε την κατάσταση ως S'_t = [S_t, S_{t-1}]. Τώρα η κατάσταση περιέχει επαρκείς πληροφορίες για να υπολογιστεί το επόμενο στοιχείο της ακολουθίας. Η στρατηγική αυτή μπορεί να εφαρμοστεί γενικότερα σε οποιοδήποτε σύστημα στο οποίο ένα πεπερασμένο σύνολο διαδοχικών καταστάσεων Κ περιέχει επαρκείς πληροφορίες για τη μετάβαση στην επόμενη κατάσταση

Είμαστε τώρα σε θέση να παρουσιάσουμε τη διατύπωση του MDP για ένα πρόβλημα ενισχυτικής μάθησης. Ένα MDP ορίζεται από μια 4-πλειάδα $S, A, P(.), R(.)$ Όπου:

- S είναι το σύνολο των περιβαλλόντων.
- Α είναι το σύνολο των ενεργειών.
- $P\left(S_{t+1}\middle|S_{t,}a_t\right)$ η συνάρτηση μετάβασης του περιβάλλοντος .
- R=$(S_{t,}a_t, S_{t+1})$ είναι η συνάρτηση ανταμοιβής του περιβάλλοντος.

Μια σημαντική υπόθεση πίσω από τα προβλήματα ενισχυτικής μάθησης που συζητήθηκαν είναι ότι οι πράκτορες δεν έχουν πρόσβαση στη συνάρτηση μετάβασης $P\left(S_{t+1}\middle|S_{t,}a_t\right)$,ή τη συνάρτηση ανταμοιβής R=$(S_{t,}a_t, S_{t+1})$. Ο μόνος τρόπος με τον οποίο ένας πράκτορας μπορεί να πάρει πληροφορίες σχετικά με αυτές τις συναρτήσεις είναι μέσω των περιβαλλόντων, των δράσεων και των ανταμοιβών που βιώνει πραγματικά στο περιβάλλον, δηλαδή οι πλειάδες $(S_{t,}a_t, r_t)$.

Για να ολοκληρώσουμε τη διαμόρφωση του προβλήματος , θα πρέπει επίσης να διατυπώσουμε το μοτίβο με το οποίο ένας πράκτορας μεγιστοποιεί το στόχο. Πρώτα, ας ορίσουμε το $return\, R(t)$ χρησιμοποιώντας μια τροχιά από ένα επεισόδιο $\tau = (s_0, a_0, r_0)$,….,(s_T, a_T, r_T)*:*

$$R(t) = r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots + \gamma^T r_T = \sum_{t=0}^{T} \gamma^t r_t \qquad \textit{(1.6)}$$

Η εξίσωση 1.6 ορίζει το return(επιστροφή) ως το εκπτωτικό άθροισμα των ανταμοιβών σε μια τροχιά, όπου το $\gamma \ \mathcal{E}$ [0,1] είναι ο εκπτωτικός συντελεστής.

Μετά ο στόχος $J_{(T)}$ είναι απλά η πρόβλεψη των returns για πολλές τροχιές, όπως φαίνεται στην εξίσωση 1.7

$$J_{(T)} = \mathbb{E}_{T\sim\pi}[R(t)] = \mathbb{E}_{\tau}[\sum_{t=0}^{t=T} \gamma^t r_t] \quad \textit{(1.7)}$$

Η επιστροφή $R(t)$ είναι το άθροισμα των εκπτωτικών ανταμοιβών $\gamma^t r_t$ σε όλα τα χρονικά βήματα $t = 0, \dots, T$. Ο στόχος $J_{(T)}$ είναι η επιστροφή που υπολογίζεται κατά μέσο όρο σε πολλά επεισόδια. Η προσδοκία λαμβάνει υπόψιν τη στοχαστικότητα ως προς τις ενέργειες και το περιβάλλον, δηλαδή σε επαναλαμβανόμενες διαδρομές η επιστροφή μπορεί να μην καταλήγει πάντα η ίδια . Μεγιστοποιώντας το στόχο είναι το ίδιο με τη μεγιστοποίηση της επιστροφής.

Ο συντελεστής προεξόφλησης $\gamma\ \mathcal{E}$ [0,1] είναι μια σημαντική μεταβλητή που αλλάζει τον τρόπο που αποτιμώνται οι μελλοντικές ανταμοιβές. Όσο μικρότερο γ, τόσο μικρότερο βάρος δίνεται στις ανταμοιβές σε μελλοντικά χρονικά βήματα, καθιστώντας το «κοντόφθαλμο». Στην ακραία περίπτωση με $\gamma = 0$, ο στόχος λαμβάνει υπόψιν μόνο την αρχική ανταμοιβή r_0, όπως φαίνεται στην εξίσωση 1.8.

$$R(\tau)_{\gamma=0} = \sum_{t=0}^{t=T} \gamma^t r_t = r_0 \quad (1.8)$$

Όσο μεγαλύτερος είναι ο συντελεστής γ , τόσο μεγαλύτερη βαρύτητα δίνεται στις ανταμοιβές στα μελλοντικά χρονικά βήματα: ο στόχος γίνεται πιο «διορατικός». Εάν γ = 1, οι ανταμοιβές από κάθε βήμα χρόνου σταθμίζονται εξίσου, όπως παρουσιάζεται στην εξίσωση 1.9.

$$R(\tau)_{\gamma=1} = \sum_{t=0}^{t=T} \gamma^t r_t = \sum_{t=0}^{t=T} r_t \quad (1.9)$$

Για προβλήματα με άπειρο χρονικό ορίζοντα, πρέπει να ορίσουμε γ < 1 για να αποτρέψουμε τον στόχο από το να καταστεί απεριόριστος. Για προβλήματα πεπερασμένου χρονικού ορίζοντα, το γ είναι μια σημαντική παράμετρος καθώς ένα πρόβλημα ενδέχεται να είναι περισσότερο ή λιγότερο δύσκολο να επιλυθεί, ανάλογα με τον συντελεστή προεξόφλησης που χρησιμοποιούμε.

1.4 Συναρτήσεις Αξίας στην Ενισχυτική Μάθηση

Με την ενισχυτική μάθηση που διατυπώνεται ως MDP, το φυσικό ερώτημα που πρέπει να κάνουμε είναι, τι πρέπει να μάθει ένας πράκτορας; Είδαμε ότι ένας πράκτορας μπορεί να μάθει μια συνάρτηση παραγωγής δράσης γνωστή ως πολιτική. Ωστόσο, υπάρχουν και άλλες ιδιότητες ενός περιβάλλοντος που μπορεί να είναι χρήσιμες σε έναν πράκτορα. Συγκεκριμένα, υπάρχουν τρεις κύριες λειτουργίες που πρέπει να είναι γνωστές στην ενισχυτική μάθηση:

1. Μια πολιτική ,π, η οποία αντιστοιχεί ένα περιβάλλον σε μια ενέργεια,$a \sim \pi(s)$
2. Μια συνάρτηση αξίας (value function), $V^{\pi}(s)$ ή $Q^{\pi}(s,a)$, για τον υπολογισμό της προσδοκώμενης ανταμοιβής $\mathbb{E}_{\tau}[R(t)]$
3. Το μοντέλο περιβάλλοντος , $P(s' \mid s,a)$

Μια πολιτική , π, είναι ο τρόπος με τον οποίο ένας πράκτορας ενεργεί στο περιβάλλον για να μεγιστοποιήσει το στόχο. Δεδομένου του ενισχυτικού βρόχου ελέγχου μάθησης, ένας πράκτορας πρέπει να παράγει μια ενέργεια σε κάθε χρονικό βήμα μετά την παρατήρηση μιας κατάστασης s. Μια πολιτική είναι θεμελιώδης για αυτόν τον βρόχο ελέγχου, δεδομένου ότι δημιουργεί τις ενέργειες για να το κάνει να δουλέψει.

Μια πολιτική μπορεί να είναι στοχαστική. Δηλαδή, μπορεί να παράγει πιθανολογικά διαφορετικές ενέργειες για την ίδια κατάσταση. Μπορούμε να το γράψουμε αυτό ως $\pi(a,s)$ για να δηλώσουμε την πιθανότητα μιας ενέργειας a δεδομένου ενός περιβάλλοντος s. Ένα δείγμα ενέργειας από μια πολιτική γράφεται ως $a \sim \pi(s)$.

Οι συναρτήσεις αξίας (value functions) παρέχουν πληροφορίες σχετικά με το στόχο. Βοηθούν έναν πράκτορα να κατανοήσει πόσο καλές είναι οι καταστάσεις και οι διαθέσιμες δράσεις όσον αφορά την αναμενόμενη μελλοντική επιστροφή. Έρχονται σε δύο μορφές - τις συναρτήσεις , $V^{\pi}(s)$ και $Q^{\pi}(s,a)$.

$$V^{\pi}(s) = \mathbb{E}_{s_{0=s,\tau\sim\pi}}[\sum_{t=0}^{t=T} \gamma^t r_t] \tag{1.10}$$

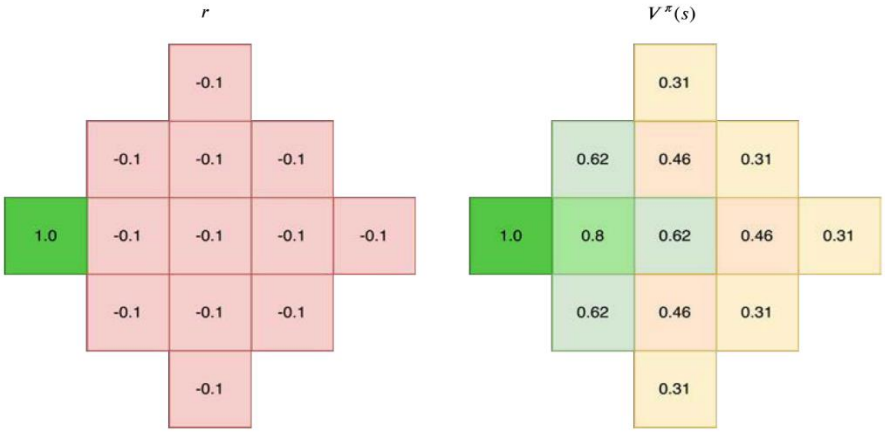
$$Q^{\pi}(s,a) = \mathbb{E}_{s_{0=s,a_0=a,\tau\sim\pi}}[\sum_{t=0}^{t=T} \gamma^t r_t] \tag{1.11}$$

Η συνάρτηση αξίας V^{π} που εμφανίζεται στην εξίσωση 1.10 αξιολογεί πόσο καλή ή κακή είναι μια κατάσταση. Η V^{π} μετρά την αναμενόμενη επιστροφή από την κατάσταση s, υποθέτοντας ότι ο πράκτορας συνεχίζει να ενεργεί σύμφωνα με την τρέχουσα πολιτική της π. Η επιστροφή $R(t) = \sum_{t=0}^{T} \gamma^t r_t$ μετράται από την τρέχουσα κατάσταση s στο τέλος ενός επεισοδίου. Είναι ένα μέτρο προσανατολισμένο στο μέλλον, αφού όλες οι ανταμοιβές που λαμβάνονται πριν το περιβάλλον s αγνοούνται.

Για να κατανοήσουμε καλύτερα τη συνάρτηση αξίας V^{π}, ας εξετάσουμε ένα απλό παράδειγμα. Το σχήμα 1.4 απεικονίζει ένα περιβάλλον κόσμου-πλέγματος (grid-world) στο οποίο ένας πράκτορας μπορεί να μετακινηθεί από κελί σε κελί κάθετα ή οριζόντια. Κάθε κελί είναι μια κατάσταση με ανταμοιβή συσχετισμού, όπως φαίνεται στα αριστερά του σχήματος. Το περιβάλλον τερματίζεται όταν ο πράκτορας φτάσει στην κατάσταση στόχου με ανταμοιβή $r = +1$.

Στα δεξιά, δείχνουμε την αξία $V^{\pi}(s)$ που υπολογίζεται για κάθε κατάσταση από τις ανταμοιβές χρησιμοποιώντας την εξίσωση 1.10, με $\gamma = 0.9$. Η συνάρτηση αξίας V^{π} εξαρτάται πάντα από μια συγκεκριμένη πολιτική π. Σε αυτό το παράδειγμα, επιλέξαμε μια πολιτική π ,η οποία ακολουθεί πάντα τη συντομότερη διαδρομή προς τη κατάσταση στόχου. Αν είχαμε επιλέξει μια άλλη πολιτική – για παράδειγμα, μια πολιτική που κινείται πάντα σωστά ,τότε οι τιμές θα ήταν διαφορετικές.

Εδώ μπορούμε να δούμε την προοδευτική ιδιότητα της συνάρτησης αξίας και την ικανότητά της να βοηθάει έναν πράκτορα να διακρίνει τη διαφορά μεταξύ καταστάσεων που δίνουν την ίδια ανταμοιβή. Όσο πιο κοντά είναι ένας πράκτορας στην κατάσταση στόχου, τόσο υψηλότερη είναι η τιμή.



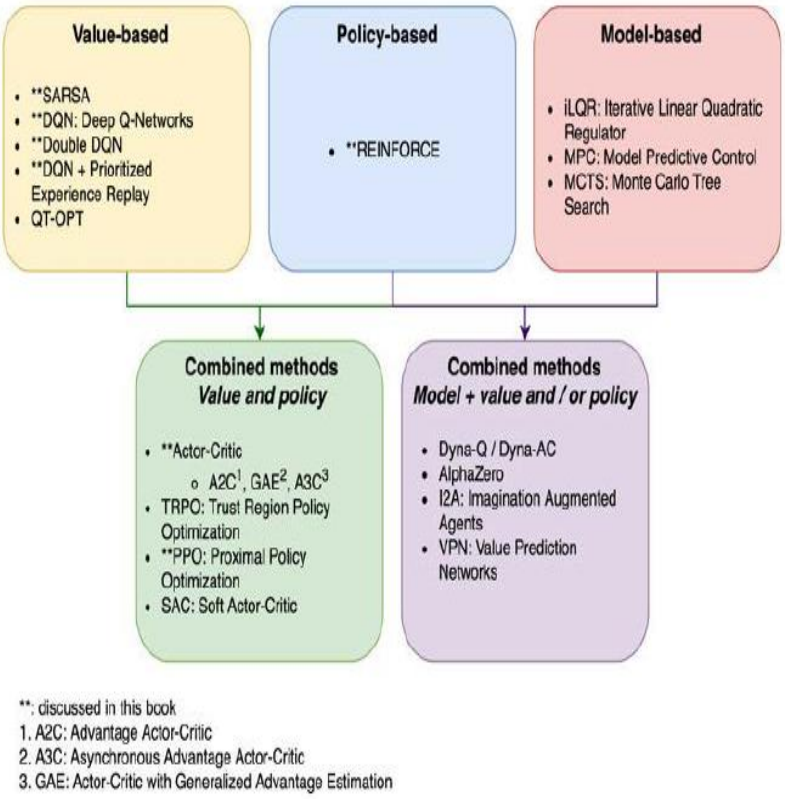
Σχήμα 1.4 Rewards r και $V^{\pi}(s)$ για κάθε κατάσταση s σε ένα απλό περιβάλλον κόσμου πλέγματος. Η αξία μιας κατάστασης υπολογίζεται από τις ανταμοιβές χρησιμοποιώντας την ξίσωση 1,10 με γ = 0,9 χρησιμοποιώντας μια πολιτική π που ακολουθεί πάντα τη συντομότερη διαδρομή προς την κατάσταση στόχου με r=1.

Η συνάρτηση Q-αξίας(Q-value function) Q^{π} που εμφανίζεται στην εξίσωση 1.11 αξιολογεί πόσο καλό ή κακό είναι ένα ζεύγος κατάστασης-δράσης . Η Q^{π} μετρά την αναμενόμενη επιστροφή από την ανάληψη δράσης σε μια κατάσταση υποθέτοντας ότι ο πράκτορας συνεχίζει να ενεργεί σύμφωνα με την τρέχουσα πολιτική του, π. Με τον ίδιο τρόπο όπως στην V^{π}, η επιστροφή μετράται από την τρέχουσα κατάσταση s έως το τέλος ενός επεισοδίου. Αυτό είναι επίσης ένα μελλοντοστραφές μέτρο, δεδομένου ότι όλες οι ανταμοιβές που λαμβάνονται πριν την κατάσταση s αγνοούνται.

Η συνάρτηση μετάβασης(transition function) $P(s' \mid s, a)$ παρέχει πληροφορίες σχετικά με το περιβάλλον. Εάν ένας πράκτορας μαθαίνει αυτή τη συνάρτηση, είναι σε θέση να προβλέψει την επόμενη κατάσταση s' ,όπου το περιβάλλον θα μεταβεί μετά την ανάληψη δράσης a σε μια κατάσταση s. Εφαρμόζοντας τη συνάρτηση μετάβασης που μάθαμε, ένας πράκτορας μπορεί να «φανταστεί» τις συνέπειες των πράξεων του χωρίς να αγγίξει πραγματικά το περιβάλλον. Στη συνέχεια, μπορεί να χρησιμοποιήσει αυτές τις πληροφορίες για να σχεδιάσει καλές ενέργειες.

1.5 Αλγόριθμοι Deep Learning για Ενισχυτική Μάθηση

Στην ενισχυτική μάθηση, ένας πράκτορας μαθαίνει συναρτήσεις που τον βοηθάνε να δρά και να μεγιστοποιεί το στόχο. Αυτή η εργασία ασχολείται με τη βαθιά ενισχυτική μάθηση (deep reinforcement learning). Αυτό σημαίνει ότι χρησιμοποιούμε βαθιά νευρωνικά δίκτυα ως μέθοδο προσέγγισης συναρτήσεων. Στην Ενότητα 1.4, είδαμε τις τρεις κύριες μαθησιακές συναρτήσεις στην ενισχυτική μάθηση. Αντίστοιχα, υπάρχουν τρεις μεγάλες οικογένειες βαθιάς ενισχυτικής μάθησης αλγόριθμων— μέθοδοι βασισμένες σε πολιτικές(policy based), βασισμένες σε αξίες(value based) και βασισμένες σε μοντέλα(model based), οι οποίες μαθαίνουν πολιτικές, συναρτήσεις αξίας και μοντέλα, αντίστοιχα. Υπάρχουν επίσης συνδυασμένες μέθοδοι στις οποίες οι πράκτορες εκπαιδεύονται σε περισσότερες από μία από αυτές τις συναρτήσεις— για παράδειγμα, μια συνάρτηση πολιτικής και αξίας ή μια συνάρτηση αξίας και ένα μοντέλο. Το σχήμα 1.5 παρέχει μια επισκόπηση των σημαντικών αλγόριθμων βαθιάς ενισχυτικής μάθησης σε κάθε οικογένεια και πώς σχετίζονται.



Σχήμα 1.5 Οικογένειες deep learning αλγόριθμων ενισχυτικής μάθησης.

1.5.1 Αλγόριθμοι βασισμένοι στη πολιτική (value based)

Οι αλγόριθμοι σε αυτή την οικογένεια μαθαίνουν μια πολιτική π. Οι καλές πολιτικές πρέπει να παράγουν δράσεις οι οποίες παράγουν τροχιές τ που μεγιστοποιούν τον στόχο ενός πράκτορα, $J(\tau) = \mathbb{E}_{\tau\sim\pi} \sum_{t=0}^{T} \gamma^t r_t$. Αυτή η προσέγγιση είναι αρκετά διαισθητική - εάν ένας πράκτορας πρέπει να ενεργήσει σε ένα περιβάλλον, είναι λογικό να μάθει μια πολιτική. Το τι συνιστά μια καλή ενέργεια σε μια δεδομένη στιγμή εξαρτάται από την κατάσταση, οπότε μια συνάρτηση πολιτικής π παίρνει μια κατάσταση s ως είσοδο για να παράγει μια ενέργεια $a\sim\pi(s)$. Αυτό σημαίνει ότι ένας πράκτορας μπορεί να λάβει καλές αποφάσεις σε διαφορετικά πλαίσια. Η ΕΝΊΣΧΥΣΗ(REINFORCE) [5], που παρουσιάζεται στο κεφάλαιο 3, είναι ο πιο γνωστός αλγόριθμος που βασίζεται σε πολιτικές και αποτελεί τη βάση της ανάπτυξης πολλών μεταγενέστερων αλγορίθμων.

Ένα σημαντικό πλεονέκτημα των αλγορίθμων που βασίζονται σε πολιτικές είναι ότι αποτελούν μια πολύ γενική κατηγορία μεθόδων βελτιστοποίησης. Μπορούν να εφαρμοστούν σε προβλήματα με οποιοδήποτε είδος ενεργειών, δηλαδή διακριτό, συνεχές ή μείγμα (πολλαπλές δράσεις). Επίσης, βελτιστοποιούν άμεσα τον στόχο $J(\tau)$ για το

πράγμα το οποίο ένας πράκτορας ενδιαφέρεται περισσότερο .Επιπλέον, αυτή η κατηγορία μεθόδων είναι εγγυημένο ότι θα συγκλίνει σε μια τοπικά βέλτιστη πολιτική, όπως αποδεικνύεται από τους Sutton et al. με το Θεώρημα πολιτικής κλίσης (Policy Gradient Theorem)[6]. Ένα μειονέκτημα αυτών των μεθόδων είναι ότι έχουν υψηλή διακύμανση και είναι δειγματοληπτικά αναποτελεσματικές.

1.5.2 Αλγόριθμοι βασισμένοι στη συνάρτηση αξίας (value function)

Ένας πράκτορας μαθαίνει είτε τη συνάρτηση $V^{\pi}(s)$ είτε την $Q^{\pi}(s,a)$.Χρησιμοποιεί τη συνάρτηση αξίας για την αξιολόγηση (s,a) ζευγαριών και δημιουργεί μια πολιτική. Για παράδειγμα, η πολιτική ενός πράκτορα θα μπορούσε να επιλέγει πάντα τη δράση a στην κατάσταση s με την υψηλότερη εκτιμώμενη $Q^{\pi}(s,a)$. Η μάθηση $Q^{\pi}(s,a)$ είναι πολύ πιο συχνή από τη $V^{\pi}(s)$ για προσεγγίσεις που βασίζονται σε καθαρές τιμές, επειδή είναι ευκολότερο να μετατραπεί σε πολιτική. Αυτό συμβαίνει επειδή η $Q^{\pi}(s,a)$ περιέχει πληροφορίες σχετικά με αντιστοιχισμένες καταστάσεις και ενέργειες ενώ η $V^{\pi}(s)$ περιέχει μόνο πληροφορίες σχετικά με τις καταστάσεις.

Ο SARSA(State–Action–Reward–State–Action) [7] , είναι ένας από τους παλαιότερους αλγόριθμους ενισχυτικής μάθησης . Παρά την απλότητά του, ο SARSA ενσωματώνει πολλές από τις βασικές ιδέες που βασίζονται σε μεθόδους αξίας(value-based), οπότε είναι ένας καλός αλγόριθμος για να μελετήσουμε πρώτα σε αυτήν την οικογένεια. Ωστόσο, δεν χρησιμοποιείται συνήθως σήμερα λόγω της υψηλής διακύμανσης και της αναποτελεσματικότητας του δείγματος κατά τη διάρκεια της εκπαίδευσης. Βαθιά δίκτυα Q (Deep Q Networks -DQN) [8] και οι μεταγενέστεροί του αλγόριθμοι, όπως το double DQN [9] και DQN με επανάληψη εμπειρίας προτεραιότητας (Per Instance-PER) [10], είναι πολύ πιο δημοφιλείς και αποτελεσματικοί αλγόριθμοι.

Οι αλγόριθμοι που βασίζονται σε συναρτήσεις αξίας(value functions) είναι συνήθως πιο αποδοτικοί ως προς το δείγμα από ό,τι οι αλγόριθμοι που βασίζονται σε πολιτικές. Αυτό συμβαίνει επειδή έχουν χαμηλότερη διακύμανση και αξιοποιούν καλύτερα τα δεδομένα που συλλέγονται από το περιβάλλον. Ωστόσο, δεν υπάρχουν εγγυήσεις ότι αυτοί οι αλγόριθμοι θα συγκλίνουν σε ένα βέλτιστο. Στην τυποποιημένη σύνθεσή τους, εφαρμόζονται επίσης μόνο σε περιβάλλοντα με διακριτά πεδία δράσης. Αυτό ήταν ιστορικά ένας σημαντικός περιορισμός, αλλά με τις πιο πρόσφατες εξελίξεις, όπως το QT-OPT(Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation)[11],μπορούν να εφαρμοστούν αποτελεσματικά σε περιβάλλοντα με πεδία συνεχούς δράσης.

1.5.3 Αλγόριθμοι βασισμένοι σε μοντέλα(model-based algorithms)

Οι αλγόριθμοι σε αυτή την οικογένεια είτε μαθαίνουν ένα μοντέλο της δυναμικής μετάβασης ενός περιβάλλοντος είτε κάνουν χρήση ενός γνωστού μοντέλου δυναμικής. Μόλις ένας πράκτορας έχει ένα μοντέλο του περιβάλλοντος, $P(s' \mid s, a)$, μπορεί να «φανταστεί» τι θα συμβεί στο μέλλον προβλέποντας την τροχιά για λίγα χρονικά βήματα. Εάν το περιβάλλον βρίσκεται σε κατάσταση s, ένας πράκτορας μπορεί να εκτιμήσει πώς η κατάσταση θα αλλάξει αν κάνει μια ακολουθία ενεργειών $a_1, a_2, \dots, a_n$ εφαρμόζοντας επανειλημμένα $P(s' \mid s, a)$, όλα αυτά χωρίς να παράγουν πραγματικά μια ενέργεια για την αλλαγή του περιβάλλοντος. Έτσι η προβλεπόμενη τροχιά αντιλαμβάνεται από τον πράκτορα χρησιμοποιώντας ένα μοντέλο. Ένας πράκτορας μπορεί να ολοκληρώσει πολλές διαφορετικές προβλέψεις τροχιάς με διαφορετικές ακολουθίες ενεργειών και, στη συνέχεια, εξετάζει αυτές τις επιλογές για να αποφασίσει για την καλύτερη ενέργεια a που πρέπει να κάνει πραγματικά.

Οι καθαρά βασισμένες σε μοντέλα προσεγγίσεις εφαρμόζονται συνήθως σε παιχνίδια με στόχο κατάσταση(target state), όπως νίκη ή ήττα σε μια παρτίδα σκάκι ή εργασίες πλοήγησης με κατάσταση στόχου (goal-state) s^*. Αυτό συμβαίνει επειδή οι συναρτήσεις μετάβασής τους δεν διαμορφώνουν ανταμοιβές. Για να χρησιμοποιηθούν για την σχεδίαση δράσεων, ορισμένες πληροφορίες σχετικά με τον στόχο ενός πράκτορα πρέπει να κωδικοποιηθούν από τις ίδιες τις καταστάσεις.

Η αναζήτηση δέντρων του Monte Carlo (Monte Carlo Tree Search-MCTS) είναι μια γνωστή μέθοδος που βασίζεται σε μοντέλο και μπορεί να εφαρμόζεται σε ζητήματα που αφορούν ντετερμινιστικούς διακριτούς χώρους κατάστασης με γνωστές συναρτήσεις μετάβασης. Πολλά επιτραπέζια παιχνίδια όπως το σκάκι και το Go εμπίπτουν σε αυτήν την κατηγορία και μέχρι πρόσφατα, το MCTS τροφοδότησε πολλούς υπολογιστές Go προγραμμάτων [12]. Δεν χρησιμοποιεί καθόλου μηχανική μάθηση, αλλά τυχαία δείγματα ακολουθιών ενεργειών, γνωστές ως ανάπτυξη του Monte-Carlo, για να εξερευνήσει τις καταστάσεις ενός παιχνιδιού και να εκτιμήσει την αξία τους [12]. Υπήρξαν αρκετές βελτιώσεις σε αυτόν τον αλγόριθμο, αλλά αυτή είναι η βασική ιδέα.

Άλλες μέθοδοι, όπως οι επαναληπτικοί γραμμικοί τετραγωνικοί ρυθμιστές (iterative Linear Quadratic Control-iLQR) [13] ή το μοντέλο προγνωστικού ελέγχου (Model Predictive Control-MPC), περιλαμβάνουν την μάθηση της δυναμικής μετάβασης, συχνά κάτω από αρκετά περιοριστικές υποθέσεις. Για να μάθει τη δυναμική, ένας πράκτορας θα

πρέπει να ενεργήσει σε ένα περιβάλλον για τη συγκέντρωση παραδειγμάτων πραγματικών μεταβάσεων (s, a, r, s').

Οι αλγόριθμοι που βασίζονται σε μοντέλα είναι πολύ ελκυστικοί επειδή ένα τέλειο μοντέλο προσκομίζει σε έναν πράκτορα προνοητικότητα, ούτως ώστε να μπορεί να παίξει σενάρια και να κατανοήσει τις συνέπειες των πράξεων του χωρίς να χρειάζεται να δράσει πραγματικά σε ένα περιβάλλον. Αυτό μπορεί να είναι ένα σημαντικό πλεονέκτημα στις καταστάσεις, όπου είναι πολύ χρονοβόρα ή δαπανηρή η συλλογή εμπειριών από το περιβάλλον— για παράδειγμα, στη ρομποτική. Σε σύγκριση με την μέθοδο βασισμένη σε πολιτική(policy based) ή την μέθοδο βασισμένη στη συνάρτηση αξίας (value based), αυτοί οι αλγόριθμοι τείνουν επίσης να απαιτούν πολύ λιγότερα δείγματα δεδομένων για να μάθουν καλά πολιτικές, δεδομένου ότι η ύπαρξη ενός μοντέλου επιτρέπει σε έναν πράκτορα να συμπληρώνει τις πραγματικές του εμπειρίες με φανταστικές.

Ωστόσο, για τα περισσότερα προβλήματα, τα μοντέλα είναι δύσκολο να βρεθούν. Πολλά περιβάλλοντα είναι στοχαστικά και η δυναμική μετάβασής τους δεν είναι γνωστή. Σε αυτές τις περιπτώσεις, το μοντέλο πρέπει να είναι πολυμαθής. Η προσέγγιση αυτή βρίσκεται ακόμη σε πρώιμο στάδιο ανάπτυξης και αντιμετωπίζει μια σειρά προκλήσεων. Πρώτον, ένα περιβάλλον με μεγάλο χώρο καταστάσεων και χώρο δράσης μπορεί να είναι πολύ δύσκολο να μοντελοποιηθεί- κάτι τέτοιο μπορεί ακόμη και να είναι δυσεπίλυτο, ειδικά εάν οι μεταβάσεις είναι εξαιρετικά πολύπλοκες. Δεύτερον, τα μοντέλα είναι χρήσιμα μόνο όταν μπορούν να προβλέψουν με ακρίβεια τις μεταβάσεις ενός περιβάλλοντος πολλά βήματα στο μέλλον. Ανάλογα με την ακρίβεια του μοντέλου, τα σφάλματα πρόβλεψης ενδέχεται να επιδεινωθούν για κάθε βήμα τη φορά και να αναπτυχθούν γρήγορα και να κάνουν το μοντέλο αναξιόπιστο.

Η έλλειψη καλών μοντέλων αποτελεί επί του παρόντος σημαντικό περιορισμό για τη λειτουργικότητα των προσεγγίσεων βασισμένες σε μοντέλα. Ωστόσο, οι μέθοδοι που βασίζονται σε μοντέλα μπορεί να είναι πολύ ισχυρές. Όταν λειτουργούν, είναι συχνά μια ή δύο τάξεις μεγέθους πιο αποδοτικές από τις μεθόδους χωρίς μοντέλα.

Η διάκριση μεταξύ μοντέλου(model based) και χωρίς μοντέλο(model free) χρησιμοποιείται επίσης για την ταξινόμηση των αλγόριθμων ενισχυτικής μάθησης. Ένας αλγόριθμος που βασίζεται σε μοντέλο είναι απλά οποιοσδήποτε αλγόριθμος που χρησιμοποιεί τη δυναμική μετάβασης ενός περιβάλλοντος, είτε την έχει μάθει είτε του είναι γνωστή εκ των προτέρων. Οι αλγόριθμοι χωρίς μοντέλα είναι εκείνοι που δεν χρησιμοποιούν ρητά τη δυναμική μετάβασης περιβάλλοντος.

1.5.4 Συνδυασμένες Μέθοδοι (combined methods)

Αυτοί οι αλγόριθμοι μαθαίνουν δύο ή περισσότερες από τις κύριες λειτουργίες της ενισχυτικής μάθησης. Δεδομένων των πλεονεκτημάτων και των αδυναμιών καθεμιάς από τις τρεις μεθόδους που συζητήθηκαν μέχρι στιγμής, είναι φυσικό να προσπαθήσουμε να τα συνδυάσουμε για να πάρουμε το καλύτερο από το καθένα. Μια ευρέως χρησιμοποιούμενη ομάδα των αλγορίθμων μαθαίνει μια πολιτική και μια συνάρτηση αξίας. Αυτοί εύστοχα ονομάζονται Πράκτορας-Κριτής(Actor-Critic) αλγόριθμοι επειδή η πολιτική ενεργεί και η συνάρτηση αξίας αξιολογεί τις ενέργειες. Η βασική ιδέα είναι ότι κατά τη διάρκεια της εκπαίδευσης, μια μαθημένη συνάρτηση αξίας (learned value function) μπορεί να παρέχει ένα πιο ενημερωτικό σήμα ανατροφοδότησης σε μια πολιτική από την ακολουθία των ανταμοιβών που είναι διαθέσιμες από το περιβάλλον. Η πολιτική μαθαίνει χρησιμοποιώντας τις πληροφορίες που παρέχονται από την μαθημένη συνάρτηση αξίας. Στη συνέχεια, η πολιτική χρησιμοποιείται για τη δημιουργία δράσεων, όπως στις μεθόδους βασισμένες στη πολιτική.

Οι αλγόριθμοι Actor-Critic είναι ένας ενεργός τομέας έρευνας και υπήρξαν πολύ ενδιαφέρουσες εξελίξεις τα τελευταία χρόνια — Βελτιστοποίηση πολιτικής περιοχής εμπιστοσύνης(Trust Region Policy Optimization) (TRPO) [14], Βελτιστοποίηση εγγύς πολιτικής (Proximal Policy Optimization) (PPO) [15], Βαθιά ντετερμινιστική πολιτική Διαβαθμίσεων(Deterministic Policy Gradients) (DDPG) [16], και Soft Actor-Critic (SAC) [17], για να αναφέρουμε μερικούς. Από αυτούς, ο PPO είναι σήμερα ο πιο ευρέως χρησιμοποιούμενος.

Οι αλγόριθμοι μπορούν επίσης να χρησιμοποιούν ένα μοντέλο της δυναμικής μετάβασης περιβάλλοντος σε συνδυασμό με συνάρτηση τιμής ή/και πολιτική. Το 2016, ερευνητές από το DeepMind ανέπτυξαν το AlphaGo, το οποίο συνδύαζε το MCTS με την μάθηση V^{π} και μια πολιτική π για να κυριαρχήσει το παιχνίδι του Go. Το Dyna- Q [18] είναι ένας άλλος γνωστός αλγόριθμος που επαναληπτικά μαθαίνει ένα μοντέλο χρησιμοποιώντας πραγματικά δεδομένα από το περιβάλλον και, στη συνέχεια, χρησιμοποιεί τα φανταστικά δεδομένα που δημιουργούνται από ένα μαθητευόμενο μοντέλο για να μάθει τη συνάρτηση Q.

Τα παραδείγματα που δίνονται σε αυτή την ενότητα είναι μόνο μερικά από των πολλών αλγορίθμων βαθιάς ενισχυτικής μάθησης. Δεν πρόκειται σε καμία περίπτωση για εξαντλητικό κατάλογο· αντίθετα, η πρόθεσή ήταν να δοθεί μια επισκόπηση των κύριων ιδεών στη βαθιά ενισχυτική μάθηση και τους τρόπους με τους οποίους οι πολιτικές, οι

συναρτήσεις αξίας και τα μοντέλα μπορούν να χρησιμοποιηθούν και να συνδυαστούν. Η βαθιά ενισχυτική μάθηση είναι ένας πολύ ενεργός τομέας έρευνας και φαίνεται ότι κάθε λίγους μήνες υπάρχουν συναρπαστικές νέες εξελίξεις στον τομέα.

1.5.5 On policy και Off policy αλγόριθμοι

Μια τελευταία σημαντική διάκριση μεταξύ των αλγορίθμων βαθιάς ενισχυτικής μάθησης είναι εάν είναι on policy ή off policy. Αυτό επηρεάζει τον τρόπο με τον οποίο οι επαναλήψεις εκπαίδευσης χρησιμοποιούν τα δεδομένα.

Ένας αλγόριθμος είναι on-policy εάν μαθαίνει πάνω στην πολιτική - δηλαδή, η εκπαίδευση μπορεί να χρησιμοποιήσει μόνο δεδομένα που δημιουργούνται από την τρέχουσα πολιτική π. Αυτό σημαίνει ότι καθώς η εκπαίδευση επαναλαμβάνεται μέσω εκδόσεων των πολιτικών,π_1, π_2, π_3, ...,κάθε επανάληψη εκπαίδευσης χρησιμοποιεί μόνο την τρέχουσα πολιτική εκείνη τη στιγμή για τη δημιουργία δεδομένων μάθησης. Ως αποτέλεσμα, όλα τα δεδομένα πρέπει να απορρίπτονται μετά την μάθηση, δεδομένου ότι καθίστανται άχρηστα. Αυτό καθιστά τις μεθόδους πολιτικής αναποτελεσματικές — απαιτούν περισσότερα δεδομένα εκπαίδευσης.

ΚΕΦΑΛΑΙΟ 2ο

Νευρωνικά Δίκτυα και Βαθιά Μάθηση

2.1 Εισαγωγή

Τα νευρωνικά δίκτυα εμπνεύστηκαν από το βραβευμένο με Νόμπελ έργο του Hubel και Wiesel στον πρωτογενή οπτικό φλοιό των γατών(primary visual cortex of cats) [19]. Τα επιδραστικά τους πειράματα έδειξαν ότι τα νευρωνικά δίκτυα οργανώθηκαν σε ιεραρχικά στρώματα κυττάρων για την επεξεργασία οπτικών ερεθισμάτων. Το πρώτο μαθηματικό μοντέλο του νευρωνικού δικτύου , που ονομάστηκε Neocognitron το 1980 [20], είχε πολλά από τα χαρακτηριστικά των σημερινών βαθιών συνελικτικών νευρωνικών δικτύων (ή DCNN), συμπεριλαμβανομένης μιας πολυεπίπεδης δομής, συνέλιξης, μέγιστης συγκέντρωσης και μη γραμμικούς δυναμικούς κόμβους. Η πρόσφατη επιτυχία των βαθιών συνελικτικών νευρωνικών δικτύων (Deep Convolutional Neural Networks-DCNNs) στην υπολογιστική όραση έχει επιτευχθεί χάρη σε δύο κρίσιμες συνιστώσες: (i) τη συνεχή αύξηση της υπολογιστικής ισχύος και (ii) τα εξαιρετικά μεγάλα σύνολα δεδομένων με ετικέτες που εκμεταλλεύονται τη δύναμη μιας βαθιάς πολυεπίπεδης αρχιτεκτονικής. Πράγματι, αν και η θεωρητική ενασχόληση με τα νευρωνικά δίκταυα έχει ιστορικό σχεδόν τεσσάρων δεκαετιών, η ανάλυση του συνόλου δεδομένων ImageNet το 2012 [21] αποτέλεσε ορόσημο για τα νευρωνικά δίκτυα και τη βαθιά μάθηση [22]. Πριν από αυτό το σύνολο δεδομένων, υπήρχε ένας αριθμός διαθέσιμων συνόλων δεδομένων με περίπου δεκάδες χιλιάδες επισημασμένες εικόνες. Η ImageNet παρείχε πάνω από 15 εκατομμύρια επισημασμένες εικόνες υψηλής ανάλυσης με περισσότερες από 22.000 κατηγορίες. Τα βαθιά συνελικτικά νευρωνικά δίκτυα, τα οποία είναι μόνο μία δυνητική κατηγορία νευρωνικών δικτύων, έχουν έκτοτε μεταμορφώσει το πεδίο της υπολογιστικής όρασης κυριαρχώντας στις μετρήσεις απόδοσης σχεδόν σε κάθε ουσιαστική εργασία υπολογιστικής όρασης που προορίζεται για ταξινόμηση και αναγνώριση.

Τα τελευταία χρόνια υπήρξαν ορισμένες κρίσιμες καινοτομίες ,οι οποίες καθιέρωσαν τα πολυστρωματικά δίκτυα τροφοδοσίας ως μια κατηγορία καθολικών προσεγγιστών [23]. Τα τελευταία πέντε χρόνια σημειώθηκε τεράστια πρόοδος στις αρχιτεκτονικές νευρωνικών δικτύων, πολλές σχεδιασμένες και προσαρμοσμένες για συγκεκριμένους τομείς εφαρμογών. Καινοτομίες προέρχονται από αλγοριθμικές τροποποιήσεις που έχουν οδηγήσει σε σημαντική αύξηση της απόδοσης σε διάφορους τομείς. Αυτές οι και-

νοτομίες περιλαμβάνουν προκατάρτιση, dropout, ενότητες έναρξης, αύξηση δεδομένων με εικονικά παραδείγματα, κανονικοποίηση παρτίδων ή/και υπολειπόμενη μάθηση (Βλ. Αναφορά [24] για λεπτομερή έκθεση των NN). Αυτός είναι μόνο ένας μερικός κατάλογος πιθανών αλγοριθμικών καινοτομιών, υπογραμμίζοντας έτσι τον συνεχή και ταχύ ρυθμό προόδου στον τομέα. Αξιοσημείωτα νευρωνικά δίκτυα δεν συμπεριλήφθηκαν καν ως ένας από τους 10 κορυφαίους αλγόριθμους εξόρυξης δεδομένων το 2008 [25]. Αλλά μια δεκαετία αργότερα, ο αναμφισβήτητος και αυξανόμενος κατάλογος επιτυχιών του σχετικά με τα σύνολα δεδομένων Challenge, το καθιστούν ίσως το πιο σημαντικό εργαλείο εξόρυξης δεδομένων για την αναδυόμενη γενιά επιστημόνων και μηχανικών.

Όπως φαίνεται ήδη, όλη η μηχανική μάθηση έχει ως πυρήνα προβλήματα και μεθόδους βελτιστοποίησης. Τα νευρωνικά δίκτυα βελτιστοποιούνται ειδικά σε σχέση με μια σύνθεση συνάρτηση

$$argmin_{A_j}(f_M(A_M, \dots, f_2(A_2, f_1(A_1, x)) \dots) + \lambda g(A_j)) \qquad (2.1)$$

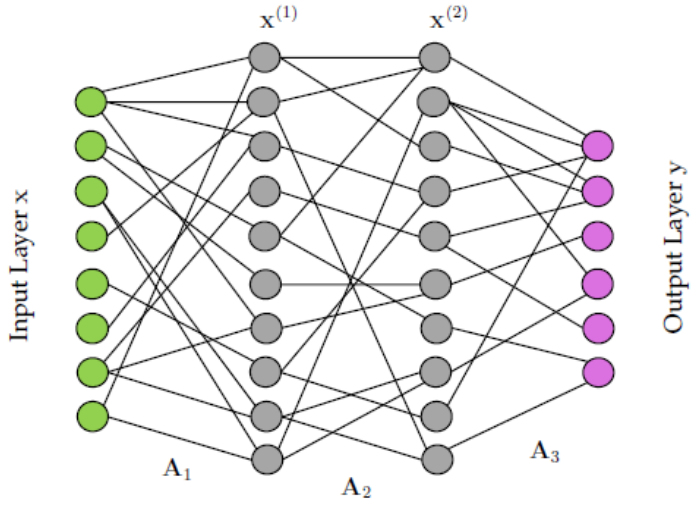
,η οποία συχνά επιλύεται χρησιμοποιώντας αλγόριθμους στοχαστικής καθοδικής κλίσης (stochastic gradient descent) και διάδοσης προς τα πίσω(back propagation). Κάθε πίνακας A_k υποδηλώνει τα βάρη που συνδέουν το νευρωνικό δίκτυο από το k έως το $(k+1)$ στρώμα. Πρόκειται για ένα μαζικά απροσδιόριστο σύστημα το οποίο κανονικοποιείται από το $g(A_j)$. Η σύνθεση και η κανονικοποίηση είναι κρίσιμες για τη δημιουργία εκφραστικών αναπαραστάσεων των δεδομένων και αποφυγή υπέρπροσαρμογής, αντίστοιχα. Αυτό το γενικό πλαίσιο βελτιστοποίησης βρίσκεται στο επίκεντρο των αλγόριθμων βαθιάς μάθησης και η λύση του θα εξεταστεί σε αυτό το κεφάλαιο.

2.2 Νευρωνικά δίκτυα

Η γενική αρχιτεκτονική ενός πολυστρωματικού νευρωνικού δικτύου φαίνεται στο σχήμα 2.1. Για την ταξινόμηση καθηκόντων, ο στόχος του νευρωνικού δικτύου είναι να αντιστοιχίσει ένα σύνολο δεδομένων εισόδου σε μια ταξινόμηση. Συγκεκριμένα, εκπαιδεύουμε το νευρωνικό δίκτυο να αντιστοιχεί με ακρίβεια τα δεδομένα x_j στη σωστή ετικέτα τους y_j. Όπως φαίνεται στο σχήμα 2.1, ο χώρος εισόδου έχει τη διάσταση των ανεπεξέργαστων δεδομένων x_j Ε $\mathbb{R}^n$. Το επίπεδο εξόδου έχει τη διάσταση της σχεδιασμένης ταξινόμησης χώρου. Η δόμηση του επιπέδου εξόδου θα συζητηθεί περαιτέρω στη συνέχεια.[26]

Αμέσως, μπορεί κανείς να δει ότι υπάρχει ένας μεγάλος αριθμός ερωτήσεων σχεδιασμού σχετικά με τα νευρωνικά δίκτυα. Πόσα στρώματα πρέπει να χρησιμοποιηθούν; Ποια πρέπει να είναι η διάσταση των στρωμάτων; Πώς πρέπει να σχεδιαστεί το επίπεδο εξόδου; Σε περίπτωση μίας χρήσης όλα-προς-όλα ή αραιές συνδέσεις μεταξύ των επιπέδων; Πώς πρέπει να γίνει η αντιστοίχιση μεταξύ των επιπέδων: γραμμική αντιστοίχιση ή μη γραμμική αντιστοίχιση; Όπως και οι επιλογές συντονισμού σε Support Vector Machines -SVM και δέντρα ταξινόμησης, έτσι και τα νευρωνικά δίκτυα έχουν σημαντικό αριθμό επιλογών σχεδίασης που μπορούν να ρυθμιστούν για τη βελτίωση της απόδοσης.

Αρχικά, εξετάζουμε την αντιστοίχιση μεταξύ των στρωμάτων του Σχήματος 2.1. Δηλώνουμε τα διάφορα επίπεδα μεταξύ εισόδου και εξόδου ως $x^{(k)}$,όπου k είναι ο αριθμός στρώματος.



Σχήμα 2.1: Απεικόνιση μιας αρχιτεκτονικής νευρωνικού δικτύου που χαρτογραφεί ένα επίπεδο εισόδου x σε ένα επίπεδο εξόδου y. Τα μεσαία (κρυφά) επίπεδα συμβολίζονται $x^{(j)}$ όπου το j καθορίζει τη διαδοχική διάταξή τους. Οι πίνακες A_j περιέχουν τους συντελεστές που αντιστοιχίζουν κάθε μεταβλητή από το ένα επίπεδο στο άλλο. Αν και η διάσταση του επιπέδου εισόδου x Ε $\mathbb{R}^n$ είναι γνωστή, υπάρχει μεγάλη ευελιξία στην επιλογή της διάστασης των εσωτερικών στρωμάτων καθώς και τον τρόπο δομής του στρώματος εξόδου. Ο αριθμός των επιπέδων και ο τρόπος αντιστοίχισης μεταξύ των επιπέδων επιλέγεται επίσης από το χρήστη. Αυτή η ευέλικτη αρχιτεκτονική δίνει μεγάλη ελευθερία στην διαμόρφωση ενός καλού ταξινομητή.

Για γραμμική αντιστοίχιση μεταξύ επιπέδων, ισχύουν οι ακόλουθες σχέσεις:

$$x^{(1)} = A_1 x \quad (2.2.1)$$
$$x^{(2)} = A_2 x \quad (2.2.2)$$
$$y = A_3 x \quad (2.2.3)$$

Αυτό σχηματίζει μια συνθετική δομή έτσι ώστε η αντιστοίχιση μεταξύ εισόδου και εξόδου να μπορεί να αναπαρασταθεί ως:

$$y = A_3 A_2 A_1 x \quad (2.3)$$

Αυτή η βασική αρχιτεκτονική μπορεί να κλιμακωθεί σε επίπεδα M έτσι ώστε μια γενική αναπαράσταση μεταξύ των δεδομένων εισόδου και του επιπέδου εξόδου για ένα γραμμικό NN να δίνεται από τον τύπο:

$$y = A_M A_{M-1} \ldots\ldots A_2 A_1 x \quad (2.4)$$

Αυτό είναι γενικά ένα εξαιρετικά αόριστο σύστημα που απαιτεί ορισμένους περιορισμούς στη λύση για την επιλογή μιας μοναδικής λύσης. Ένας περιορισμός είναι άμεσα προφανής: Η αντιστοίχιση πρέπει να παράγει M διακριτούς πίνακες που δίνουν την καλύτερη αντιστοίχιση. Πρέπει να σημειωθεί ότι οι γραμμικές αντιστοιχίσεις, ακόμη και με μια συνθετική δομή, μπορεί να παράγουν μόνο ένα περιορισμένο φάσμα λειτουργικών αποκρίσεων λόγω των περιορισμών της γραμμικότητας.

Οι μη γραμμικές αντιστοιχίσεις είναι επίσης δυνατές, και γενικά χρησιμοποιούνται, στην διαμόρφωση των νευρωνικών δικτύων. Πράγματι, οι μη γραμμικές συναρτήσεις ενεργοποίησης επιτρέπουν ένα πλουσιότερο σύνολο λειτουργικών αποκρίσεων από τις γραμμικές αντίστοιχες. Σε αυτήν την περίπτωση, οι συνδέσεις μεταξύ των στρωμάτων δίνονται από τον τύπο:

$$x^{(1)} = f_1(A_1, x) \quad (2.5.1)$$
$$x^{(2)} = f_2(A_2, x^{(1)}) \quad (2.5.2)$$
$$y = f_3(A_3, x^{(2)}) \quad (2.5.3)$$

Να σημειωθεί ότι έχουμε χρησιμοποιήσει διαφορετικές μη γραμμικές συναρτήσεις $f_j(\cdot)$ μεταξύ των επιπέδων. Συχνά χρησιμοποιείται μία μόνο συνάρτηση. Ωστόσο, δεν υπάρχει κανένας περιορισμός ότι αυτό είναι απαραίτητο. Όσον αφορά την αντιστοίχιση των δεδομένων μεταξύ εισόδου και εξόδου σε επίπεδα M, προκύπτουν τα εξής:

$$y = f_M(A_M, \dots, f_2\big(A_2, f_1(A_1, x)\big) \dots) \tag{2.6}$$

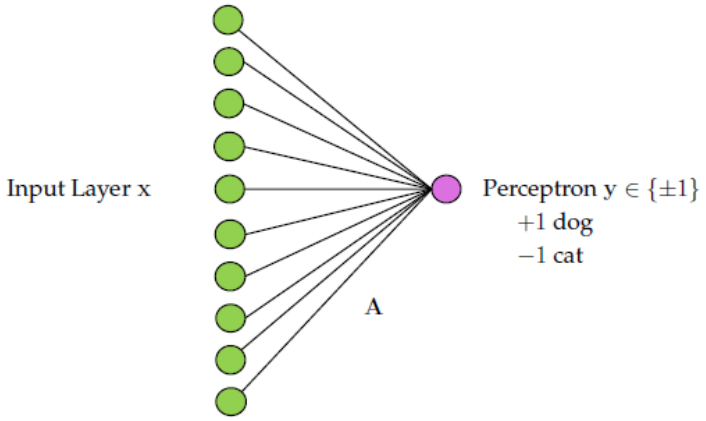
που μπορεί να συγκριθεί με τη (2.1) για τη γενική βελτιστοποίηση που δομεί το νευρωνικό δίκτυο. Ως ένα εξαιρετικά ανεπαρκώς καθορισμένο σύστημα, οι περιορισμοί θα πρέπει να επιβάλλονται για την εξαγωγή ενός επιθυμητού τύπου λύσεως, όπως στο σημείο (2.1). Για εφαρμογές μεγάλων δεδομένων, όπως το ImageNET και εργασίες υπολογιστικής όρασης, η βελτιστοποίηση που σχετίζεται με αυτό το συνθετικό πλαίσιο είναι δαπανηρή δεδομένου του αριθμού μεταβλητών που πρέπει να προσδιοριστούν. Ωστόσο, για δίκτυα μέτριου μεγέθους, μπορεί να εκτελεστεί σε χώρους εργασίας και φορητούς υπολογιστές. Σύγχρονοι στοχαστικοί αλγόριθμοι καθοδικής κλίσης (stochastic gradient descent) και διάδοσης προς τα πίσω(back propagation) επιτρέπουν αυτή τη βελτιστοποίηση, και τα δύο καλύπτονται σε επόμενες ενότητες.

2.2.1 Δίκτυο ενός επιπέδου

Για να αποκτήσουμε μια εικόνα για το πώς μπορεί να δομηθεί ένα νευρωνικό δίκτυο, θα εξετάσουμε ένα δίκτυο ενός επιπέδου, που είναι βελτιστοποιημένο για τη δημιουργία ενός ταξινομητή μεταξύ σκύλων και γατών. Το σχήμα 2.2 δείχνει την δομή μας. Για να γίνει αυτό όσο το δυνατόν πιο απλό, εξετάζουμε την απλή έξοδο νευρωνικού δικτύου:

$$y = \{dog, cat\} = \{+1, -1\} \tag{2.7}$$

το οποίο επισημαίνει κάθε διάνυσμα δεδομένων με έξοδο $y\ E\ \{\pm 1\}$. Σε αυτή την περίπτωση η έξοδος του στρώματος είναι ένας μόνο κόμβος. Όπως και σε προηγούμενους αλγόριθμους εποπτευόμενης μάθησης, ο στόχος είναι να προσδιορίσουμε μια αντιστοίχιση έτσι ώστε κάθε διάνυσμα δεδομένων x_j να επισημαίνεται σωστά από το y_j .



Σχήμα 2.2: Μονοστρωματικό δίκτυο για δυαδική ταξινόμηση μεταξύ σκύλων και γατών. Το στρώμα εξόδου για αυτή την περίπτωση είναι ένα perceptron με $y\ E\ \{\pm 1\}$. Γραμμική αντιστοίχιση μεταξύ του χώρου εικόνας εισόδου και του επιπέδου εξόδου μπορεί να κατασκευαστεί για δεδομένα εκπαίδευσης με την επίλυση $A = YX^{*}$. Αυτό δίνει μια ελάχιστη τετραγωνική παλινδρόμηση για τη μήτρα A που αντιστοιχεί τις εικόνες στην επισήμανση του χώρου.

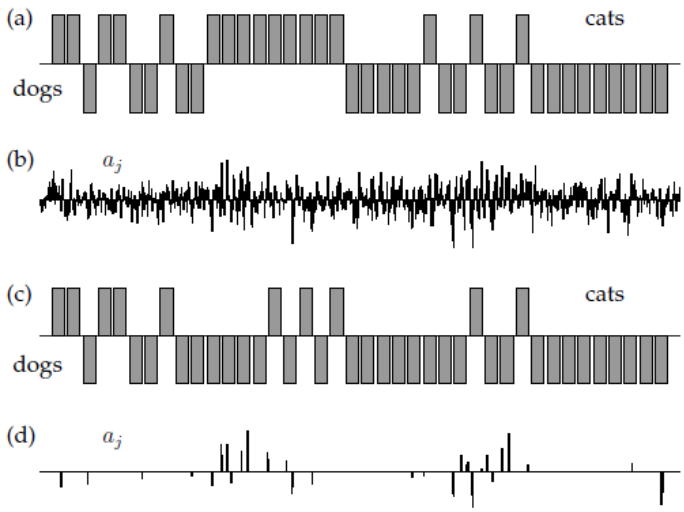
Η ευκολότερη αντιστοίχιση χαρτογράφηση είναι μια γραμμική αντιστοίχιση μεταξύ των εικόνων εισόδου x_j Ε $\mathbb{R}^n$ και του επιπέδου εξόδου. Αυτό δίνει ένα γραμμικό σύστημα $AX = Y$ της μορφής:

$$AX = Y \rightarrow [a_1\ a_2\ \dots a_n] = \begin{bmatrix} | & | & | \\ X_1 & X_2 & X_p \\ | & | & | \end{bmatrix} = [+1\ \ +1 \dots -1\ -1] \qquad (2.8)$$

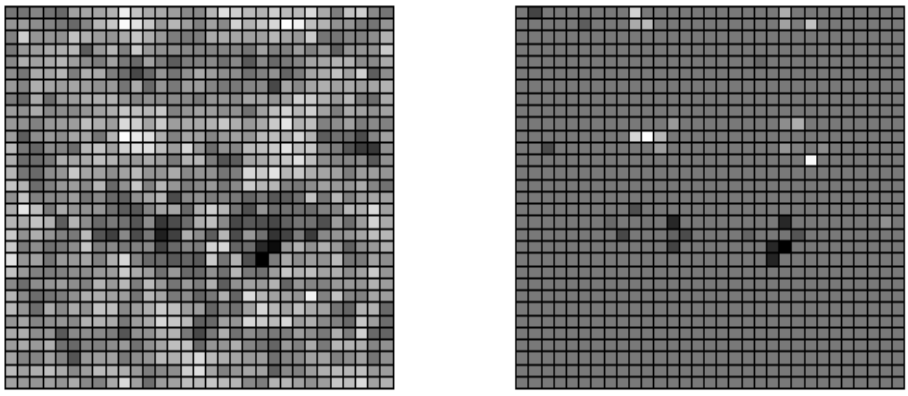
όπου κάθε στήλη του πίνακα X είναι μια εικόνα σκύλου ή γάτας και οι στήλες του Y είναι οι αντίστοιχες ετικέτες του. Δεδομένου ότι το επίπεδο εξόδου είναι ένας μόνο κόμβος, τόσο το A όσο και το Y καταλήγουν σε διανύσματα. Σε αυτή την περίπτωση, ο στόχος μας είναι να προσδιορίσουμε τον πίνακα (διάνυσμα) A με στοιχεία a_j . Η απλούστερη λύση είναι να πάρουμε το ψευδο-αντίστροφο του πίνακα δεδομένων X:

$$A = YX^{*} \qquad (2.9)$$

Έτσι, ένα μόνο στρώμα εξόδου μας επιτρέπει να δομήσουμε ένα νευρωνικό δίκτυο χρησιμοποιώντας την τοποθέτηση ελαχίστων τετραγώνων. Φυσικά, θα μπορούσαμε επίσης να λύσουμε αυτό το γραμμικό σύστημα με διάφορους άλλους τρόπους, μεταξύ άλλων με μεθόδους προώθησης της αραιότητας.



Σχήμα 2.3: Ταξινόμηση των παρακρατηθέντων δεδομένων που δοκιμάστηκαν σε εκπαιδευμένο, μονοστρωματικό δίκτυο με γραμμική αντιστοίχιση μεταξύ των εισόδων (χώρος εικονοστοιχείων) και μίας μόνο εξόδου. (a) και (c) είναι το γράφημα της βαθμολογίας επιπέδου εξόδου $y\ E\ \{\pm 1\}$ που επιτεύχθηκε για τα παρακρατημένα δεδομένα χρησιμοποιώντας ένα ψευδο-αντίστροφο για την εκπαίδευση και το LASSO για την εκπαίδευση αντίστοιχα. Τα αποτελέσματα δείχνουν και στις δύο περιπτώσεις ότι οι σκύλοι ταξινομούνται συχνότερα λανθασμένα από τις γάτες .(b) και (d) εμφανίζουν τους συντελεστές της μήτρας A για το ψευδο-αντίστροφο και το LASSO αντίστοιχα. Σημειώστε ότι το LASSO έχει μόνο έναν μικρό αριθμός μη μηδενικών στοιχείων, υποδηλώνοντας έτσι ότι το νευρωνικό δίκτυο είναι εξαιρετικά αραιό.



Σχήμα 2.4: Σταθμίσεις του πίνακα A αναδιαμορφωμένο σε πίνακες 32x32. Ο αριστερός πίνακας δείχνει τον πίνακα A που υπολογίζεται με παλινδρόμηση ελαχίστων τετραγώνων (το ψευδο-αντίστροφο) ενώ ο δεξιός πίνακας δείχνει τον πίνακα A που υπολογίζεται από το LASSO. Αμφότεροι οι πίνακες παρέχουν παρόμοιες βαθμολογίες ταξινόμησης σε παρακρατημένα δεδομένα. Επιπλέον παρέχουν ερμηνευτικότητα με την έννοια ότι τα αποτελέσματα από το ψευδο-αντίστροφο παρουσιάζουν πολλά από τα χαρακτηριστικά των σκύλων και των γατών, ενώ το LASSO δείχνει ότι η μέτρηση κοντά στα μάτια και τα αυτιά μόνο μπορεί να δώσει τα χαρακτηριστικά που απαιτούνται για τη διάκριση μεταξύ σκύλων και γατών.

Τα σχήματα 2.3 και 2.4 δείχνουν τα αποτελέσματα αυτού του γραμμικού μονοστρωματικού νευρωνικού δικτύου με μονό στρώμα εξόδου κόμβου. Συγκεκριμένα, οι τέσσερις σειρές του σχήματος 2.3 δείχνουν το επίπεδο εξόδου των δεδομένων δοκιμής που δεν έχουν παρακρατηθεί τόσο για την ψευδο-αντίστροφη όσο και για τη μέθοδο LASSO μαζί με ένα γράφημα των σταθμίσεων 32x32 (1024 εικονοστοιχεία) του πίνακα A. Να σημειωθεί ότι όλα τα στοιχεία μήτρας είναι μη μηδενικά στη ψευδο-αντίστροφη λύση, ενώ το LASSO επισημαίνει έναν μικρό αριθμό pixel που μπορούν να ταξινομήσουν τις εικόνες καθώς και τη χρήση όλων των εικονοστοιχείων. Στο σχήμα 2.4 παρουσιάζεται η μήτρα A για τις δύο στρατηγικές λύσεων αναδιαμορφωμένες σε εικόνες 32x32. Να σημειωθεί ότι για το ψευδο-αντίστροφο, οι σταθμίσεις των στοιχείων μήτρας A δείχνουν πολλά χαρακτηριστικά του προσώπου της γάτας και του σκύλου. Για τη μέθοδο LASSO, απαιτούνται μόνο λίγα pixel που είναι ομαδοποιημένα κοντά στα μάτια και τα αυτιά. Έτσι, για αυτό το δίκτυο ενός επιπέδου, ερμηνεύσιμα αποτελέσματα επιτυγχάνονται εξετάζοντας τα βάρη που παράγονται στον πίνακα A.

2.2.2 Πολυεπίπεδα-Πολυστρωματικά δίκτυα και συναρτήσεις ενεργοποίησης(activation functions)

Η προηγούμενη ενότητα καθόρισε αυτό που είναι ίσως το απλούστερο δυνατό νευρωνικό δίκτυο. Αυτό ήταν γραμμικό, είχε ένα μόνο στρώμα και έναν νευρώνα επιπέδου εξόδου. Οι δυνατότητες γενικεύσεων είναι ατελείωτες, αλλά θα επικεντρωθούμε σε δύο απλές επεκτάσεις του νευρωνικού δικτύου σε αυτήν την ενότητα. Η πρώτη επέκταση αφορά την υπόθεση της γραμμικότητας στην οποία υποθέσαμε ότι υπάρχει ένας γραμμικός μετασχηματισμός από το χώρο της εικόνας στο επίπεδο εξόδου: $AX = Y$ στην (2.8). Επισημαίνουμε εδώ τους κοινούς μη γραμμικούς μετασχηματισμούς από το χώρο εισόδου σε αυτόν της εξόδου που αναπαρίστανται ως:

$$y = f(A, x) \tag{2.10}$$

όπου $f(\cdot)$είναι μια καθορισμένη συνάρτηση ενεργοποίησης-activation function (transfer function) για τη χαρτογράφησή μας.

Η γραμμική αντιστοίχιση που χρησιμοποιήθηκε προηγουμένως, αν και απλή, δεν προσφέρει την ευελιξία και απόδοση που προσφέρουν άλλες αντιστοιχίσεις. Κάποια τυπικές συναρτήσεις ενεργοποίησης δίνονται από τον τύπο :

$f(x) = x$ -linear (2.11.a)

$f(x) = \begin{cases} 0 \; x \leq 0 \\ 1 \; x > 0 \end{cases}$ -binary step (2.11.b)

$f(x) = \frac{1}{1+\exp(-x)}$ -logistic(soft step) (2.11.c)

$f(x) = \tanh(x)$ -TanH (2.11.d)

$f(x) = \begin{cases} 0 \; x \leq 0 \\ x \; x > 0 \end{cases}$ -rectified linear unit (ReLU) (2.11.e)

Υπάρχουν και άλλες δυνατότητες, αλλά αυτές είναι ίσως οι πιο συχνά εξεταζόμενες στην πράξη και θα εξυπηρετήσουν τους σκοπούς μας. Είναι σημαντικό ότι η επιλεγμένη συνάρτηση $f(x)$ θα διαφοροποιηθεί προκειμένου να χρησιμοποιηθεί σε αλγόριθμους καθοδικής κλίσης για βελτιστοποίηση. Κάθε μία από τις παραπάνω συναρτήσεις είναι είτε διαφορίσιμη είτε τμηματικά διαφορίσιμη. Ίσως η πιο συχνά χρησιμοποιούμενη συνάρτηση ενεργοποίησης αυτή τη στιγμή είναι η ReLU, την οποία δηλώνουμε $f(x) = Relu(x)$.

Με μη γραμμική συνάρτηση ενεργοποίησης $f(x)$ ή εάν υπάρχουν περισσότερα από ένα επίπεδα και, στη συνέχεια, τυπικές ρουτίνες γραμμικής βελτιστοποίησης, όπως το ψευδο-αντίστροφο και το LASSO δεν μπορεί πλέον να χρησιμοποιηθούν. Αν και αυτό μπορεί να μην φαίνεται αμέσως σημαντικό, υπενθυμίζουμε ότι βελτιστοποιούμε σε έναν χώρο υψηλών διαστάσεων όπου κάθε καταχώρηση του πίνακα A πρέπει να βρεθεί μέσω βελτιστοποίησης. Ακόμη και για μέτρια ή μικρά προβλήματα μπορεί να είναι υπολογιστικά δαπανηρή η επίλυσή τους χωρίς τη χρήση εξειδικευμένων μεθόδων βελτιστοποίησης. Ευτυχώς, τα δύο κυρίαρχα στοιχεία βελτιστοποιήσης για την εκπαίδευση των νευρωνικών δικτύων, την καθοδική στοχαστική κλίση (stochastic gradient descent) και την διάδοση προς τα πίσω (backpropagation), περιλαμβάνονται στις συναρτήσεις που αφορούν νευρωνικά δίκτυα στο MATLAB. Επειδή αυτές οι μέθοδοι είναι κρίσιμες, και οι δύο εξετάζονται λεπτομερώς στις επόμενες δύο ενότητες του παρόντος κεφαλαίου.

Πολλαπλά επίπεδα μπορούν επίσης να θεωρηθούν όπως φαίνεται στα σημεία (2.4) και (2.5). Σε αυτήν την περίπτωση, η βελτιστοποίηση πρέπει να προσδιορίζει ταυτόχρονα πολλαπλές μήτρες σύνδεσης $A_1, A_2, \dots, A_M$, σε αντίθεση με τη γραμμική περίπτωση όπου μόνο ένας πίνακας προσδιορίζεται $\bar{A} = A_M \dots . A_2 A_M$. Η δομή πολλαπλών στρωμάτων αυξάνει σημαντικά το μέγεθος του προβλήματος βελτιστοποίησης αφού πρέπει να προσδιοριστεί κάθε πίνακας δεδομένων της μήτρας M. Ακόμη και για μια δομή ενός επιπέδου, μια ρουτίνα βελτιστοποίησης όπως το Fminsearch θα αμφισβητηθεί σοβαρά όταν εξεταστεί μια μη γραμμική συνάρτηση μεταφοράς και πρέπει να

μετακινηθεί σε έναν αλγόριθμο βασισμένο στη μέθοδο καθοδικής κλίσης(gradient descent-based).

2.3 Ο αλγόριθμος backpropagation

Όπως φάνηκε για τα νευρωνικά δίκτυα των δύο τελευταίων ενοτήτων, απαιτούνται δεδομένα εκπαίδευσης για τον προσδιορισμό των βαρών του δικτύου. Συγκεκριμένα, τα βάρη του δικτύου είναι αποφασισμένα έτσι ώστε να ταξινομούν καλύτερα τις εικόνες σκύλου έναντι γάτας. Στο δίκτυο ενός επιπέδου, αυτό έγινε χρησιμοποιώντας τόσο την παλινδρόμηση ελαχίστων τετραγώνων όσο και το LASSO. Αυτό δείχνει ότι στον πυρήνα του, απαιτείται μια ρουτίνα βελτιστοποίησης και μια συνάρτηση στόχου για τον προσδιορισμό των βαρών. Η συνάρτηση στόχου θα πρέπει να ελαχιστοποιεί τις λανθασμένα ταξινομημένες εικόνες. Η βελτιστοποίηση, ωστόσο, μπορεί να τροποποιηθεί επιβάλλοντας μια κανονικοποίηση ή περιορισμούς, όπως η τιμωρία του $l1$ στο LASSO.

Στην πράξη, η συνάρτηση στόχου(objective function) που επιλέγεται για βελτιστοποίηση δεν είναι η αληθινή συνάρτηση στόχου που επιθυμούμε, αλλά μάλλον ένα υποκατάστατο της. Οι πληρεξούσιοι (proxies) επιλέγονται σε μεγάλο βαθμό λόγω της ικανότητας διαφοροποίησης της συνάρτησης στόχου σε ένα υπολογιστικά εύχρηστο τρόπο. Υπάρχουν επίσης πολλές διαφορετικές συναρτήσεις στόχου για διαφορετικά θέματα εργασίας. Αντ' αυτού, συχνά θεωρείται μια κατάλληλα επιλεγμένη συνάρτηση απώλειας, ώστε να προσεγγίζει τον πραγματικό στόχο. Τελικά, η υπολογιστική ελκτικότητα είναι κρίσιμη για την κατάρτιση των νευρωνικών δικτύων.

Ο αλγόριθμος backpropagation (backprop) εκμεταλλεύεται τη συνθετική φύση των νευρωνικών δικτύων προκειμένου να πλαισιώσει ένα πρόβλημα βελτιστοποίησης για τον προσδιορισμό των βαρών του δικτύου. Συγκεκριμένα, παράγει μια φόρμουλα σύμφωνα με την τυπική βελτιστοποίηση καθοδικής κλίσης . Το backprop βασίζεται σε μια απλή μαθηματική αρχή: ο κανόνας της αλυσίδας για τη διαφοροποίηση. Επιπλέον, μπορεί να αποδειχθεί ότι ο υπολογιστικός χρόνος που απαιτείται για την αξιολόγηση της κλίσης είναι εντός πέντε φορές του χρόνου που απαιτείται για τον υπολογισμό της πραγματικής συνάρτησης [21]. Αυτό είναι γνωστό ως θεώρημα Baur-Strassen. Το σχήμα δίνει το απλούστερο παράδειγμα backprop και πώς πρέπει να πραγματοποιηθεί η καθοδική κλίση.

Σχήμα 2.5: Σύνοψη της απόδοσης σφάλματος μέσω πινάκων σύγχυσης της αρχιτεκτονικής νευρωνικού δικτύου για την εκπαίδευση, της επικύρωσης και σετ δοκιμών.

Η σχέση εισόδου-εξόδου για αυτόν τον μεμονωμένο κόμβο, δίκτυο ενός κρυφού επιπέδου, δίνεται από τον τύπο:

$$y = g(z, b) = g(f(x, a), b) \qquad (2.12)$$

Έτσι δοσμένης μιας συνάρτηση $f(\cdot)$ και $g(\cdot)$ με σταθερές στάθμισης a και b, το σφάλμα εξόδου που παράγεται από το δίκτυο μπορεί να υπολογιστεί με βάση την βασική θεώρηση, ως

$$E = \frac{1}{2}(y_0 - y)^2 \qquad (2.13)$$

όπου y_0 είναι η σωστή έξοδος και y είναι η προσέγγιση του νευρωνικού δικτύου στην έξοδο. Ο στόχος είναι να βρεθεί το a και το b για να ελαχιστοποιηθεί το σφάλμα. Η ελαχιστοποίηση απαιτεί:

$$\frac{\partial E}{\partial a} = -(y_0 - y)\frac{\partial y}{\partial z}\frac{\partial z}{\partial a} = 0 \qquad (2.14)$$

Μια σημαντική παρατήρηση είναι ότι η συνθετική φύση του δικτύου μαζί με τον κανόνα αλυσίδας (chain rule) ωθούν την βελτιστοποίηση στη παραγωγή σφάλματος BackPropagate μέσω του δικτύου. Ειδικότερα, οι όροι $\frac{\partial y}{\partial z}, \frac{\partial z}{\partial a}$ δείχνουν πώς συμβαίνει αυτό το backprop. Δεδομένων των συναρτήσεων $f(\cdot)$ και $g(\cdot)$, ο κανόνας της αλυσίδας μπορεί να υπολογιστεί ρητά. Το backprop έχει ως αποτέλεσμα έναν επαναληπτικό κανόνα ενημέρωσης gradient descent:

$$a_{k+1} = a_k + \delta\frac{\partial E}{\partial a_k} \qquad (2.15a)$$

$$b_{k+1} = b_k + \delta\frac{\partial E}{\partial b_k} \qquad (2.15b)$$

όπου δ είναι ο λεγόμενος ρυθμός μάθησης και $\frac{\partial E}{\partial a}$ μαζί με $\frac{\partial E}{\partial b}$ μπορεί να είναι υπολογισμένα ρητά με χρήση της (2.14). Ο αλγόριθμος επανάληψης εκτελείται για να συγκλίνει. Όπως συμβαίνει με όλες τις επαναληπτικές βελτιστοποιήσεις, μια καλή αρχική εικασία είναι κρίσιμη για να επιτευχθεί μια καλή λύση σε εύλογο χρονικό διάστημα.

Το backprop προχωρά ως εξής: (i) Καθορίζεται ένα νευρωνικό δίκτυο μαζί με ένα τυποποιημένο σετ εκπαίδευσης. ii) Οι αρχικές σταθμίσεις του δικτύου ορίζονται σε τυχαίες τιμές. Είναι σημαντικό ότι δεν πρέπει να αρχικοποιήθουν τα βάρη στο μηδέν, παρόμοια με αυτά που μπορεί να γίνεται σε άλλους αλγόριθμους μηχανικής μάθησης. Εάν τα βάρη αρχικοποιηθούν στο μηδέν, μετά από κάθε ενημέρωση, τα εξερχόμενα βάρη κάθε νευρώνα θα είναι πανομοιότυπα, επειδή οι κλίσεις θα είναι ίδιες. Επιπλέον, τα νευρωνικά δίκτυα συχνά κολλάνε σε τοπικό επίπεδο optima όπου η κλίση (gradient) είναι μηδέν αλλά που δεν είναι καθολικά ελάχιστα, οπότε η τυχαία αρχικοποίηση βάρους επιτρέπει σε κάποιον να έχει την ευκαιρία να το παρακάμψει ξεκινώντας σε πολλές διαφορετικές τυχαίες τιμές. (iii) Τα δεδομένα εκπαίδευσης κινούνται μέσω του δικτύου για την παραγωγή μιας εξόδου y, της οποίας η ιδανική έξοδος είναι y_0. Οι παράγωγοι ανάλογα με το κάθε βάρος του δικτύου υπολογίζονται στη συνέχεια χρησιμοποιώντας τους backprop τύπους (2.14). (iv) Για ένα δεδομένο ρυθμό μάθησης δ, τα βάρη του δικτύου ενημερώνονται όπως στο (2.15). (v) Επιστρέφουμε στο βήμα (iii) και

συνεχίζουμε την επανάληψη μέχρι να φτάσουμε το μέγιστο αριθμό επαναλήψεων ή όταν επιτυγχάνεται σύγκλιση.

Ως απλό παράδειγμα, εξετάζεται η συνάρτηση γραμμικής ενεργοποίησης:

$$f(\xi, \alpha) = g(\xi, \alpha) = \alpha\xi \qquad (2.16)$$

Σε αυτήν την περίπτωση έχουμε στο σχήμα 2.6:

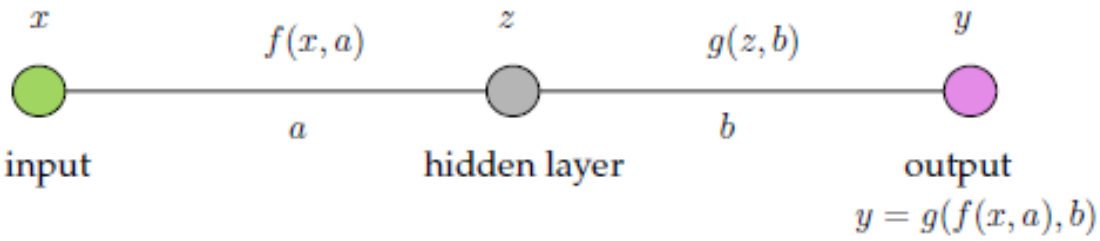
$$z = ax \qquad (2.17a)$$
$$y = bz \qquad (2.17b)$$

Τώρα μπορούμε να υπολογίσουμε σαφώς τις κλίσεις όπως (2.14). Αυτό δίνει:

$$\frac{\partial E}{\partial a} == -(y_0 - y)\frac{\partial y}{\partial z}\frac{\partial z}{\partial a} == -(y_0 - y)b\, x \qquad (2.18a)$$

$$\frac{\partial E}{\partial b} == -(y_0 - y)\frac{\partial y}{\partial b} == -(y_0 - y)z = -(y_0 - y)ax \qquad (2.18b)$$

Έτσι με τις τρέχουσες τιμές των a και b, μαζί με το ζεύγος εισόδου-εξόδου x και y και στόχο y_0 , κάθε παράγωγος μπορεί να υπολογιστεί. Αυτό παρέχει τις απαιτούμενες πληροφορίες για την εκτέλεση της ενημέρωσης (2.15).



Σχήμα 2.6: Απεικόνιση του αλγορίθμου backpropagation για ένα δίκτυο ενός κόμβου, ενός κρυφού επιπέδου. Ο συνθετικός χαρακτήρας του δικτύου δίνει την σχέση εισόδου-εξόδου $y = g(z, b) = g(f(x, a), b)$. Ελαχιστοποιώντας το σφάλμα μεταξύ της εξόδου y και της επιθυμητής εξόδου y_0, η σύνθεση μαζί με τον κανόνας αλυσίδας παράγει έναν ρητό τύπο (2.14) για την ενημέρωση των τιμών των βαρών. Να σημειωθεί ότι ο κανόνας αλυσίδας προωθεί το σφάλμα σε όλη τη διαδρομή του δικτύου. Έτσι, ελαχιστοποιώντας την έξοδο, ο κανόνας αλυσίδας δρα στη σύνθεση για την παραγωγή ενός γινομένου παράγωγων όρων που προχωρούν προς τα πίσω μέσω του δικτύου.

Το backprop για ένα βαθύτερο δίκτυο ακολουθεί παρόμοιο τρόπο. Εξετάζουμε ένα δίκτυο με Μ κρυφά στρώματα από z_1 έως z_m με το βάρος πρώτης σύνδεσης a μεταξύ x και z_1 . Η γενίκευση του σχήματος 2.6 και της σχέσης (2.14) δίνεται από τον τύπο:

$$\frac{\partial E}{\partial a} == -(y_0 - y)\frac{\partial y}{\partial z_m}\frac{\partial z_m}{\partial z_{m-1}} \dots \frac{\partial z_2}{\partial z_1}\frac{\partial z_1}{\partial z_a} \tag{2.19}$$

Η ακολουθία των παραγώγων που προκαλείται από τη σύνθεση και τις επισημάνσεις του κανόνα αλυσίδας στην αναπαραγωγή σφαλμάτων προς τα πίσω (backpropagation of errors) παρουσιάζεται κατά την ελαχιστοποίηση του σφάλματος ταξινόμησης.

Μια πλήρης γενίκευση του backprop περιλαμβάνει πολλαπλά επίπεδα καθώς και πολλαπλούς κόμβους ανά επίπεδο. Η γενική κατάσταση απεικονίζεται στο σχήμα 2.1. Ο στόχος είναι ο προσδιορισμός των στοιχείων μήτρας για κάθε μήτρα A_j . Έτσι ένας σημαντικός αριθμός των παραμέτρων του δικτύου πρέπει να επικαιροποιείται στην καθοδική κλίση. Πράγματι, η εκπαίδευση ενός δικτύου μπορεί συχνά να είναι υπολογιστικά ανέφικτη, παρόλο που οι ενημερωμένοι κανόνες για τα επιμέρους βάρη δεν είναι δύσκολοι. Έτσι, τα νευρωνικά δίκτυα μπορούν να καταγράφουν ζητήματα απορρέοντα από τις διαστάσεις τους καθώς κάθε μήτρα από το ένα επίπεδο στο άλλο απαιτεί ενημέρωση n^2 συντελεστών για μια n -διάστατη είσοδο, υποθέτοντας ότι τα δύο συνδεδεμένα στρώματα είναι και τα δύο n -διαστάσεων.

Δηλώνοντας όλα τα βάρη που πρέπει να ενημερωθούν από το διάνυσμα w, όπου w περιέχει όλα τα στοιχεία των πινάκων A_j που απεικονίζονται στην Εικόνα 2.1, τότε:

$$w_{k+1} = w_k + \delta\nabla E \tag{2.20}$$

όπου η κλίση του σφάλματος ∇E, μέσω του κανόνα σύνθεσης και αλυσίδας, παράγει τον αλγόριθμο backpropagation για την ενημέρωση των βαρών και τη μείωση του σφάλματος. Εκφρασμένο με τρόπο ανά στοιχείο:

$$w_{k+1}^j = w_k^j + \delta\frac{\partial E}{\partial w_k^j} \tag{2.21}$$

όπου αυτή η εξίσωση ισχύει για την j -ιοστή συνιστώσα του διανύσματος w. Ο όρος $\frac{\partial E}{\partial w^j}$ παράγει το backpropagation μέσω του κανόνα της αλυσίδας, δηλαδή παράγει το διαδοχικό σύνολο συναρτήσεων για εκτίμηση όπως στο (2.19). Μέθοδοι επίλυσης αυτής της βελτιστοποίησης πιο γρήγορα, ή ακόμα και απλά καθιστώντας τον υπολογισμό δυνατό, παραμένουν ενεργά ερευνητικά ενδιαφέροντα. Ίσως η πιο σημαντική μέθοδος είναι η στοχαστική καθοδική κλίση που εξετάζεται στην επόμενη ενότητα.

2.4 Ο στοχαστικός αλγόριθμος καθοδικής κλίσης

Η εκπαίδευση νευρωνικών δικτύων είναι υπολογιστικά δαπανηρή λόγω του μεγέθους των νευρωνικών δικτύων που εκπαιδεύονται. Ακόμη και τα νευρωνικά δίκτυα μέτριου μεγέθους μπορούν να γίνουν απαγορευτικά δαπανηρά εάν οι ρουτίνες βελτιστοποίησης που χρησιμοποιούνται για την εκπαίδευση δεν είναι καλά ενημερωμένες. Δύο αλγόριθμοι ήταν ιδιαίτερα κρίσιμοι για την ενεργοποίηση της εκπαίδευσης των νευρωνικών δικτύων: στοχαστική καθοδική κλίση (stochastic gradient descent -SGD) και backprop. Το backprop επιτρέπει έναν αποτελεσματικό υπολογισμό της κλίσης της συνάρτησης στόχου, ενώ η SGD παρέχει μια ταχύτερη αξιολόγηση των βέλτιστων βαρών δικτύου. Αν και εναλλακτικές μέθοδοι βελτιστοποίησης για την εκπαίδευση των νευρωνικών δικτύων συνεχίζουν να παρέχουν υπολογιστικές βελτιώσεις, το backprop και το SGD εξετάζονται εδώ λεπτομερώς, ώστε να δώσουν στον αναγνώστη μια ιδέα της βασικής αρχιτεκτονικής για την κατάρτιση νευρωνικών δικτύων.

Ο αλγόριθμος καθοδικής κλίσης αναπτύχθηκε για μη γραμμικές παλινδρομήσεις , όπου τα δεδομένα έχουν την εξής μορφή:

$$f(x) = f(x, \beta) \qquad (2.22)$$

,όπου β χρησιμοποιούνται συντελεστές προσαρμογής για την ελαχιστοποίηση του σφάλματος. Στα νευρωνικά δίκτυα, οι παράμετροι β είναι τα βάρη του δικτύου, έτσι μπορούμε να το ξαναγράψουμε με τη μορφή:

$$f(x) = f(x, A_1, A_2, \dots, A_M) \qquad (2.23)$$

,όπου το A_jείναι οι πίνακες συνδεσιμότητας από το ένα επίπεδο στο επόμενο στο νευρωνικό δίκτυο. Έτσι, το A_1συνδέει το πρώτο και το δεύτερο επίπεδο και υπάρχουν κρυμμένα επίπεδα M.

Ο στόχος της εκπαίδευσης του νευρωνικού δικτύου είναι να ελαχιστοποιήσει το σφάλμα μεταξύ του δικτύου και των δεδομένων. Το τυπικό σφάλμα τετραγώνου root-mean για αυτήν την περίπτωση ορίζεται ως:

$$argmin_{A_j} E(A_1, A_2, \dots, A_M) = argmin_{A_j} \sum_{k=1}^{n} ((f(x_k, A_1, A_2, \dots, A_M) - y_k)^2 \quad (2.24)$$

,η οποία μπορεί να ελαχιστοποιηθεί ρυθμίζοντας τη μερική παράγωγο σε σχέση με κάθε συστατικό μήτρας στο μηδέν, δηλαδή χρειαζόμαστε $\frac{\partial E}{\partial (a_{ij})_k} = 0$ όπου $(a_{ij})_k$ είναι η i-ιοστή γραμμή και j-ιοστή στήλη του k -ιοστού πίνακα ($k = 1,2, \dots, M$). Θυμηθείτε ότι η μηδενικη παράγωγος είναι ελάχιστο, δεδομένου ότι δεν υπάρχει μέγιστο σφάλμα. Αυτό δίνει την κλίση $\nabla f(x)$ της συνάρτησης σε σχέση με τις παραμέτρους του νευρωνικού δικτύου. Σημειώστε περαιτέρω ότι f(·) είναι η συνάρτηση που εκτιμάται σε κάθε ένα από τα n σημεία δεδομένων.

Αυτό οδηγεί σε ένα σχήμα επανάληψης Newton-Raphson για την εύρεση των ελαχίστων

$$x_{j+1}(\delta) = x_j - \delta \nabla f(x_j) \quad (2.25)$$

Όπου δ είναι μια παράμετρος που καθορίζει πόσο μακριά πρέπει να γίνει ένα βήμα κατά μήκος της κατεύθυνσης κλίσης. Στα νευρωνικά δίκτυα, αυτή η παράμετρος ονομάζεται ρυθμός μάθησης(learning rate). Σε αντίθεση με την τυπική καθοδική κλίσης(gradient descent), μπορεί να είναι υπολογιστικά απαγορευτικό να υπολογιστεί ένας βέλτιστος ρυθμό μάθησης.

Αν και η σύνθεση της βελτιστοποίησης δομείται εύκολα, ο υπολογισμός (2.24) είναι συχνά υπολογιστικά δυσεπίλυτος για τα νευρωνικά δίκτυα. Αυτό οφείλεται σε δύο λόγους: (i) ο αριθμός των παραμέτρων στάθμισης μήτρας για κάθε A_j είναι αρκετά μεγάλος και (ii) ο αριθμός των σημείων δεδομένων n είναι γενικά επίσης μεγάλος.

Για να καταστεί ο υπολογισμός (2.24) δυνητικά εφικτός, η SGD δεν εκτιμά την κλίση στο (2.25) χρησιμοποιώντας όλα τα n σημεία δεδομένων. Μάλλον, ένα μόνο, τυχαία το επιλεγμένο σημείο δεδομένων ή ένα υποσύνολο για την μερική καθοδική κλίση (batch gradient descent) χρησιμοποιείται για την προσέγγιση της κλίσης σε κάθε βήμα της επανάληψης. Σε αυτή την περίπτωση, μπορούμε να αναδιατυπώσουμε την διαμόρφωση ελαχίστων τετραγώνων (2.24) έτσι ώστε:

$$E(A_1, A_2, \dots, A_M) = \sum_{k=1}^{n} E_k(A_1, A_2, \dots, A_M) \quad (2.26)$$

Και

$$E_k(A_1, A_2, \dots, A_M) = (\ (f_k(x_k, A_1, A_2, \dots, A_M) - y_k)^2 \quad (2.27)$$

Όπου $f_k(\cdot)$ είναι τώρα η συνάρτηση προσαρμογής για κάθε σημείο δεδομένων και οι καταχωρήσεις του οι πίνακες A_j καθορίζονται από τη διαδικασία βελτιστοποίησης.

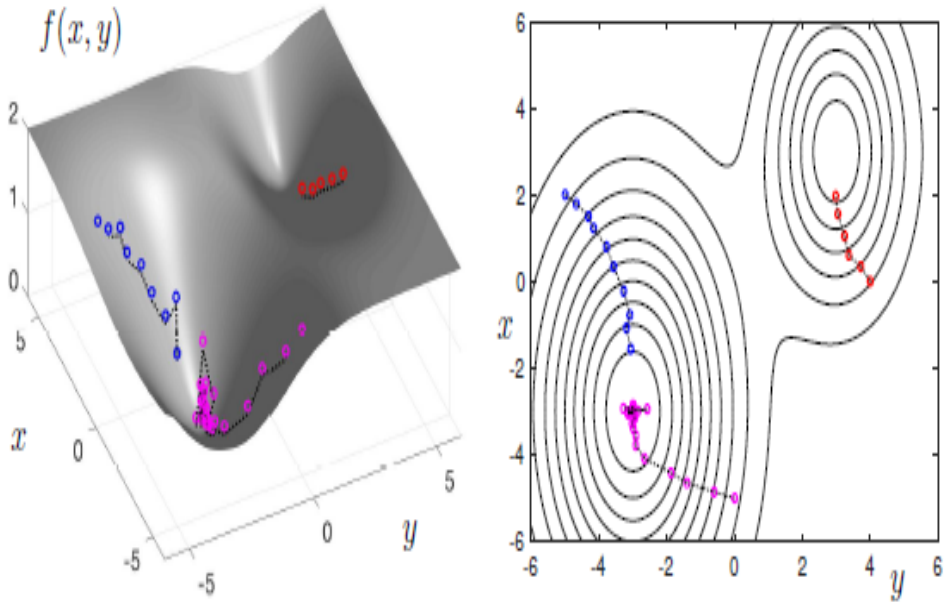
Ο επαναληπτικός αλγόριθμος καθοδικής κλίσης(gradient descent) (2.25) είναι τώρα ενημερωμένος ως εξής:

$$w_{j+1}(\delta) = w_j - \delta \nabla f_k(w_j) \quad (2.28)$$

,όπου w_j είναι το διάνυσμα όλων των βαρών δικτύου από το $A_j (j = 1,2, \dots, M)$ στην j -ιοστή επανάληψη και η κλίση υπολογίζεται χρησιμοποιώντας μόνο το k -ιοστό σημείο δεδομένων και $f_k(\cdot)$. Έτσι, αντί να υπολογιστεί η κλίση με όλα τα n σημεία, μόνο ένα το σημείο δεδομένων επιλέγεται και χρησιμοποιείται τυχαία. Στην επόμενη επανάληψη, ένα άλλο τυχαία επιλεγμένο σημείο χρησιμοποιείται για τον υπολογισμό της κλίσης και την ενημέρωση της λύσης. Ο αλγόριθμος μπορεί να απαιτεί πολλαπλά περάσματα μέσα από όλα τα δεδομένα για να συγκλίνει, αλλά κάθε βήμα είναι πλέον εύκολο να αξιολογηθεί σε σχέση με τον ακριβό υπολογισμό της Jacobian που απαιτείται για την κλίση. Εάν αντί για ένα μόνο σημείο, χρησιμοποιείται ένα υποσύνολο των σημείων , τότε έχουμε τον ακόλουθο αλγόριθμο μερικής καθοδικής κλίσης (batch gradient descent)

$$w_{j+1}(\delta) = w_j - \delta \nabla f_k(w_j)$$

όπου $K\ E\ [K_1, K_2, \dots, K_p]$ δηλώνει τα p τυχαία επιλεγμένα σημεία δεδομένων K_j που χρησιμοποιούνται για την προσέγγιση της κλίσης.



Σχήμα 2.7: Στοχαστική καθοδική κλίση .Η σύγκλιση μπορεί να συγκριθεί με έναν αλγόριθμο πλήρους καθοδικής κλίσης . Κάθε βήμα της καθοδικής στοχαστικής (παρτίδας) κλίσης επιλέγει 100 σημεία δεδομένων για την προσέγγιση του gradient, αντί για το 10^4 σημεία δεδομένων των δεδομένων. Εμφανίζονται τρεις αρχικές συνθήκες $(x_0, y_0) = \{(4,0), (0,-5), (-5,2)\}$. Ο πρώτος από αυτούς (κόκκινοι κύκλοι) κολλάει σε ένα τοπικό ελάχιστο ενώ οι άλλες δύο αρχικές συνθήκες (μπλε και μοβ) βρίσκουν το γενικό ελάχιστο.

ΚΕΦΑΛΑΙΟ 3ο

Ενίσχυση (reinforce)

3.1 Εισαγωγή

Ο αλγόριθμος REINFORCE(ΕΝΙΣΧΥΣΗ), που αναπτύχθηκε από τον Ronald J. Williams το 1992 στην εργασία του "Απλοί στατιστικοί αλγόριθμοι που ακολουθούν την μέθοδο κλίσης για τη συνδετική ενισχυτική μάθηση» (Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning)[27], μαθαίνει μια παραμετροποιημένη πολιτική που παράγει πιθανότητες δράσης από καταστάσεις. Οι πράκτορες χρησιμοποιούν αυτήν την πολιτική απευθείας για να ενεργούν σε ένα περιβάλλον. Η βασική ιδέα είναι ότι κατά τη διάρκεια της μάθησης, οι ενέργειες που οδήγησαν σε καλά αποτελέσματα θα πρέπει γίνονται πιο πιθανές - αυτές οι ενέργειες ενισχύονται θετικά. Αντιστρόφως, οι δράσεις με κακά αποτελέσματα θα πρέπει να γίνουν λιγότερο πιθανές. Εάν η μάθηση είναι επιτυχής, με το πέρας πολλών επαναλήψεων οι παραγόμενες πιθανότητες δράσης μετατοπίζονται στην κατανομή που έχει ως αποτέλεσμα την καλή απόδοση σε ένα περιβάλλον. Οι πιθανότητες δράσης αλλάζουν ακολουθώντας την πολιτική κλίσης , επομένως η ΕΝΙΣΧΥΣΗ είναι γνωστός ως πολιτικής κλίσης αλγόριθμος.

Ο αλγόριθμος χρειάζεται τρία συστατικά:

1. Μια παραμετροποιημένη πολιτική.
2. Ένα στόχο που πρέπει να μεγιστοποιηθεί.
3. Μια μέθοδο για την ενημέρωση των παραμέτρων πολιτικής.

Η ενότητα 3.2 εισάγει παραμετροποιημένες πολιτικές. Στη συνέχεια, στην Ενότητα 3.3 συζητάμε τη συνάρτηση στόχου(objective function) που καθορίζει τον τρόπο αξιολόγησης των αποτελεσμάτων. Το τμήμα 3.4 περιέχει τον πυρήνα του αλγορίθμου ΕΝΙΣΧΥΣΗ —η πολιτική κλίσης. Η πολιτική κλίσης παρέχει έναν τρόπο για την εκτίμηση της κλίσης του στόχου σε σχέση με τις παραμέτρους πολιτικής. Αυτό είναι ένα σημαντικό βήμα, δεδομένου ότι η πολιτική κλίσης χρησιμοποιείται για την τροποποίηση των παραμέτρων πολιτικής, ώστε να μεγιστοποιήσει τον στόχο.

3.2 Πολιτική (Policy)

Στον αλγόριθμο REINFORCE, ένας πράκτορας μαθαίνει μια πολιτική και τη χρησιμοποιεί για να ενεργήσει σε ένα περιβάλλον.

Τα νευρωνικά δίκτυα είναι ισχυροί και ευέλικτοι προσεγγιστές συναρτήσεων, ώστε να μπορούμε να αναπαραστήσουμε μια πολιτική χρησιμοποιώντας ένα βαθύ νευρωνικό δίκτυο που αποτελείται από μαθησιακές παραμέτρους θ. Αυτό συχνά αναφέρεται ως δίκτυο πολιτικής π_θ. Λέμε ότι η πολιτική παραμετροποιείται από θ.

Κάθε συγκεκριμένο σύνολο τιμών των παραμέτρων του δικτύου πολιτικής αντιπροσωπεύει μια συγκεκριμένη πολιτική. Για να δούμε γιατί, ας σκεφτούμε $\theta_1 \neq \theta_2$. Για οποιαδήποτε δεδομένη κατάσταση, διαφορετικά δίκτυα πολιτικής μπορεί να παράγουν διαφορετικά σύνολα πιθανοτήτων δράσης, δηλαδή $\pi_{\theta_1}(s) \neq \pi_{\theta_2}(s)$. Οι αντιστοιχίσεις από τις πιθανότητες δράσης σε καταστάσεις είναι διαφορετικές, οπότε λέμε ότι τα π_{θ_1} και π_{θ_2} είναι διαφορετικές πολιτικές. Ένα ενιαίο νευρωνικό δίκτυο είναι επομένως ικανό να αντιπροσωπεύει πολλές διαφορετικές πολιτικές.

Διατυπωμένη με αυτόν τον τρόπο, η διαδικασία μάθησης μιας καλής πολιτικής αντιστοιχεί στην αναζήτηση ενός καλού συνόλου τιμών για το θ. Για το λόγο αυτό, είναι σημαντικό το δίκτυο πολιτικής να είναι διαφορίσιμο. Θα δούμε στην Ενότητα 3.3 ότι ο μηχανισμός με τον οποίο η πολιτική βελτιώνεται είναι μέσω της ανοδικής κλίσης (gradient ascent) στο χώρο παραμέτρων.

3.3 Η συνάρτηση στόχου (objective function)

Σε αυτήν την ενότητα, ορίζουμε τον στόχο που μεγιστοποιείται από έναν πράκτορα στον αλγόριθμο ΕΝΊΣΧΥΣΗ. Ένας σκοπός μπορεί να γίνει κατανοητός ως στόχος ενός πράκτορα, όπως όταν κερδίζουμε ένα παιχνίδι ή παίρνουμε την υψηλότερη δυνατή βαθμολογία. Πρώτον, εισάγουμε την έννοια της επιστροφής, η οποία υπολογίζεται χρησιμοποιώντας μια τροχιά. Στη συνέχεια, τη χρησιμοποιούμε για να διατυπώσουμε τον στόχο.

Θυμόμαστε από το Κεφάλαιο 1 ότι ένας παράγοντας που ενεργεί σε ένα περιβάλλον δημιουργεί μια τροχιά, η οποία περιέχει μια ακολουθία ανταμοιβών μαζί με τις καταστάσεις και τις ενέργειες. Μια τροχιά συμβολίζεται ως: $\tau = s_0, a_0, r_0, \ldots, s_T, a_T, r_T$.

Η επιστροφή μιας τροχιάς $R_t(\tau)$ ορίζεται ως ένα προεξοφλημένο άθροισμα ανταμοιβών από το χρονικό βήμα t μέχρι το τέλος μιας τροχιάς, όπως φαίνεται στην εξίσωση 3.1.

$$R_t(\tau) = \sum_{t'=t}^{T} \gamma^{t'-t} r_t' \qquad (3.1)$$

Να σημειωθεί ότι το άθροισμα ξεκινά από το βήμα χρόνου t, αλλά η δύναμη στην οποία υψώνεται ο συντελεστής προεξόφλησης γ , κατά την άθροιση για την επιστροφή, ξεκινά από το 0, οπότε πρέπει να αντισταθμίσουμε την ισχύ με το βήμα του χρόνου έναρξης t χρησιμοποιώντας $t' - t$.

Όταν $t = 0$, η επιστροφή είναι απλώς η επιστροφή της πλήρους τροχιάς. Αυτό είναι επίσης γραμμένο ως $R_0(\tau) = R(\tau)$ για συντομία. Στόχος είναι η αναμενόμενη απόδοση των συνολικών τροχιών που δημιουργούνται από έναν πράκτορα. Αυτό ορίζεται στην εξίσωση 3.2.

$$J(\pi_\theta) = E_{\tau\sim\pi_\theta}[R(\tau)] = E_{\tau\sim\pi_\theta} \sum_{t=0}^{T} \gamma^t r_t \qquad (3.2)$$

Η εξίσωση 3.2 λέει ότι η προσδοκία υπολογίζεται σε πολλές τροχιές από μια πολιτική, δηλαδή, $\tau\sim\pi_\theta$. Αυτή η προσδοκία προσεγγίζει την πραγματική τιμή καθώς συγκεντρώνονται περισσότερα δείγματα και συνδέονται με τη συγκεκριμένη πολιτική π_θ που χρησιμοποιείται.

3.4 Η πολιτική κλίσης (policy gradient)

Έχουμε πλέον καθορίσει την πολιτική π_θ και τον στόχο $J(\pi_\theta)$ που είναι δύο κρίσιμα στοιχεία για την εξαγωγή του αλγορίθμου πολιτικής κλίσης. Η πολιτική παρέχει έναν τρόπο στον πράκτορα για να δράσει και ο σκοπός παρέχει έναν στόχο για μεγιστοποίηση.

Το τελευταίο στοιχείο του αλγορίθμου είναι η πολιτική κλίσης. Επισήμως, λέμε ο αλγόριθμος πολιτικής κλίσης επιλύει το ακόλουθο πρόβλημα:

$$\max_\theta J(\pi_\theta) = E_{\tau\sim\pi_\theta}[R(\tau)] \qquad (3.3)$$

Για να μεγιστοποιήσουμε τον στόχο, εκτελούμε ανύψωση κλίσης στις παραμέτρους πολιτικής θ . Θυμηθείτε από τον λογισμό ότι η κλίση δείχνει προς την κατεύθυνση της πιο απότομης ανάβασης. Για βελτίωση του στόχου, υπολογίζουμε την κλίση και την χρησιμοποιούμε για να ενημερώσουμε τις παραμέτρους όπως φαίνεται στο εξίσωση 3.4

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta) \tag{3.4}$$

Όπου ,α είναι ένα βαθμωτό μέγεθος, γνωστό ως ρυθμός μάθησης, το οποίο ελέγχει το μέγεθος της παραμέτρου ενημέρωσης. Ο όρος $\nabla_\theta J(\pi_\theta)$ είναι γνωστός ως πολιτική κλίσης. Ορίζεται στην εξίσωση 3.5.

$$\nabla_\theta J(\pi_\theta) = E_{\tau \sim \pi_\theta} \sum_{t=0}^{T} R_t(\tau)\, \nabla_\theta \log \pi_\theta\, (a_t|s_t) \tag{3.5}$$

Ο όρος $\pi_\theta(a_t|s_t)$ είναι η πιθανότητα της ενέργειας που αναλαμβάνει ο πράκτορας στο χρονικό βήμα t. Η δράση αποτελεί δείγμα από την πολιτική, στο $a_t \sim \pi_\theta(s_t)$. Η δεξιά πλευρά της εξίσωσης δηλώνει ότι η κλίση της λογαριθμικής πιθανότητας της ενέργειας σε σχέση με το θ πολλαπλασιάζεται επί την επιστροφή $R_t(\tau)$.

Η εξίσωση 3.5 δηλώνει ότι η κλίση του στόχου είναι ισοδύναμη με το αναμενόμενο άθροισμα των κλίσεων των λογαριθμικών πιθανοτήτων των ενεργειών a_t πολλαπλασιασμένες με τις αντίστοιχες επιστροφές $R_t(\tau)$.

Η πολιτική κλίσης είναι ο μηχανισμός με τον οποίο οι πιθανότητες δράσης που παράγονται από την πολιτική αλλάζουν. Εάν η επιστροφή$R_t(\tau) > 0$, τότε η πιθανότητα της ενέργειας $\pi_\theta(a_t|s_t)$ αυξάνεται. Αντίθετα, αν η επιστροφή $R_t(\tau) < 0$, τότε η πιθανότητα της ενέργειας $\pi_\theta(a_t|s_t)$ μειώνεται. Κατά τη διάρκεια πολλών ενημερώσεων (εξίσωση 3.4), η πολιτική θα μάθει να παράγει ενέργειες που οδηγούν σε υψηλό $R_t(\tau)$.

Η εξίσωση 3.5 είναι το θεμέλιο όλων των μεθόδων πολιτικής κλίσης. Η ΕΝΙΣΧΥΣΗ ήταν ο πρώτος αλγόριθμος που το χρησιμοποίησε στην απλούστερη μορφή του. Οι νεότεροι αλγόριθμοι βασίζονται σε αυτό με τροποποίηση της συνάρτησης για τη βελτίωση της απόδοσης. Ωστόσο, παραμένει ένα τελευταίο ερώτημα - πώς μπορεί κανείς να εφαρμόσει και να εκτιμήσει την ιδανική εξίσωση κλίσης πολιτικής.

3.4.1 Πηγή της πολιτικής κλίσης.

Εδώ εξάγουμε την πολιτική κλίσης (εξίσωση 3.5) από την κλίση του στόχου όπως παρουσιάζεται στην εξίσωση 3.6. Να σημειωθεί ότι αυτή η ενότητα μπορεί να παραλειφθεί σε πρώτη ανάγνωση.

$$\nabla_\theta J(\pi_\theta) = \nabla_\theta E_{\tau\sim\pi_\theta}[R(\tau)] \qquad (3.6)$$

Η εξίσωση 3.6 παρουσιάζει ένα πρόβλημα επειδή δεν μπορούμε να διαφορίσουμε τον όρο $R(\tau)$= $\sum_{t=0}^{T} \gamma^t\, r_t$ σε σχέση με το θ. Οι ανταμοιβές r_t δημιουργούνται από μια άγνωστη συνάρτηση ανταμοιβής $R(s_t, a_t, s_{t+1})$,η οποία δεν μπορεί να διαφοριστεί. Ο μόνος τρόπος για τις μεταβλητές πολιτικής θ να επηρεάσουν το $R(\tau)$ είναι με την αλλαγή της κατάστασης και των κατανομών δράσης που, με τη σειρά τους, αλλάζουν τις ανταμοιβές που λαμβάνει ένας πράκτορας.

Επομένως, πρέπει να μετατρέψουμε την εξίσωση 3.6 σε μια μορφή όπου μπορούμε να πάρουμε μια κλίση σε σχέση με θ. Για να το κάνουμε αυτό, θα χρησιμοποιήσουμε ορισμένες εύχρηστες ταυτότητες.

Δεδομένης μιας συνάρτησης $f(x)$, μιας παραμετροποιημένης κατανομής πιθανότητας $p(x, \theta)$, και της προσδοκίας $\mathbb{E}_{x\sim p(x,\theta)}[f(x)]$, η κλίση της προσδοκίας μπορεί να ξαναγραφεί ως εξής:

$$\nabla_\theta \mathbb{E}_{x\sim p(x,\theta)}[f(x)]$$

$$= \nabla_\theta \int dx\, f(x)\, p(x,\theta) \qquad \textit{(ορισμός προσδοκίας)} \quad (3.7)$$

$$= \int dx\, \nabla_\theta f(x)\, p(x,\theta) \qquad \textit{(φέρνουμε μέσα τον όρο } \nabla_\theta) \quad (3.8)$$

$$= \int dx\, (f(x)\nabla_\theta\, p(x,\theta) + p(x,\theta)\nabla_\theta f(x)) \qquad \textit{(κανόνας αλυσίδας)} \quad (3.9)$$

$$= \int dx\, f(x)\nabla_\theta\, p(x,\theta) \qquad (\nabla_\theta\, p(x,\theta) = 0) \quad (3.10)$$

$$= \int dx\, f(x)\, p(x,\theta)\, \tfrac{\nabla_\theta p(x,\theta)}{p(x,\theta)} \qquad \textit{(πολλαπλασιασμός με } \tfrac{p(x,\theta)}{p(x,\theta)}) \quad (3.11)$$

$$= \int dx\, f(x)\, p(x,\theta)\nabla_\theta \log p(x,\theta) \qquad (3.12)$$

$= \mathbb{E}_x[f(x)\nabla_\theta \log p(x,\theta)]$ *(ορισμός προσδοκίας)* *(3.13)*

Αυτή η ταυτότητα λέει ότι η κλίση μιας προσδοκίας είναι ισοδύναμη με την προσδοκία της κλίσης της λογαριθμικής πιθανότητας πολλαπλασιαζόμενη με την αρχική συνάρτηση. Η πρώτη γραμμή είναι απλά ο ορισμός μιας προσδοκίας. Μια ολοκληρωμένη μορφή χρησιμοποιείται γενικότερα λαμβάνοντας υπόψη ότι η $f(x)$ να είναι μια συνεχής συνάρτηση, αλλά ισχύει εξίσου για μια μορφή άθροισης για μια διακριτή συνάρτηση.

Παρατηρούμε ότι η εξίσωση 3.10 έχει λύσει το αρχικό μας πρόβλημα αφού μπορούμε να πάρουμε την κλίση του $p(x,\theta)$, αλλά το $f(x)$ είναι μια συνάρτηση μαύρου κουτιού (black box function) που δεν μπορεί να ενσωματωθεί. Για να αντιμετωπιστεί αυτό, πρέπει να μετατρέψουμε την εξίσωση σε προσδοκία, ώστε να μπορεί να εκτιμηθεί μέσω δειγματοληψίας. Πρώτον, το πολλαπλασιάζουμε πανομοιότυπα με $\frac{p(x,\theta)}{p(x,\theta)}$ στην εξίσωση 3.11. Το προκύπτον κλάσμα $\frac{\nabla_\theta p(x,\theta)}{p(x,\theta)}$ μπορεί να ξαναγραφεί με το λογαριθμικό παράγωγο τέχνασμα στην Εξίσωση 3.14.

$$\nabla_\theta \log p(x,\theta) = \frac{\nabla_\theta p(x,\theta)}{p(x,\theta)} \qquad (3.14)$$

Η αντικατάσταση της εξίσωσης 3.14 στην 3.11 δίνει την εξίσωση 3.12. Αυτό μπορεί να γραφτεί ως μια προσδοκία ούτως ώστε να δοθεί η εξίσωση 3.13. Τέλος, απλά ξαναγράφουμε την έκφραση ως προσδοκία.

Τώρα, θα πρέπει να είναι προφανές ότι αυτή η ταυτότητα μπορεί να εφαρμοστεί στον στόχο μας.Αντικαθιστώντας $x = \tau, f(x) = R(\tau), p(x|\theta) = p(\tau,\theta)$,η εξίσωση 3.6 μπορεί να γραφτεί ως:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[R(\tau)\nabla_\theta \log p(\tau,\theta)] \qquad (3.15)$$

Ωστόσο, ο όρος $p(\tau,\theta)$ στην εξίσωση 3.15 πρέπει να σχετίζεται με την πολιτική π_θ, της οποίας έχουμε τον έλεγχο. Ως εκ τούτου, πρέπει να επεκταθεί περαιτέρω.

Παρατηρούμε ότι η τροχιά τ είναι απλώς μια ακολουθία παρεμβαλλόμενων γεγονότων, a_t και s_{t+1}, δείγματα, αντίστοιχα, από την πιθανότητα δράσης του παράγοντα $\pi_\theta(a_t, s_t)$ και την πιθανότητα μετάβασης περιβάλλοντος $p(s_{t+1}|s_t, a_t)$. Δεδομένου ότι

οι πιθανότητες είναι υπό όρους ανεξάρτητες, η πιθανότητα ολόκληρης της τροχιάς είναι το γινόμενο των μεμονωμένων πιθανοτήτων, όπως παρουσιάζεται στην εξίσωση 3.16.

$$p(\tau,\theta) = \prod_{t\geq 0} p(s_{t+1}|s_t,a_t)\pi_\theta(a_t,s_t) \qquad (3.16)$$

Εφαρμόζουμε λογάριθμους και στις δύο πλευρές για να ταιριάξουμε την εξίσωση 3.16 με την εξίσωση 3.15.

$$\log p(\tau,\theta) = \log \prod_{t\geq 0} p(s_{t+1}|s_t,a_t)\pi_\theta(a_t,s_t) \qquad (3.17)$$

$$\log p(\tau,\theta) = \sum_{t\geq 0}(\log p(s_{t+1}|s_t,a_t) + \log \pi_\theta(a_t,s_t)) \qquad (3.18)$$

$$\nabla_\theta \log p(\tau,\theta) = \nabla_\theta \sum_{t\geq 0}(\log p(s_{t+1}|s_t,a_t) + \log \pi_\theta(a_t,s_t)) \qquad (3.19)$$

$$\nabla_\theta \log p(\tau,\theta) = \nabla_\theta \sum_{t\geq 0} \log \pi_\theta(a_t,s_t) \qquad (3.20)$$

Η εξίσωση 3.18 προκύπτει από το γεγονός ότι ο λογάριθμος του γινομένου ισούται με το άθροισμα των λογάριθμων των συστατικών του. Από εκεί, μπορούμε να εφαρμόσουμε την κλίση ∇_θ και στις δύο πλευρές παίρνοντας την εξίσωση 3.19. Η κλίση μπορεί να μετακινηθεί μέσα στους όρους άθροισης. Από τη στιγμή που το $\log p(s_{t+1}|s_t,a_t)$ είναι ανεξάρτητο από το θ, η κλίση του είναι μηδέν και μπορεί να αφαιρεθεί. Αυτό αποδίδει την εξίσωση 3.20 ,που εκφράζει την πιθανότητα $p(\tau,\theta)$ με όρους $\pi_\theta(a_t,s_t)$. Επίσης να σημειωθεί ότι η τροχιά τ στα αριστερά αντιστοιχεί σε μια άθροιση των μεμονωμένου χρόνου βημάτων t στα δεξιά.

Με αυτό, είμαστε επιτέλους έτοιμοι να ξαναγράψουμε $\nabla_\theta J(\pi_\theta)$ από την εξίσωση 3.6 σε μια μορφή που μπορεί να διαφοριστεί. Αντικαθιστώντας την εξίσωση 3.20 με την 3.15 και φέρνοντας μέσα το πολλαπλασιαστή $R(\tau)$,, λαμβάνουμε:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[\sum_{t=0}^{T} R_t(\tau)\nabla_\theta \log \pi_\theta(a_t,s_t)] \qquad (3.21)$$

Το πρόβλημά μας ήταν ότι η εξίσωση 3.6 περιείχε μια συνάρτηση που δεν ήταν διαφορίσιμη. Μετά από μια σειρά μετασχηματισμών, φτάσαμε στην Εξίσωση 3.20. Αυτό μπορεί να εκτιμηθεί αρκετά χρησιμοποιώντας εύκολα ένα δίκτυο πολιτικής π_θ , με τον υπολογισμό gradient να γίνεται από την αυτόματη δυνατότητα διαφόρισης ,η οποία είναι διαθέσιμη σε βιβλιοθήκες νευρωνικών δικτύων.

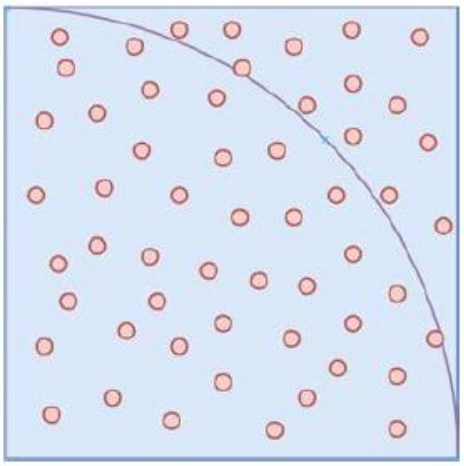
3.5 Δειγματοληψία Monte-Carlo

Ο αλγόριθμος ΕΝΙΣΧΥΣΗ υπολογίζει αριθμητικά την πολιτική κλίσης χρησιμοποιώντας δειγματοληψία Monte-Carlo. Η δειγματοληψία Monte Carlo αναφέρεται σε οποιαδήποτε μέθοδο που χρησιμοποιεί τυχαία δειγματοληψία για τη δημιουργία δεδομένων που χρησιμοποιούνται για την προσέγγιση μιας συνάρτησης. Στην ουσία, είναι απλώς "προσέγγιση με τυχαία δειγματοληψία». Είναι μια τεχνική που έγινε δημοφιλής χάρη στον Stanislaw Ulam, έναν μαθηματικό που εργάστηκε στο ερευνητικό εργαστήριο του Los Alamos τη δεκαετία του 1940.

Για να δούμε πώς λειτουργεί το Monte Carlo, ας δούμε ένα παράδειγμα για το πώς μπορεί να χρησιμοποιηθεί για να υπολογίσουμε την τιμή του π (η μαθηματική σταθερά)—ο λόγος της περιφέρειας ενός κύκλου προς τη διάμετρό του. Μια προσέγγιση του Monte Carlo για την επίλυση αυτού του προβλήματος είναι να ληφθεί ένας κύκλος ακτίνας r = 1 κεντραρισμένος στην αρχή και να τον εγγράψουμε σε ένα τετράγωνο. Οι περιοχές τους είναι πr^2 και $(2r)^2$, αντίστοιχα. Ως εκ τούτου, η αναλογία αυτών των περιοχών είναι απλά:

$$\frac{area\ of\ circle}{area\ of\ square} = \frac{\pi r^2}{(2r)^2} = \frac{\pi}{4} \qquad (3.22)$$

Αριθμητικά, το τετράγωνο έχει εμβαδόν 4, αλλά επειδή δεν γνωρίζουμε ακόμα το π, το εμβαδόν του κύκλου είναι άγνωστο. Ας εξετάσουμε ένα τεταρτημόριο. Για να ληφθεί μια εκτίμηση για το π, δειγματοληπτούνται πολλά σημεία μέσα στο τετράγωνο χρησιμοποιώντας μια ομοιόμορφα τυχαία κατανομή. Ένα σημείο (x, y) που βρίσκεται στον κύκλο έχει απόσταση μικρότερη από 1 από την αρχή-δηλαδή, $\sqrt{(x-0)^2 + (y-0)^2} \leq 1$. Αυτό φαίνεται στο σχήμα 3.1. Χρησιμοποιώντας αυτό, αν μετρήσουμε τον αριθμό σημείων στον κύκλο και, στη συνέχεια, μετρήσουμε τον αριθμό των σημείων δειγματοληψίας συνολικά, η αναλογία τους είναι περίπου ίση με την εξίσωση 3.22. Με επαναληπτική δειγματοληψία περισσότερων σημείων και ενημέρωση της αναλογίας, η εκτίμησή μας θα πλησιάσει την ακριβή τιμή. Πολλαπλασιασμός αυτής της αναλογίας επί 4 μας δίνει την εκτιμώμενη τιμή $\pi \approx 3.14159$.



Σχήμα 3.1 Δειγματοληψία Monte-Carlo χρησιμομοιηθείσα για τον υπολογισμό π.

Τώρα, ας επιστρέψουμε στη βαθιά ενισχυτική μάθηση και να δούμε πώς το Monte Carlo μπορεί να χρησιμοποιηθεί αριθμητικά για την εκτίμηση της κλίσης πολιτικής στην εξίσωση 3.5. Είναι πολύ απλό. Η προσδοκία $\mathbb{E}_{\tau\sim\pi_\theta}$υπονοεί ότι καθώς λαμβάνονται δείγματα από περισσότερες τροχιές τ_s χρησιμοποιώντας μια πολιτική π_θ και υπολογίζεται ο μέσος όρος, προσεγγίζει την πραγματική πολιτική κλίσης $\nabla_\theta J(\pi_\theta)$. Αντί της δειγματοληψίας πολλών τροχιών ανά πολιτική, μπορούμε να δοκιμάσουμε μόνο μια όπως φαίνεται στην Εξίσωση 3.23.

$$\nabla_\theta J(\pi_\theta) \approx \sum_{t=0}^{T} R_t(\tau)\nabla_\theta \log \pi_\theta(a_t, s_t) \qquad (3.23)$$

Αυτός είναι ο τρόπος με τον οποίο εφαρμόζεται η πολιτική κλίσης— όπως εκτιμά το Monte Carlo σε δειγματοληπτικές τροχιές.

3.6 Ο αλγόριθμος ενίσχυσης (REINFORCE algorithm)

Αυτή η ενότητα εξετάζει τον αλγόριθμο ενίσχυσης και εισάγει την έννοια του αλγόριθμου πολιτικής(on-policy algorithm). Στη συνέχεια, εξετάζουμε ορισμένους από τους περιορισμούς του και εισάγουμε μια γραμμή βάσης για την βελτίωση της απόδοσης.

Ο αλγόριθμος εμφανίζεται στον **Αλγόριθμο 3.1**. Είναι πολύ απλό. Πρώτον, αρχικοποιούμε το ρυθμό μάθησης (rate) α και δημιουργούμε ένα δίκτυο πολιτικής π_θ με τυχαία αρχικοποιημένα βάρη.

Στη συνέχεια, επαναλαμβάνουμε για πολλά επεισόδια ως εξής: χρησιμοποιούμε το δίκτυο πολιτικής π_θ για να δημιουργήσουμε μια τροχιά $\tau = s_0, a_0, r_0, \dots, s_T, a_T, r_T$ για ένα επεισόδιο. Στη συνέχεια, για κάθε χρονικό βήμα t στη τροχιά, υπολογίζουμε την επιστροφή $R_t(\tau)$. Χρησιμοποιούμε το $R_t(\tau)$ για να υπολογίσουμε την πολιτική κλίσης. Αθροίζουμε τις πολιτικές κλίσης για όλα τα χρονικά βήματα και, στη συνέχεια, χρησιμοποιούμε το αποτέλεσμα για να ενημερώσουμε τις παραμέτρους θ του δικτύου πολιτικής.

Αλγόριθμος 3.1 Αλγόριθμος ενίσχυσης

1: Initialize learning rate α
2: Initialize weights θ of a policy network π_θ
3: **for** *episode* $= 0, \dots,$ *MAX_EPISODE* **do**
4: Sample a trajectory $\tau = s_0, a_0, r_0, \dots, s_T, a_T, r_T$
5: Set $\nabla_\theta J(\pi_\theta) = 0$
6: **for** $t = 0, \dots, T$ **do**
7: $R_t(\tau) = \sum_{t'=t}^{T} \gamma^{t'-t} r'_t$
8: $\nabla_\theta J(\pi_\theta) = \nabla_\theta J(\pi_\theta) + R_t(\tau) \nabla_\theta \log \pi_\theta(a_t \mid s_t)$
9: **end for**
10: $\theta = \theta + \alpha \nabla_\theta J(\pi_\theta)$
11: **end for**

Είναι σημαντικό μια τροχιά να απορρίπτεται μετά από κάθε ενημέρωση παραμέτρων , αφού δεν μπορεί να επαναχρησιμοποιηθούν. Αυτό συμβαίνει επειδή η ΕΝΙΣΧΥΣΗ είναι ένας αλγόριθμος πολιτικής. Θυμηθείτε ότι ένας αλγόριθμος είναι ενσωματωμένος στην πολιτική, εάν η εξίσωση ενημέρωσης παραμέτρων εξαρτάται από την τρέχουσα πολιτική. Αυτό είναι σαφές από τη γραμμή 8, δεδομένου ότι η πολιτική κλίσης εξαρτάται άμεσα από τις πιθανότητες ενέργειας $\pi_\theta(a_t, s_t)$ που δημιουργούνται από την τρέχουσα πολιτική π_θ, αλλά όχι από κάποια προηγούμενη πολιτική $\pi_{\theta'}$. Αντίστοιχα, η επιστροφή $R_t(\tau)$, όπου $\tau \sim \pi_\theta$,πρέπει επίσης να δημιουργηθεί από π_θ, διαφορετικά, οι πιθανότητες ενέργειας θα προσαρμοστούν με βάση τις αποδόσεις που η πολιτική δεν θα είχε δημιουργήσει.

3.6.1 Βελτίωση Αλγορίθμου

Η διατύπωση του αλγορίθμου ΕΝΙΣΧΥΣΗ εκτιμά την πολιτική κλίσης χρησιμοποιώντας δειγματοληψία Monte Carlo με μία μόνο τροχιά. Αυτή είναι μια αμερόληπτη εκτίμηση της πολιτικής κλίσης, αλλά ένα μειονέκτημα αυτής της προσέγγισης είναι ότι έχει μεγάλη διακύμανση. Σε αυτό το σημείο, εισάγουμε μια γραμμή βάσης για να μειώσουμε τη διακύμανση της εκτίμησης. Μετά από αυτό, θα συζητηθεί επίσης η εξομάλυνση των ανταμοιβών για την αντιμετώπιση του ζητήματος της κλιμάκωσής τους.

Ένας τρόπος για να μειώσουμε τη διακύμανση της εκτίμησης είναι να τροποποιήσουμε τις αποδόσεις αφαιρώντας μια κατάλληλη τιμή βάσης ανεξάρτητη από τη δράση, όπως φαίνεται στην εξίσωση 3.24.

$$\nabla_\theta J(\pi_\theta) \approx \sum_{t=0}^{T}(R_t(\tau) - b(s_t))\nabla_\theta \log \pi_\theta(a_t, s_t) \qquad (3.24)$$

Μια επιλογή για τη γραμμή βάσης είναι η συνάρτηση αξίας V^π. Αυτή η επιλογή της γραμμής βάσης ενεργοποιεί τον αλγόριθμο Actor-Critic.

Μια εναλλακτική λύση είναι να χρησιμοποιήσουμε τις μέσες αποδόσεις κατά τη διάρκεια της τροχιάς. Έστω $b = \frac{1}{T}\sum_{t=0}^{T} R_t(\tau)$. Να σημειωθεί ότι αυτή είναι μια σταθερή γραμμή βάσης ανά τροχιά που δεν διαφέρει ανάλογα με την κατάσταση s_t. Έχει σαν αποτέλεσμα την επικέντρωση των επιστροφών για κάθε τροχιά γύρω από το 0. Για κάθε τροχιά, κατά μέσο όρο, το καλύτερο 50% των δράσεων θα ενθαρρυνθεί και οι άλλες θα αποθαρρυνθούν.

Για να δούμε γιατί αυτό είναι χρήσιμο, εξετάζουμε την περίπτωση όπου όλες οι ανταμοιβές για ένα περιβάλλον είναι αρνητικές. Χωρίς μια γραμμή βάσης, ακόμη και όταν ένας πράκτορας παράγει μια πολύ καλή δράση, αποθαρρύνεται επειδή οι αποδόσεις είναι πάντα αρνητικές. Με την πάροδο του χρόνου, αυτό μπορεί ακόμα να οδηγήσει σε καλές πολιτικές αφού οι χειρότερες ενέργειες θα αποθαρρυνθούν ακόμη περισσότερο, αυξάνοντας έτσι έμμεσα τις πιθανότητες καλύτερων ενεργειών. Ωστόσο, μπορεί να οδηγήσει σε πιο αργή μάθηση επειδή οι προσαρμογές πιθανότητας μπορούν να γίνουν μόνο προς μία κατεύθυνση. Το αντίστροφο συμβαίνει για περιβάλλοντα όπου όλες οι ανταμοιβές είναι θετικές. Η μάθηση είναι πιο αποτελεσματική όταν μπορούμε και να αυξήσουμε αλλά και να μειώσουμε τις πιθανότητες δράσης. Αυτό απαιτεί να έχουμε τόσο θετικές όσο και αρνητικές επιστροφές.

ΚΕΦΑΛΑΙΟ 4ο

Συνάρτηση πλεονεκτήματος σε αλγόριθμους Actor-Critic (A2C)

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο, εξετάζουμε τους αλγόριθμους Πράκτορα-Κριτή (Actor-Critic) που συνδυάζουν επιμελώς τις ιδέες που έχουμε δει μέχρι στιγμής - δηλαδή, την πολιτική κλίσης και μια μαθητευόμενη συνάρτηση αξίας. Σε αυτούς τους αλγόριθμους, μια πολιτική ενισχύεται με ένα μαθητευόμενο ενισχυτικό σήμα που παράγεται χρησιμοποιώντας μια μαθητευόμενη συνάρτηση αξίας. Αυτό έρχεται σε αντίθεση με την ΕΝΙΣΧΥΣΗ που χρησιμοποιεί μιας υψηλής διακύμανσης Monte Carlo εκτίμηση της επιστροφής για την ενίσχυση της πολιτικής.

Όλοι οι αλγόριθμοι Πράκτορα-Κριτή έχουν δύο συνιστώσες που εκπαιδεύονται από κοινού - ένας πράκτορας , που μαθαίνει μια παραμετροποιημένη πολιτική και έναν κριτή που μαθαίνει μια συνάρτηση αξίας για να αξιολογήσει τα ζεύγη κατάστασης-δράσης. Ο κριτής δίνει ένα ενισχυτικό σήμα στον πράκτορα.

Το κύριο κίνητρο πίσω από αυτούς τους αλγόριθμους είναι ότι ένα μαθητευόμενο ενισχυτικό σήμα μπορεί να είναι πιο ενημερωτικό για μια πολιτική από τις ανταμοιβές που διατίθενται από ένα περιβάλλον. Για παράδειγμα, μπορεί να μετατρέψει μια αραιή ανταμοιβή στην οποία ο πράκτορας λαμβάνει μόνο +1 σε περίπτωση επιτυχίας σε ένα πυκνό ενισχυτικό σήμα. Επιπλέον, οι μαθητευόμενες συναρτήσεις αξίας συνήθως έχουν χαμηλότερη διακύμανση από τις εκτιμήσεις του Monte Carlo για την απόδοση. Αυτό μειώνει την αβεβαιότητα σύμφωνα με την οποία μια πολιτική μαθαίνει [28], διευκολύνοντας τη μαθησιακή διαδικασία. Ωστόσο, η εκπαίδευση γίνεται επίσης πιο περίπλοκη. Τώρα η μάθηση της πολιτικής εξαρτάται από την ποιότητα της εκτίμησης της συνάρτησης αξίας που εκπαιδεύεται ταυτόχρονα. Μέχρι η συνάρτηση αξίας να δημιουργεί εύλογα μηνύματα για την πολιτική, η μάθηση πώς να επιλέγονται καλές ενέργειες θα είναι πρόκληση.

Είναι σύνηθες να μαθαίνουμε τη συνάρτηση πλεονεκτήματος $A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s)$ ως τα ενισχυτικά σήματα σε αυτές τις μεθόδους. Η βασική ιδέα είναι ότι είναι καλύτερο να επιλέξουμε μια ενέργεια με βάση την απόδοσή της σε σχέση με τις άλλες ενέργειες που είναι διαθέσιμες σε μια συγκεκριμένη κατάσταση, αντί της χρήσης της

απόλυτης τιμής αυτής της ενέργειας όπως μετράται από τη συνάρτηση Q. Το πλεονέκτημα ποσοτικοποιεί πόσο καλύτερη ή χειρότερη είναι μια ενέργεια από τη μέση διαθέσιμη ενέργεια. Οι αλγόριθμοι πράκτορα-κριτή που μαθαίνουν τη συνάρτηση πλεονεκτήματος είναι γνωστοί ως πλεονεκτικοί αλγόριθμοι πράκτορα-κριτή (A2C).

Οι πράκτορες μαθαίνουν παραμετροποιημένες πολιτικές π_θ χρησιμοποιώντας την πολιτική κλίσης όπως φαίνεται στην Εξίσωση 4.1. Αυτό είναι πολύ παρόμοιο με το REINFORCE εκτός από το ότι τώρα χρησιμοποιούμε το πλεονέκτημα A_t^π όπως ένα ενισχυτικό σήμα αντί για μια εκτίμηση του Monte Carlo για την επιστροφή $R_t(\tau)$.

Actor-Critic: $$\nabla_\theta J(\pi_\theta) = \mathbb{E}_t[A_t^\pi \nabla_\theta \log \pi_\theta(a_t|s_t)] \quad (4.1)$$

REINFORCE: $$\nabla_\theta J(\pi_\theta) = \mathbb{E}_t[R_t(\tau) \nabla_\theta \log \pi_\theta(a_t|s_t)] \quad (4.2)$$

Οι κριτές είναι υπεύθυνοι για να μάθουν πώς να αξιολογούν τα (s, a) ζεύγη και να το χρησιμοποιούν για να δημιουργήσουν το A^π.

Στη συνέχεια, περιγράφουμε πρώτα τη συνάρτηση πλεονεκτήματος και γιατί είναι μια καλή επιλογή για ένα ενισχυτικό σήμα. Στη συνέχεια, παρουσιάζουμε δύο μεθόδους για την εκτίμηση της συνάρτησης πλεονεκτήματος—επιστροφές n-βημάτων και γενικευμένη εκτίμηση πλεονεκτήματος [29].

4.2.1 Η συνάρτηση πλεονεκτήματος (advantage function)

Διαισθητικά, η συνάρτηση πλεονεκτήματος $A^\pi(s_t, a_t)$ μετρά το βαθμό στον οποίο μια ενέργεια είναι καλύτερη ή χειρότερη από τη μέση δράση της πολιτικής σε μια συγκεκριμένη κατάσταση. Το πλεονέκτημα ορίζεται στην εξίσωση 4.3.

$$A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t) \quad (4.3)$$

Έχει μια σειρά από ενδιαφέρουσες ιδιότητες. Πρώτον, $\mathbb{E}_{aEA}[A^\pi(s_t, a)] = 0$. Αυτό σημαίνει ότι αν όλες οι ενέργειες είναι ουσιαστικά ισοδύναμες, τότε το A^π θα είναι 0 για όλες τις ενέργειες και η πιθανότητα ανάληψης αυτών των ενεργειών θα παραμείνει αμετάβλητη όταν η πολιτική εκπαιδευτεί χρησιμοποιώντας το A^π. Συγκρίνουμε αυτό με ένα ενισχυτικό σήμα που βασίζεται σε απόλυτες τιμές κατάστασης ή κατάστασης-δράσης. Αυτό το σήμα θα έχει σταθερή τιμή στην ίδια κατάσταση, αλλά μπορεί να μην είναι

0. Κατά συνέπεια, θα ενθαρρύνει ενεργά (εάν είναι θετική) ή θα αποθαρρύνει (εάν είναι αρνητική) την ενέργεια που λαμβάνονται. Δεδομένου ότι όλες οι δράσεις ήταν ισοδύναμες, αυτό μπορεί να μην είναι προβληματικό στην πράξη, αν και είναι μη ενστικτώδες.

Ένα πιο προβληματικό παράδειγμα είναι εάν η δράση που αναλήφθηκε ήταν χειρότερη από τη μέση δράση, αλλά η αναμενόμενη απόδοση εξακολουθεί να είναι θετική. Δηλαδή, $Q^{\pi}(s_t, a) > 0$, αλλά $A^{\pi}(s_t, a) < 0$. Στην ιδανική περίπτωση, τα μέτρα που λαμβάνονται θα πρέπει να καθίστανται λιγότερο πιθανά, δεδομένου ότι υπήρχαν καλύτερες διαθέσιμες επιλογές. Σε αυτή την περίπτωση η χρήση του A^{π} αποδίδει συμπεριφορά που ταιριάζει περισσότερο με τη διαίσθησή μας, καθώς θα αποθαρρύνει τα μέτρα που λαμβάνονται. Η χρήση της Q^{π}ή ακόμα και της Q^{π} με μια γραμμή βάσης, μπορεί να ενθαρρύνει τη δράση.

Το πλεονέκτημα είναι επίσης ένα σχετικό μέτρο. Για μια συγκεκριμένη κατάσταση s και δράση a, εξετάζει την αξία του ζεύγους κατάστασης-δράσης, $Q^{\pi}(s_t, a)$, και αξιολογεί εάν το a θα οδηγήσει την πολιτική σε καλύτερη ή χειρότερη θέση, μετρούμενη σε σχέση με το $V^{\pi}(s)$. Το πλεονέκτημα αποφεύγει να τιμωρεί μια ενέργεια για την πολιτική που πρόσφατα βρίσκεται σε μια ιδιαίτερα κακή κατάσταση. Αντιστρόφως, δεν αποδίδει τα εύσημα σε μια ενέργεια για το γεγονός ότι η πολιτική είναι σε καλή κατάσταση. Αυτό είναι επωφελές επειδή το a μπορεί να επηρεάσει μόνο τη μελλοντική πορεία, αλλά όχι τον τρόπο με τον οποίο μια πολιτική έφτασε στη τρέχουσα κατάσταση. Θα πρέπει να αξιολογήσουμε τη δράση με βάση το πώς αλλάζει την αξία στο μέλλον.

Ας δούμε ένα παράδειγμα. Στην εξίσωση 4.4, η πολιτική είναι σε καλή κατάσταση με $V^{\pi}(s) = 100$, ενώ στην εξίσωση 4.5, είναι σε κακή κατάσταση με $V^{\pi}(s)$=-100. Και στις δύο περιπτώσεις, η δράση a αποδίδει μια σχετική βελτίωση του 10, η οποία λαμβάνεται από κάθε περίπτωση που έχει το ίδιο πλεονέκτημα. Ωστόσο, αυτό δεν θα ήταν σαφές αν εξετάζαμε μόνο το $Q^{\pi}(s, a)$.

$$Q^{\pi}(s, a) = 110, \quad V^{\pi}(s) = 100, \quad A^{\pi}(s, a) = 10 \qquad (4.4)$$

$$Q^{\pi}(s, a) = -90, \quad V^{\pi}(s) = -100, \quad A^{\pi}(s, a) = 10 \qquad (4.5)$$

Καταλαβαίνοντας αυτόν τον τρόπο, η συνάρτηση πλεονεκτήματος είναι σε θέση να συλλάβει τις μακροπρόθεσμες επιπτώσεις μιας ενέργειας, επειδή εξετάζει όλα τα μελλοντικά χρονικά βήματα, αγνοώντας τις επιπτώσεις όλων των δράσεων μέχρι σήμερα.

Οι Schulman et al. παρουσιάζουν μια παρόμοια ερμηνεία στην εργασία τους "Γενικευμένη Εκτίμηση Πλεονεκτήματος» [29].

Έχοντας δει γιατί η συνάρτηση πλεονεκτήματος $A^{\pi}(s,a)$ είναι μια καλή επιλογή ενισχυτικού σήματος για χρήση σε έναν αλγόριθμο Actor-Critic, ας δούμε δύο τρόπους εκτίμησής του.

4.2.1.1 Εκτίμηση πλεονεκτήματος: n-βημάτων επιστροφές(returns)

Για να υπολογίσουμε το πλεονέκτημα A^{π}, χρειαζόμαστε μια εκτίμηση για την μάθηση Q^{π} και την μάθηση V^{π}. Μια ιδέα είναι ότι εμείς θα μπορούσαμε να μάθουμε τις Q^{π}και V^{π} ξεχωριστά με διαφορετικά νευρωνικά δίκτυα. Ωστόσο, αυτό έχει δύο μειονεκτήματα. Πρώτον, πρέπει να ληφθεί μέριμνα ώστε οι δύο εκτιμήσεις να είναι συνεπείς. Δεύτερον, είναι λιγότερο αποτελεσματικό στην εκπαίδευση. Αντ 'αυτού, συνήθως εκπαιδεύουμε μόνο τη V^{π}και το συνδυάζουμε με ανταμοιβές από μια τροχιά για την εκτίμηση της Q^{π}.

Η μάθηση V^{π} προτιμάται από την μάθηση Q^{π} για δύο λόγους. Πρώτον, η Q^{π} είναι πιο περίπλοκη συνάρτηση και μπορεί να χρειαστεί περισσότερα δείγματα για να μάθει μια καλή εκτίμηση. Αυτό μπορεί να είναι ιδιαίτερα προβληματικό σε περίπτωση όπου ο πράκτορας και ο κριτής εκπαιδεύονται από κοινού. Δεύτερον, μπορεί να είναι πιο δαπανηρή υπολογιστικά για την εκτίμηση της V^{π}από της Q^{π}.Εκτιμώντας την $V^{\pi}(s)$ από $Q^{\pi}(s,a)$ απαιτεί τον υπολογισμό των τιμών για όλες τις πιθανές ενέργειες στην κατάσταση s και, στη συνέχεια, τη λήψη του σταθμισμένου μέσου όρου δράσης-πιθανότητας για την απόκτηση της $V^{\pi}(s)$.Επιπλέον, αυτό είναι δύσκολο για περιβάλλοντα με συνεχείς ενέργειες, δεδομένου ότι η εκτίμηση της V^{π} θα απαιτούσε ένα αντιπροσωπευτικό δείγμα ενεργειών από ένα συνεχές χώρο.

Ας δούμε πώς να εκτιμήσουμε την Q^{π} από την V^{π}.

Αν υποθέσουμε για μια στιγμή ότι έχουμε μια τέλεια εκτίμηση της $V^{\pi}(s)$, τότε η συνάρτηση Q μπορεί να ξαναγραφεί ως συνδυασμός των αναμενόμενων ανταμοιβών για n χρονικά βήματα, ακολουθούμενη από $V^{\pi}(s_{n+1})$ όπως φαίνεται στην εξίσωση 4.6. Για να γίνει αυτό επιτεύξιμο για εκτίμηση, χρησιμοποιούμε μια μόνο τροχιά ανταμοιβών $(r_1, \dots, r_n)$ στη θέση της προσδοκίας, και αντικαθιστούμε στη εκπαιδευμένη από τον κριτή $\hat{V}^{\pi}(s)$. Όπως εμφανίζεται στην εξίσωση 4.7, αυτό είναι γνωστό ως αποδόσεις n - step forward.

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{\tau\sim\pi}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^n r_{t+n}] + \gamma^{n+1} V^{\pi}(s_{t+n+1}) \qquad (4.6)$$

$$\approx r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^n r_{t+n} + \gamma^{n+1} \hat{V}^{\pi}(s_{t+n+1}) \qquad (4.7)$$

Η εξίσωση 4.7 καθιστά σαφή την αντιστάθμιση μεταξύ συστηματικού σφάλματος και διακύμανσης του εκτιμητή. Τα n βήματα των πραγματικών ανταμοιβών είναι χωρίς συστηματικό σφάλμα, αλλά έχουν υψηλή διακύμανση, καθώς προέρχονται από μόνο μία τροχιά. Η $\hat{V}^{\pi}(s)$ έχει χαμηλότερη διακύμανση αφού αντικατοπτρίζει μια προσδοκία για όλες τις τροχιές που έχουν παρατηρηθεί μέχρι στιγμής, αλλά έχει συστηματικό σφάλμα επειδή υπολογίζεται χρησιμοποιώντας μια συνάρτηση προσέγγισης. Η διαίσθηση πίσω από την ανάμειξη αυτών των δύο τύπων εκτιμήσεων είναι ότι η διακύμανση των πραγματικών ανταμοιβών συνήθως αυξάνεται όσο περισσότερα βήματα μακριά από το t γίνονται. Κοντά στο t, τα οφέλη από τη χρήση μιας εκτίμησης χωρίς συστηματικό σφάλμα μπορεί να υπερτερούν της διακύμανσης. Καθώς το n αυξάνεται, η διακύμανση στις εκτιμήσεις πιθανότατα θα αρχίσει να γίνεται προβληματική και η μετάβαση σε χαμηλότερη διακύμανση αλλά με συστηματικό σφάλμα εκτίμηση είναι καλύτερη. Ο αριθμός των βημάτων των πραγματικών ανταμοιβών, n, ελέγχει την αντιστάθμιση μεταξύ των δύο.

Συνδυάζοντας την εκτίμηση n -βημάτων για Q^{π} με $\hat{V}^{\pi}(s_t)$,παίρνουμε έναν τύπο για την εκτίμηση της συνάρτησης πλεονεκτήματος, που παρουσιάζεται στην εξίσωση 4.8:

$$A^{\pi}_{NSTEP} = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t)$$
$$\approx r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^n r_{t+n} + \gamma^{n+1} \hat{V}^{\pi}(s_{t+n+1}) - \hat{V}^{\pi}(s_t) \qquad (4.8)$$

Ο αριθμός των βημάτων των πραγματικών ανταμοιβών, n, ελέγχει το ποσό της διακύμανσης στον εκτιμητή πλεονεκτημάτων, και είναι μια πολύ σημαντική παράμετρος που πρέπει να εκτιμηθεί. Το μικρό n έχει ως αποτέλεσμα έναν εκτιμητή με χαμηλότερη διακύμανση αλλά υψηλότερο συστηματικό σφάλμα, μεγάλο n οδηγεί σε εκτιμητή με υψηλότερη διακύμανση αλλά χαμηλότερο συστηματικό σφάλμα.

4.2.1.2 Εκτίμηση πλεονεκτήματος: Γενικευμένη εκτίμηση πλεονεκτήματος (GAE)

Η γενικευμένη εκτίμηση πλεονεκτήματος (Generalized Advantage Estimate-GAE) [29] προτάθηκε από τους Schulman et al. ως βελτίωση σε σχέση με την εκτίμηση αποδόσεων n -step για τη συνάρτηση πλεονεκτήματος. Αντιμετωπίζει το πρόβλημα της ρητής επιλογής του αριθμού των βημάτων των επιστροφών, n. Η κύρια ιδέα πίσω από το GAE είναι ότι αντί να επιλέξουμε μία τιμή του n, αναμιγνύουμε πολλαπλές τιμές του n. Έτσι , υπολογίζουμε το πλεονέκτημα χρησιμοποιώντας έναν σταθμισμένο μέσο όρο μεμονωμένων πλεονεκτημάτων που υπολογίζεται με $n = 1,2,3, \dots, k$. Ο σκοπός της GAE είναι να μειώσει σημαντικά τη διακύμανση του εκτιμητή διατηρώντας παράλληλα το συστηματικό σφάλμα που εισήχθη όσο το δυνατόν χαμηλότερα.

Το GAE ορίζεται ως ένας εκθετικά σταθμισμένος μέσος όρος όλων των αποδόσεων n-step forward πλεονεκτημάτων. Παρουσιάζεται στην εξίσωση 4.9

$$A^{\pi}_{GAE} = \sum_{l=0}^{\infty} (\gamma\lambda)^{l} \delta_{t+1} \qquad (4.9)$$

Όπου $\delta_t = r_t + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t)$

Διαισθητικά, η GAE λαμβάνει έναν σταθμισμένο μέσο όρο ορισμένων εκτιμητών πλεονεκτημάτων με διαφορετικό συστηματικό σφάλμα και διακύμανση. Το GAE σταθμίζει περισσότερο το πλεονέκτημα υψηλού συστηματικού σφάλματος και χαμηλής διακύμανσης ενός βήματος, αλλά περιλαμβάνει επίσης συνεισφορές από εκτιμητές χαμηλότερου συστηματικού σφάλματος και υψηλότερης διακύμανσης χρησιμοποιώντας $2,3, \dots, n$ βήματα. Η συνεισφορά μειώνεται με εκθετικό ρυθμό όσο ο αριθμός των βημάτων αυξάνεται. Ο ρυθμός πτώσης ελέγχεται από τον συντελεστή λ. Επομένως, όσο μεγαλύτερο λ, το υψηλότερη διακύμανση.

Τόσο το GAE όσο και οι εκτιμήσεις της συνάρτησης πλεονεκτήματος n-βημάτων περιλαμβάνουν τον συντελεστή προεξόφλησης γ ,ο οποίος ελέγχει το κατά πόσο ένας αλγόριθμος «νοιάζεται» για τις μελλοντικές ανταμοιβές σε σύγκριση με τη τρέχουσα ανταμοιβή. Επιπλέον, και οι δύο έχουν μια παράμετρο που ελέγχει την αντιστάθμιση διακύμανσης και συστηματικού σφάλματος: n για τη συνάρτηση πλεονεκτήματος και λ για GAE. Τι κερδίσαμε λοιπόν με το GAE;

Παρόλο που το n και το λ ελέγχουν και τα δύο την αντιστάθμιση συστηματικού σφάλματος-διακύμανσης, το κάνουν με διαφορετικούς τρόπους. Το n αντιπροσωπεύει μια δύσκολη επιλογή, καθώς καθορίζει με ακρίβεια το σημείο στο οποίο οι ανταμοιβές υψηλής διακύμανσης αλλάζουν για την εκτίμηση της συνάρτησης V. Αντίθετα, το λ αντιπροσωπεύει μια ήπια επιλογή: μικρότερες τιμές του λ θα σταθμίσουν περισσότερο την εκτίμηση της συνάρτησης V, ενώ οι μεγαλύτερες τιμές θα σταθμίσουν περισσότερο τις πραγματικές ανταμοιβές. Ωστόσο, εκτός εάν $\lambda = 0$ ή $\lambda = 1$,χρησιμοποιώντας το λ εξακολουθεί να επιτρέπει σε υψηλότερες ή χαμηλότερες εκτιμήσεις διακύμανσης να συνεισφέρουν ,εξ ου και η ήπια επιλογή.

4.2.2 Εκπαιδεύοντας τη συνάρτηση πλεονεκτήματος(advantage function)

Έχουμε δει δύο τρόπους για να εκτιμήσουμε τη συνάρτηση πλεονεκτήματος. Και οι δύο αυτές μέθοδοι υποθέτουν ότι έχουν πρόσβαση σε μια εκτίμηση για την V^{π}, όπως φαίνεται παρακάτω.

$$A^{\pi}_{NSTEP} \approx r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^n r_{t+n} + \gamma^{n+1}\hat{V}^{\pi}(s_{t+n+1}) - \hat{V}^{\pi}(s_t)$$

$$A^{\pi}_{GAE} = \sum_{l=0}^{\infty}(\gamma\lambda)^l \delta_{t+1}$$

Όπου $\delta_t = r_t + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t)$

Εκπαιδεύουμε την V^{π} χρησιμοποιώντας τη μάθηση Temporal Difference (TD) με τον ίδιο τρόπο που τη χρησιμοποιήσαμε για να εκπαιδεύσουμε την Q^{π}για DQN. Συνοπτικά, ακολουθεί η διαδικασία εκπαίδευσης ως εξής. Παραμετροποιούμε το V^{π} με θ, δημιουργούμε V^{π}_{tar} για κάθε μία από τις εμπειρίες που συγκεντρώνει ένας πράκτορας και ελαχιστοποιούμε τη διαφορά μεταξύ $\widehat{V^{\pi}}(s;\theta)$ και V^{π}_{tar} χρησιμοποιώντας την απώλεια παλινδρόμησης, όπως στο μέσο τετραγωνικό σφάλμα(Mean Squared Error-MSE). Επαναλαμβάνουμε αυτήν τη διαδικασία για πολλά βήματα.

Η V^{π}_{tar} μπορεί να παραχθεί χρησιμοποιώντας οποιαδήποτε κατάλληλη εκτίμηση. Η απλούστερη μέθοδος είναι να ορίσουμε $V^{\pi}_{tar}(s) = r + \widehat{V^{\pi}}(s';\theta)$.Αυτό φυσικά γενικεύεται σε μια εκτίμηση n-βημάτων, όπως φαίνεται στο εξίσωση 4.10

$$V^{\pi}_{tar}(s_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^n r_{t+n} + \gamma^{n+1}\hat{V}^{\pi}(s_{t+n+1}) \qquad (4.10)$$

Εναλλακτικά, μπορούμε να χρησιμοποιήσουμε μια εκτίμηση Monte Carlo για V_{tar}^{π} που φαίνεται στην εξίσωση 4.11.

$$V_{tar}^{\pi}(s_t) = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'} \qquad (4.11)$$

Ή απλά μπορούμε να θέσουμε:

$$V_{tar}^{\pi}(s_t) = A_{GAE}^{\pi}(s_t, a_t) + \hat{V}^{\pi}(s_t) \qquad (4.12)$$

Πρακτικά, για να αποφευχθεί ο πρόσθετος υπολογισμός, η επιλογή του V_{tar}^{π} σχετίζεται συχνά με την μέθοδο που χρησιμοποιείται για την εκτίμηση του πλεονεκτήματος. Για παράδειγμα, μπορούμε να χρησιμοποιήσουμε την εξίσωση 4.10 στην εκτίμηση πλεονεκτημάτων με τη χρήση αποδόσεων n-βημάτων ή την εξίσωση 4.12 κατά την εκτίμηση πλεονεκτημάτων χρησιμοποιώντας GAE.

Είναι επίσης δυνατό να χρησιμοποιήσετε μια πιο προηγμένη διαδικασία βελτιστοποίησης κατά την μάθηση $\hat{V}^{\pi}$. Για παράδειγμα, στο έγγραφο GAE [29],η $\hat{V}^{\pi}$μαθαίνεται χρησιμοποιώντας μια μέθοδο περιοχής εμπιστοσύνης.

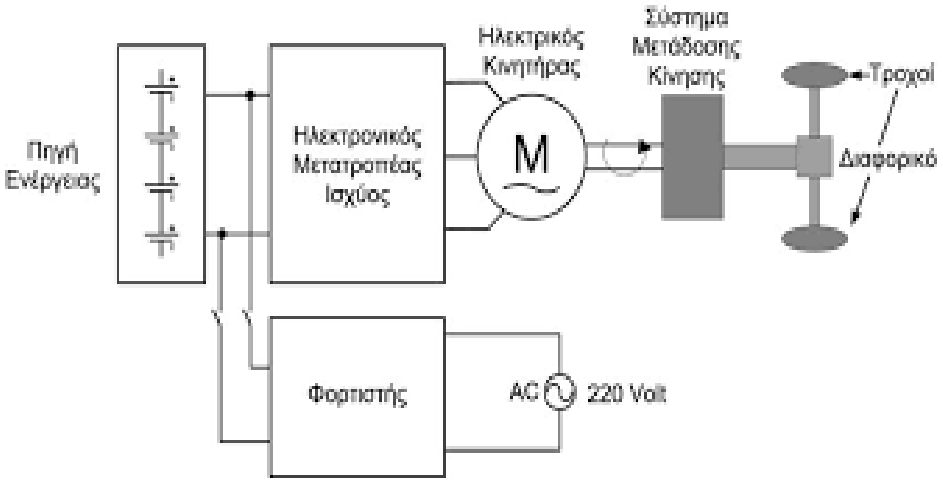
ΚΕΦΑΛΑΙΟ 5ο

Μοντελοποίηση Κινητήρα και PI Έλεγχος Συστήματος

Στην συγκεκριμένη ενότητα παρουσιάζεται το ολοκληρωμένο σύστημα του ηλεκτρικού αυτοκινήτου με το οποίο ασχολούμαστε καθώς και τα επιμέρους χαρακτηριστικά του. Αρχικά θα αναφερθούμε στη δομή του συστήματός μας .Στη συνέχεια γίνεται αναφορά στις εξισώσεις που διέπουν τόσο τον κινητήρα όσο και τον dc-ac μετατροπέα και έπειτα θα παρατεθούν οι τιμές των χαρακτηριστικών της μηχανής.

Το σύστημά μας αποτελείται από μια πηγή ρεύματος ,η οποία τροφοδοτεί τον κινητήρα του αυτοκινήτου ,στο μεσοδιάστημα των οποίων περιέχεται ένας dc-ac μετατροπέας ,ο οποίος είναι απαραίτητος για την σωστή τροφοδότηση του ηλεκτρικού σύγχρονου κινητήρα .Το όλο σύστημά μας υπόκειται σε έναν έλεγχο ,ο οποίος χωρίζεται σε δύο μέρη :έναν εξωτερικό έλεγχο ,ο οποίος αφορά τις στροφές του κινητήρα και έναν εσωτερικό ,ο οποίος αφορά τα επιμέρους ρεύματα του κινητήρα .Ο έλεγχος γίνεται με τη χρησιμοποίηση PI ελεγκτών ,τα κέρδη των οποίων θα μας εξασφαλίσουν την άρτια και σωστή ,σύμφωνα πάντα με τις προδιαγραφές ,λειτουργία του συστήματός μας .Είναι χρήσιμο να αναφερθεί ότι, πέραν της χρησιμοποίησης PI ελεγκτών , έγινε και η απαραίτητη γραμμικοποίηση στη διαδικασία του ελέγχου στις εξισώσεις εισόδου στους ελεγκτές. Με αυτόν τον τρόπο αποφεύγουμε την διερεύνηση εμπειρικών κερδών ,οι οποίοι θα μας εξυπηρετούσαν ,και χρησιμοποιούμε κέρδη τα οποία προέρχονται από μαθηματική επίλυση του θέματος.

Στο παρακάτω σχήμα παρουσιάζεται η χονδρική δομή του συστήματος ενός ηλεκτρικού αυτοκινήτου ,το οποίο αποτελείται από μια πηγή ενέργειας (ή μονάδα αποθήκευσης ενέργειας με φορτιστή), έναν ηλεκτρονικό μετατροπέα ισχύος ,έναν ηλεκτρικό κινητήρα ,ένα σύστημα μετάδοσης κίνησης και τους τροχούς του οχήματος[30].Στην περίπτωσή μας σαν πηγή ενέργειας χρησιμοποιούμε μια πηγή ρεύματος.



Σχήμα 5.1. Χονδρικό διάγραμμα ηλεκτρικού οχήματος [30]

•Μοντελοποίηση

Οι εξισώσεις της σύγχρονης μηχανής με μόνιμο μαγνήτη στο τριφασικό σύστημα είναι οι εξής:

$$\begin{bmatrix} U_{as} \\ U_{bs} \\ U_{cs} \end{bmatrix} = R_s \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_{as} \\ \lambda_{bs} \\ \lambda_{cs} \end{bmatrix} \quad (5.1)$$

όπου:

U_{as} , U_{bs} , U_{cs} η στιγμιαία τιμή της τάσεως στους ακροδέκτες του στάτη σε κάθε φάση.

i_{as} , i_{bs} , i_{cs} η στιγμιαία τιμή του ρεύματος του στάτη σε κάθε φάση.

λ_{as} , λ_{bs} , λ_{cs} η στιγμιαία τιμή της ροής του στάτη σε κάθε φάση.

R_s η αντίσταση των τυλιγμάτων του στάτη.

Με τον μετασχηματισμό Park προκύπτουν οι εξισώσεις:

$$\begin{bmatrix} V_{ds} \\ V_{qs} \end{bmatrix} = R_s \begin{bmatrix} I_{ds} \\ I_{qs} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} L_{ds} \\ L_{qs} \end{bmatrix} + \omega_s \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} I_{ds} \\ I_{qs} \end{bmatrix}$$

(5.2)

όπου:

V_{ds} , V_{qs} *η τάση d,q στο στάτη.*

I_{ds} , I_{qs} *το ρεύμα d, q στο στάτη.*

L_{ds} , L_{qs} *οι αυτεπαγωγές d, q στον άξονα του στάτη.*

R_s η αντίσταση των τυλιγμάτων του στάτη.

ω_s η γωνιακή συχνότητα του μετασχηματισμού.

Από τους λόγους κατάτμησης ισχύουν:

$V_{ds} = m_{ds}V_{dc}$ και $V_{qs} = m_{qs}V_{dc}$

Λαμβάνοντας υπόψη τις παραπάνω σχέσεις, οι εξισώσεις της μηχανής γράφονται ως εξής:

$$LI_{ds}^{\cdot} = -r_s I_{ds} + p\omega_r I_{qs} L_{qs} + m_{ds}V_{dc}$$

(5.3)

$$L_{qs}I_{qs}^{\cdot} = -r_s I_{qs} - p\omega_r I_{ds} L_{ds} - p\omega_r \psi_m + m_{qs}V_{dc}$$

(5.4)

Οι δύο αυτές εξισώσεις θα χρησιμοποιηθούν για τη μοντελοποίηση του συστήματος.
Η διαφορική εξίσωση που διέπει τη γωνιακή ταχύτητα περιστροφής του άξονα του συστήματος διάδοσης δίνεται από τον τύπο:

$$J\dot{\omega}_r = -\beta\omega_r + \frac{3}{2}p\left(L_{ds} - L_{qs}\right)I_{ds}I_{qs} + \frac{3}{2}p\psi_m I_{qs} - T_m$$

(5.5)

Η ηλεκτρομαγνητική ροπή τ_e της μηχανής δίνεται από την εξίσωση:

$$\tau_e = \frac{3}{2}p\left(L_{ds} - L_{qs}\right)I_{ds}I_{qs} + \frac{3}{2}p\psi_m I_{qs}$$

(5.6)

Μεταξύ των δυο μετατροπέων του κάτω κλάδου, αναπτύσσεται τάση V_{dc}, η οποία εκφράζεται με την ακόλουθη διαφορική εξίσωση:

$$CV_{dc}^{\cdot} = -\frac{V_{dc}}{R_{dc}} - \frac{3}{2}\left(m_{ds}I_{ds} + m_{qs}I_{qs}\right) + I_S$$

(5.7)

Όπου I_S η πηγή ρεύματος του συστήματός μας.

•Διαδικασία ελέγχου

Τώρα ερχόμαστε στη διαδικασία ελέγχου του κινητήρα μας.

Στις προηγούμενες σχέσεις έχει υιοθετηθεί προσανατολισμός στη ροή του μόνιμου μαγνήτη για τις εξισώσεις της μηχανής.Εδώ παρουσιάζεται ο εν σειρά έλεγχος του μετατροπέα από την πλευρά της μηχανής (machine-side converter , MSC). Η ανάλυση θα γίνει για την περίπτωση PMSM με κυλινδρικό δρομέα ($L_{ds} = L_{qs} = L_S$)

Ξεκινώντας από τον εσωτερικό βρόχο ελέγχου ρευμάτων, εφαρμόζεται ο μετασχηματισμός εισόδου από V σε u:

$$u_{ds} = V_{ds} + p\omega_r L_{qs} I_{qs} \Rightarrow V_{ds} = u_{ds} - p\omega_r L_{qs} I_{qs}$$

$$u_{qs} = V_{qs} - p\omega_r L_{ds} I_{ds} - p\omega_r \psi_m \Rightarrow V_{qs} = u_{qs} + p\omega_r L_{ds} I_{ds} + p\omega_r \psi_m$$

Αποζευγμένες πρωτοβάθμιες διαφορικές εξισώσεις:

$$\begin{matrix} L\dot{I_{ds}} + r_s I_{ds} = u_{ds} \\ L_{qs}\dot{I_{qs}} + r_s I_{ds} = u_{qs} \end{matrix} \text{ και επιλέγονται } \begin{matrix} u_{ds} = K_{P_{ds}}(I_{ds}^{ref} - I_{ds}) + K_{I_{ds}} \int_0^t (I_{ds}^{ref} - I_{ds}) d\tau \\ u_{qs} = K_{P_{qs}}(I_{qs}^{ref} - I_{qs}) + K_{I_{qs}} \int_0^t (I_{qs}^{ref} - I_{qs}) d\tau \end{matrix}$$

Οι πραγματικές είσοδοι ελέγχου είναι:

$$V_{ds} = -p\omega_r L_{qs} I_{qs} + K_{P_{ds}}(I_{ds}^{ref} - I_{ds}) + K_{I_{ds}} \int_0^t (I_{ds}^{ref} - I_{ds}) d\tau$$

$$V_{qs} = p\omega_r L_{ds} I_{ds} + p\omega_r \psi_m + K_{P_{qs}}(I_{qs}^{ref} - I_{qs}) + K_{I_{qs}} \int_0^t (I_{qs}^{ref} - I_{qs}) d\tau$$

Συνάρτηση μεταφοράς κλειστού βρόχου (i=d,q):

$$\frac{I_{is}}{I_{is}^{ref}} = \frac{K_{P_{is}} s + K_{I_{is}}}{L_s s^2 + (r_s + K_{P_{is}}) s + K_{I_{is}}} = \frac{1}{1 + \tau_{is} s}$$

με $K_{I_{is}} = \frac{r_s}{\tau_{is}}$ και $K_{P_{is}} = \frac{L_s}{\tau_{is}}$ όπου τ_{is} είναι η επιθυμητή σταθερά χρόνου.

Εδώ ολοκληρώνεται η σχεδίαση των εσωτερικών βρόχων ελέγχου για τον MSC. Στο συγκεκριμένο σημείο θα ασχοληθούμε με τον εξωτερικό έλεγχο στροφών.

Ξεκινώντας από τη ρύθμιση γωνιακής ταχύτητας δρομέα, ανακαλούμε τον τύπο της ηλεκτρομαγνητικής ροπής και τον υιοθετημένο προσανατολισμό στη ροή στάτη, οπότε έχουμε $\tau_e = \frac{3}{2} p\psi_m I_{qs}$.
Έτσι, κάνοντας χρήση και της θεώρησης ξεχωριστών σταθερών χρόνου σε εν σειρά βρόχου ελέγχου($I_{qs} = I_{qs}^{ref}$) τελικά ισχύει η ακόλουθη σχέση για την ηλεκτρομαγνητική ροπή $\tau_e = \frac{3}{2} p\psi_m I_{qs}^{ref}$.

Σε αυτό το σημείο θα γίνει σαφές ο τρόπος με τον οποίο θα εξαχθούν τα κέρδη του τροποποιημένου PI ελεγκτή γωνιακής ταχύτητας δρομέα $I_{qs}^{ref} = -K_{P_\omega}\omega_r + K_{I_\omega}\int_0^t(\omega_r^{ref} - \omega_r)$ κάνοντας χρήση της θεώρησης ξεχωριστών σταθερών χρόνου. Το άλλο ρεύμα συνηθίζεται να λαμβάνει αναφορά $I_{ds}^{ref} = 0$.

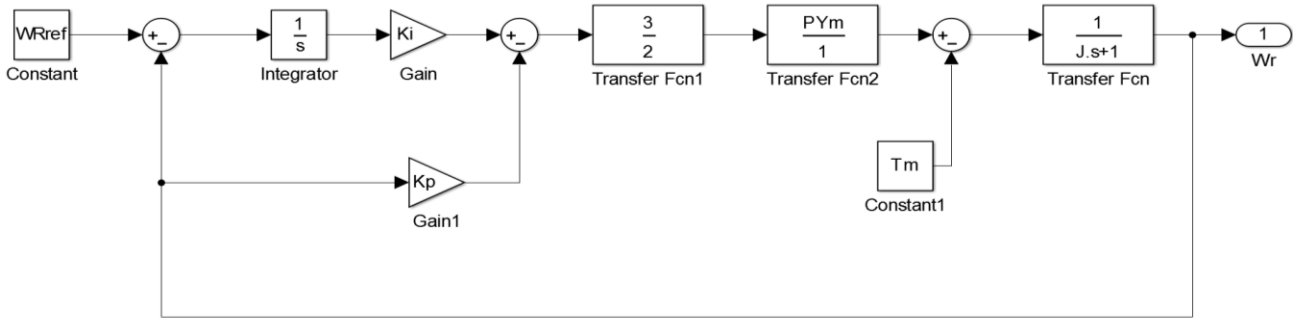
Η διαφορική εξίσωση γωνιακής ταχύτητας δρομέα γράφεται ως εξής:

$$J\dot{\omega}_r + b\omega_r = \frac{3}{2} p\psi_m I_{qs} - T_m$$

Κάνοντας χρήση του μετασχηματισμού Laplace προκύπτει οτι:

$$Js\omega_r + b\omega_r = \frac{3}{2} p\psi_m I_{qs} - T_m \Rightarrow \omega_r = \left[\frac{3}{2} p\psi_m I_{qs} - T_m\right]\frac{1}{Js + b}$$

Ανακαλώντας τη διαφορική εξίσωση ,προκύπτει το ακόλουθο σχήμα ,το οποίο απεικονίζει το σύστημά μας:



Σχήμα 5.2.Γραφική απεικόνιση συστήματος.

Από το παραπάνω σχήμα μπορούμε να βγάλουμε τη συνάρτηση μεταφοράς του κλειστού ,η οποία και είναι η παρακάτω:

$$H(s)=\frac{\frac{\frac{1}{s}K_I(\frac{3}{2}p\psi_m-T_m)(\frac{1}{Js+b})}{1+K_p(\frac{3}{2}p\psi_m-T_m)(\frac{1}{Js+b})}}{1+\frac{\frac{1}{s}K_I(\frac{3}{2}p\psi_m-T_m)(\frac{1}{Js+b})}{1+K_p(\frac{3}{2}p\psi_m-T_m)(\frac{1}{Js+b})}}\Rightarrow$$

$$H(s)=\frac{\frac{1}{s}K_I(\frac{3}{2}p\psi_m-T_m)(\frac{1}{Js+b})}{1+\frac{1}{s}K_I\left(\frac{3}{2}p\psi_m-T_m\right)\left(\frac{1}{Js+b}\right)+K_p(\frac{3}{2}p\psi_m-T_m)(\frac{1}{Js+b})}\Rightarrow$$

$$H(s)=\frac{\frac{K_I(\frac{3}{2}p\psi_m-T_m)}{(Js+b)s}}{\frac{(Js+b)+\frac{1}{s}K_I\left(\frac{3}{2}p\psi_m-T_m\right)+K_p(\frac{3}{2}p\psi_m-T_m)}{(Js+b)}}\Rightarrow$$

$$H(s)=\frac{K_I(\frac{3}{2}p\psi_m-T_m)}{s(Js+b)++sK_p\left(\frac{3}{2}p\psi_m-T_m\right)+K_I\left(\frac{3}{2}p\psi_m-T_m\right)}\Rightarrow$$

$$H(s)=\frac{K_I(\frac{3}{2}p\psi_m-T_m)}{Js^2+s\left[\left(\frac{3}{2}p\psi_m-T_m\right)K_p+b\right]+K_I\left(\frac{3}{2}p\psi_m-T_m\right)}\Rightarrow$$

$$H(s)=\frac{K_I(\frac{3}{2}p\psi_m-T_m)\frac{1}{J}}{s^2+\left[\frac{\left(\frac{3}{2}p\psi_m-T_m\right)K_p+b}{J}\right]s+\frac{K_I\left(\frac{3}{2}p\psi_m-T_m\right)}{J}}$$

Επιπλέον, θεωρώντας ως διαταραχή τη μηχανική ροπή και αγνοώντας τη, προκύπτει συνάρτηση μεταφοράς της μορφής: $\frac{\omega_r}{\omega_r^{ref}} = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$

Αρα προκύπτει οτι:$\omega_n^2 = K_I(\frac{3}{2}p\psi_m - T_m)\frac{1}{J}$ και επίσης $2\zeta\omega_n = \frac{\left(\frac{3}{2}p\psi_m - T_m\right)K_p + b}{J}$

Για την επιλογή των κερδών χρησιμοποιείται το κριτήριο απόκλισης 2%, το οποίο αναφέρει πως μετά από χρόνο αποκατάστασης T_s η απόκριση αναμένεται να διαφέρει το πολύ κατά 2% από την τιμή ισορροπίας, αυτό συνδέει τον T_s με τα ω_n και ζ ως εξής: $T_s = \frac{4}{\zeta\omega_n}$

Εύκολα, με βάση τις σχέσεις της προηγούμενη διαφάνειας, τα κέρδη του τροποποιημένου PI ελεγκτή επιλέγονται ως:

$$K_{P_\omega} = \left[\frac{8}{T_s} - \frac{b}{J}\right]\left[\frac{2J}{3p\psi_m - T_m}\right] \quad \textit{και} \quad K_{I_\omega} = \frac{32J}{[3p\psi_m - T_m]T_s^2\zeta^2}$$

Στο συγκεκριμένο στάδιο γίνεται αναφορά στις τιμές των παραμέτρων του συστήματός μας. Για την σωστή παραμετροποίηση του συστήματος χρειάστηκε να οριστούν κάποιες συγκεκριμένες παράμετροι. Συγκεκριμένα:

- Οι παράμετροι της μηχανής:

$$L_{qs} = 1.73 * 10^{-5} H$$
$$L_{ds} = 1.73 * 10^{-5} H$$
$$L_s = L_{qs} = L_{ds} = 1.73 * 10^{-5} H$$
$$r_s = 5.582 * 10^{-3} \Omega$$
$$p = 9$$
$$\psi_m = 0.0175 Wb$$
$$r_{ds} = 1\Omega$$
$$b = 0.003852 Nm\ sec$$
$$J = 1.8\ kg\ m^2$$
$$\zeta = 0.707$$
$$I_S = 10\ A$$

- Για την διασύνδεση μεταξύ των μετατροπέων υπάρχει ο πυκνωτής:

$C = 0.01F$

- Τα κέρδη που χρησιμοποιήθηκαν είναι τα παρακάτω :

Έχει ληφθεί μια σταθερά χρόνου $\tau_\iota = 1 * 10^{-3} sec$ και αντίστοιχα η σταθερά χρόνου

$\tau_s = 2\ sec.$

Για τους εσωτερικούς βρόχους οι τιμές των κερδών είναι:

$K_P^{in} = \frac{L_s}{\tau_\iota} = 0.0173$ και $K_I^{in} = \frac{r_s}{\tau_\iota} = 5.5820.$

Ενώ αντίστοιχα για τους εξωτερικούς βρόχους οι τιμές των κερδών είναι:

$K_P^{out} = 30.4599$ και $K_I^{out} = 60.9708.$

ΚΕΦΑΛΑΙΟ 6ο

Εφαρμογή Ελεγκτή Ενισχυτικής Μάθησης

Στο συγκεκριμένο κεφάλαιο παρουσιάζεται η εφαρμογή ελεγκτών ενισχυτικής μάθησης στο σύστημά του κινητήρα και η επίδρασή του σε αυτό. Σε αρχικό στάδιο ,γίνεται αναφορά στον αλγόριθμο που χρησιμοποιήθηκε και εν συνεχεία παρατίθενται τα αποτελέσματά του ως προς την απόδοση του συστήματος ελέγχου.

Παρότι για το σύστημα, το οποίο καλούμαστε να ελέγξουμε, υπάρχει αναλυτικό δυναμικό μαθηματικό μοντέλο και ως εκ τούτου μπορούμε να βρούμε επακριβώς τα βέλτιστα δυνατά βάρη των ελεγκτών ,εισάγουμε στο σύστημα ελέγχου έναν ελεγκτή ενισχυτικής μάθησης προκειμένου να διαπιστώσουμε την επίδραση ενός μηχανισμού ελέγχου ενισχυτικής μάθησης σε τέτοιου είδους συστήματα αναφοράς.

Ο αλγόριθμος που χρησιμοποιήθηκε ως ελεγκτικός μηχανισμός σε συνδυασμό με τους PI ελεγκτές είναι ο Deep Deterministic Policy Gradient Agent-DDPG(βαθιά ντετερμινιστική πολιτική κλίσης). Ο λόγος της επιλογής του συγκεκριμένου αλγορίθμου είναι ότι παρουσιάζει καλύτερα αποτελέσματα σε σχέση με υπόλοιπους αλγόριθμούς που χρησιμοποιήθηκαν ,όπως ο Deep Q-Network(DQN).

Στο σημείο αυτό, γίνεται αναφορά στη θεωρητική μορφή του συγκεκριμένου αλγόριθμου και στον τρόπο με τον οποίο λειτουργεί. Ο αλγόριθμος βαθιάς ντετερμινιστικής πολιτικής κλίσης είναι ένας αλγόριθμος μεθόδου ενισχυτικής μάθησης model free, online, off policy. Ένας πράκτορας βαθιάς ντετερμινιστικής πολιτικής κλίσης είναι ένας πράκτορας μάθησης ενίσχυσης πράκτορα-κριτή που αναζητά μια βέλτιστη πολιτική που μεγιστοποιεί την αναμενόμενη σωρευτική μακροπρόθεσμη ανταμοιβή.[31]

Οι πράκτορες βαθιάς ντετερμινιστικής πολιτικής κλίσης μπορούν να εκπαιδευτούν σε περιβάλλοντα με τους ακόλουθους χώρους παρατήρησης και δράσης:

Πεδίο Παρατήρησης (Observation Space)	**Πεδίο δράσης (Action Space)**
Συνεχής ή διακριτό (Continuous or discrete)	Συνεχής (Continuous)

Οι πράκτορες του βαθιάς ντετερμινιστικής πολιτικής κλίσης χρησιμοποιούν τις ακόλουθες αναπαραστάσεις πρακτόρων και κριτών.

Κριτής-Critic	**Πράκτορας-Actor**
Συνάρτηση αξίας κριτή Q (Q value function critic) $Q(s,a)$	Ντετερμινιστική πολιτική πράκτορα (Deterministic policy actor) $\pi(s)$

Κατά τη διάρκεια της εκπαίδευσης, ένας πράκτορας βαθιάς ντετερμινιστικής πολιτικής κλίσης:

- Ενημερώνει τις ιδιότητες του πράκτορα και του κριτή σε κάθε βήμα κατά τη διάρκεια της εκπαίδευσης.
- Αποθηκεύει προηγούμενες εμπειρίες χρησιμοποιώντας ένα buffer κυκλικής εμπειρίας. Ο πράκτορας ενημερώνει τον πράκτορα και τον κριτή χρησιμοποιώντας μια μίνι παρτίδα εμπειριών που ελήφθησαν τυχαία από το buffer.
- Διαταράσσει τη δράση που επιλέγει η πολιτική χρησιμοποιώντας ένα μοντέλο στοχαστικού θορύβου σε κάθε βήμα της εκπαίδευσης.

Για την εκτίμηση της συνάρτησης πολιτικής και τιμής, ένας πράκτορας βαθιάς ντετερμινιστικής πολιτικής κλίσης διατηρεί τέσσερις προσεγγιστές συναρτήσεων:

- Πράκτορας $\pi(s,\theta)$- Ο πράκτορας, με παραμέτρους θ, παίρνει την παρατήρηση s και επιστρέφει την αντίστοιχη δράση που μεγιστοποιεί τη μακροπρόθεσμη ανταμοιβή.
- Στόχος-actor $\pi_t(s,\theta_t)$ — Για να βελτιωθεί η σταθερότητα της βελτιστοποίησης, ο πράκτορας ορίζει περιοδικά τις παραμέτρους στόχου του πράκτορα θ_t στις τελευταίες τιμές παραμέτρων πράκτορα.
- Κριτής $Q(s,a|\varphi)$- Ο κριτής, με παραμέτρους φ, παίρνει την παρατήρηση s και τη δράση s ως εισόδους και επιστρέφει την αντίστοιχη προσδοκία της μακροπρόθεσμης ανταμοιβής.
- Στόχος κριτή $Q_t(s,a|\varphi_t)$— Για να βελτιωθεί η σταθερότητα της βελτιστοποίησης, ο πράκτορας ρυθμίζει περιοδικά τις παραμέτρους κριτικής στόχου φ_t στις τελευταίες τιμές παραμέτρων των κριτών.

Τόσο η $Q(s,a)$ όσο και η $Q_t(s,a)$ έχουν την ίδια δομή και παραμετροποίηση, ενώ το ίδιο ισχύει τόσο για την $\pi(s)$ όσο και την $\pi_t(s)$. Όταν ολοκληρωθεί η εκπαίδευση, η εκπαιδευμένη βέλτιστη πολιτική αποθηκεύεται στον πράκτορα $\pi(s)$.

Εκπαίδευση αλγορίθμου

Οι πράκτορες DDPG χρησιμοποιούν τον ακόλουθο αλγόριθμο εκπαίδευσης, στον οποίο ενημερώνουν τα μοντέλα πρακτόρων και κριτών τους σε κάθε χρονικό βήμα.

- Αρχικοποιούμε τον κριτή Q(S,A) με τυχαίες τιμές παραμέτρων φ και αρχικοποιούμε τις παραμέτρους στόχου του κριτή φ_t με τις ίδιες τιμές: $\varphi_t = \varphi$.
- Αρχικοποιούμε τον πράκτορα $\pi(s)$ με τυχαίες τιμές παραμέτρων θ και αρχικοποιούμε τις παραμέτρους στόχου του πράκτορα θ_t με τις ίδιες τιμές: $\theta_t = \theta$.
- Για κάθε χρονικό βήμα εκπαίδευσης:
  - Για την τρέχουσα παρατήρηση s, επιλέγουμε τη δράση $a = \pi(s) + N$, όπου N είναι ο στοχαστικός θόρυβος από το μοντέλο θορύβου
  - Εκτελούμε τη δράση a. Παρατηρούμε την ανταμοιβή r και την επόμενη παρατήρηση s'.
  - Αποθηκεύουμε την εμπειρία (s,a,r,s') στο experience buffer.
  - Γίνεται δειγματοληψία μιας τυχαίας μίνι παρτίδας εμπειριών M (s_i,a_i,r_i,s_i') από το experience buffer.
  - Εάν το s_i' είναι μια τερματική κατάσταση, ορίζουμε το στόχο της συνάρτηση αξίας y_i σε r_i. Διαφορετικά, το ορίζουμε σε:

    $$y_i = r_i + \gamma Q_t(s_i', \pi_t(s_i|\theta_t)|\varphi_t)$$

    Ο στόχος της συνάρτησης αξίας είναι το εμπειρικό άθροισμα της ανταμοιβής r_i και της εκπτωτικής μελλοντικής ανταμοιβής. Για να υπολογίσει τη σωρευτική ανταμοιβή, ο πράκτορας υπολογίζει πρώτα μια επόμενη ενέργεια περνώντας την επόμενη παρατήρηση s_i' από τη δειγματοληπτική εμπειρία στον πράκτορα-στόχο. Ο πράκτορας βρίσκει την αθροιστική ανταμοιβή μεταβιβάζοντας την επόμενη ενέργεια στον κριτή-στόχο
  - Ενημερώνουμε τις παραμέτρους του κριτή ελαχιστοποιώντας την απώλεια L σε όλες τις δειγματοληπτικές εμπειρίες.

    $$L = \frac{1}{M}\sum_{i=1}^{M}(y_i - Q(s_i,a_i|\varphi))^2$$
  - Ενημερώνουμε τις παραμέτρους του πράκτορα χρησιμοποιώντας την ακόλουθη μερική πολιτική κλίσης (sampled policy gradient) για να μεγιστοποιήσουμε την αναμενόμενη εκπτωτική ανταμοιβή.

    $$\nabla_\theta J \approx \frac{1}{M}\sum_{i=1}^{M} G_{ai} G_{\pi i}$$

$$G_{ai} = \nabla_a Q(s_i, a|\pi) \quad \text{όπου } a = \pi(s_i, \theta)$$

$$G_{\pi i} = \nabla_\Theta \pi(s_i, \theta)$$

Εδώ, το G_{ai} είναι η κλίση της εξόδου του κριτή σε σχέση με τη δράση που υπολογίζεται από το δίκτυο πράκτορα, και $G_{\pi i}$ είναι η κλίση της εξόδου του πράκτορα σε σχέση με τις παραμέτρους του πράκτορα. Αμφότερες οι κλίσεις υπολογίζονται για παρατήρηση s_i.

- Ενημερώνουμε τις παραμέτρους του στόχου του πράκτορα και του κριτή ανάλογα με τη μέθοδο ενημέρωσης στόχου.

Μέθοδοι ενημέρωσης στόχου (Target update methods)

Οι πράκτορες βαθιάς ντετερμινιστικής πολιτικής κλίσης ενημερώνουν τις παραμέτρους του στόχου του πράκτορα και του κριτή χρησιμοποιώντας μία από τις ακόλουθες μεθόδους ενημέρωσης στόχου:

- **Smoothing** — Ενημερώνουμε τις παραμέτρους του στόχου σε κάθε χρονικό βήμα χρησιμοποιώντας τον συντελεστή εξομάλυνσης τ.
  $\varphi_t = \tau_\varphi + (1 - \tau)\varphi_t$ παράμετροι κριτή(critic parameters)
  $\theta_t = \tau_\theta + (1 - \tau)\theta_t$ παράμετροι πράκτορα(actor parameters)
- **Periodic** — Περιοδική ενημέρωση των παραμέτρων προορισμού χωρίς εξομάλυνση (TargetSmoothFactor = 1).
- **Periodic Smoothing** — Ενημερώνουμε περιοδικά τις παραμέτρους στόχου με εξομάλυνση

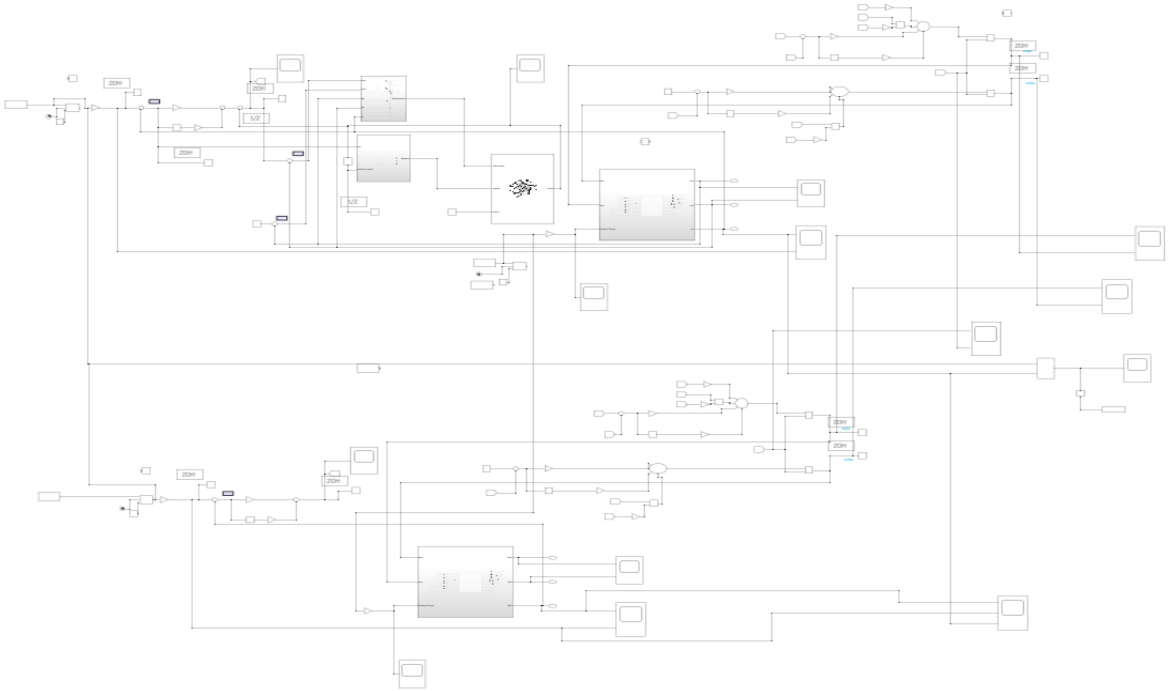
Μέθοδος Ενημέρωσης	Συχνότητα Ενημέρωσης Στόχου	Παράγοντας Εξομάλυνσης Στόχου
Smoothing (default)	1	Μικρότερο του 1
Periodic	Μεγαλύτερο του 1	1
Periodic Smoothing	Μεγαλύτερο του 1	Μικρότερο του 1

Στο συγκεκριμένο σημείο, θα αναλύσουμε τα ζητούμενα της διαδικασίας ελέγχου που έχουμε να επιληφθούμε και θα παρουσιάσουμε με ενδελεχή τρόπο τα επιμέρους χαρακτηριστικά του ελεγκτικού μηχανισμού που χρησιμοποιείται για την επίλυση του προβλήματος. Αρχικά, όπως είδαμε και προηγουμένως το σύστημα ελέγχου αποτελείται από δύο βρόχους ελέγχου˙ έναν εξωτερικό και έναν εσωτερικό. Η χρησιμοποίηση του RL ελεγκτή αφορά μόνο τον εξωτερικό βρόχο ελέγχου και σκοπός του είναι η βελτίωση της απόδοσης του εξωτερικού PI ελεγκτή .Αυτό επιτυγχάνεται μέσω της τροφοδότησης του PI ελεγκτή με συγκεκριμένες τιμές , οι οποίες προέρχονται από την έξοδο του RL ελεγκτή. Έτσι , οι τιμές στην έξοδο του RL ελεγκτή αποτελούν σήματα , τα οποία στέλνονται στον PI ελεγκτή , προκειμένου να συνδιαμορφώσουν την τελική έξοδό του.

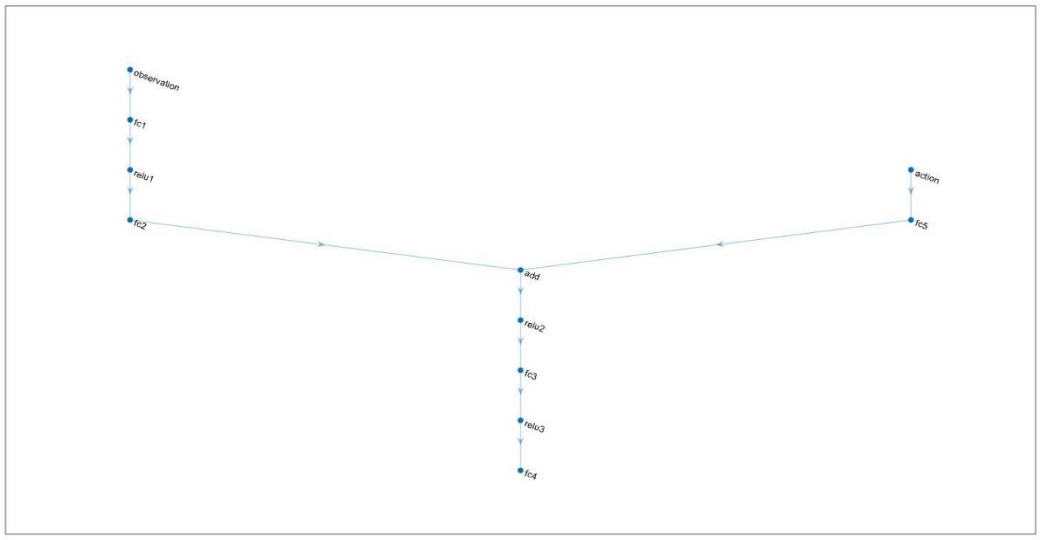
Όπως αναφέραμε και προηγουμένως για να χρησιμοποιηθεί ένας RL ελεγκτής πρέπει να οριστεί το περιβάλλον, οι καταστάσεις του περιβάλλοντος , οι ενέργειες και η συνάρτηση ανταμοιβής. Το περιβάλλον αποτελεί την αναπαράσταση λειτουργίας του κινητήρα ενός ηλεκτρικού αμαξιού και είναι πλήρως προσομοιωμένο στο Simulink της MatlabR2021a. Στο περιβάλλον αυτό έχει σχεδιαστεί επίσης και ο ελεγκτικός μηχανισμός του συστήματος. Σαν καταστάσεις έχουμε ορίσει τα προσανατολισμένα στο d-q σύστημα αναφοράς ρεύματα που διαπερνούν το στάτη(stator) της μηχανής I_{ds} και I_{qs}, τις στροφές του κινητήρα W_r και τα σφάλματα μεταξύ των ρευμάτων αναφοράς και των πραγματικών ρευμάτων του στάτη $E_{I_{ds}}$ και $E_{I_{qs}}$ αντίστοιχα. Η τιμή στην έξοδο του RL ελεγκτή αποτελεί την ενέργεια που αναλαμβάνει ο πράκτορας και το εύρος της τιμής ορίστηκε να είναι από -350 έως 350. Η σχέση που αναπαριστά τη συνάρτησης ανταμοιβής είναι η ακόλουθη: $reward\ function = -(3.2\ {e_{w_r}}^2 + 0.5|previous\ action|)$,όπου e_{w_r} είναι το σφάλμα των στροφών και $previous\ action$ είναι η προηγούμενη ενέργεια, δηλαδή $previous\ action = \frac{1}{z}\ action$.

Προκειμένου να γίνουν καλύτερα αντιληπτά αυτά τα οποία προαναφέραμε, παρακάτω ακολουθεί η εικόνα που αναπαριστά τη κύρια δομή του συστήματός μας. Έτσι, παρατίθεται το πλήρες γράφημα, το οποίο αφορά το συνολικό περιβάλλον στο οποίο δουλεύουμε. Στο άνω μέρος προσομοιώνεται ο ελεγκτικός μηχανισμός με τη χρησιμοποίηση του RL ελεγκτή ενώ στο κάτω μέρος ο PI έλεγχος του συστήματος. Όπως γίνεται αντιληπτό χρησιμοποιούνται πολλά plots, ούτως ώστε να έχουμε καθαρή εικόνα τόσο

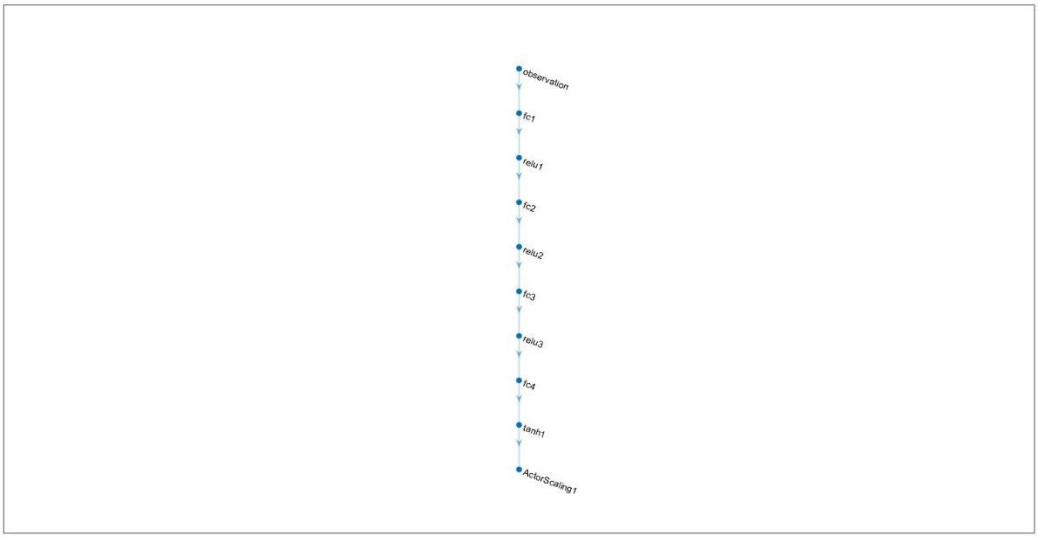
για την απόδοση του ελέγχου όσο και για τα επιμέρους χαρακτηριστικά του συστήματος.



Όσον αφορά τώρα τη δομή του πράκτορα , παρακάτω ακολουθεί η οπτική αναπαράσταση τόσο της δομής του κριτή όσο και του πράκτορα. Το ακόλουθο γράφημα αναπαριστά τη δομή του κριτή. Όπως βλέπουμε έχει δύο εισόδους (το πεδίο καταστάσεων και το πεδίο ενεργειών) και μια μονή έξοδο.



Το επόμενο γράφημα αναπαριστά τη δομή του πράκτορα. Σαν είσοδο δέχεται το πεδίο καταστάσεων και σαν έξοδο το πεδίο ενεργειών.



Να επισημάνουμε στο σημείο αυτό , κάποιες λεπτομέρειες για τη δομή του πράκτορα. Αρχικά, τα επιμέρους στρώματα-επίπεδα(layers) των νευρωνικών δικτύων αποτελούνται από εικοσιπέντε (25) νευρώνες το καθένα. Στον παρακάτω πίνακα ,παρατίθενται τα βασικά χαρακτηριστικά του αλγόριθμου επιλογής διαμόρφωσης των νευρωνικών δικτύων(rlDDPGAgentOptions).

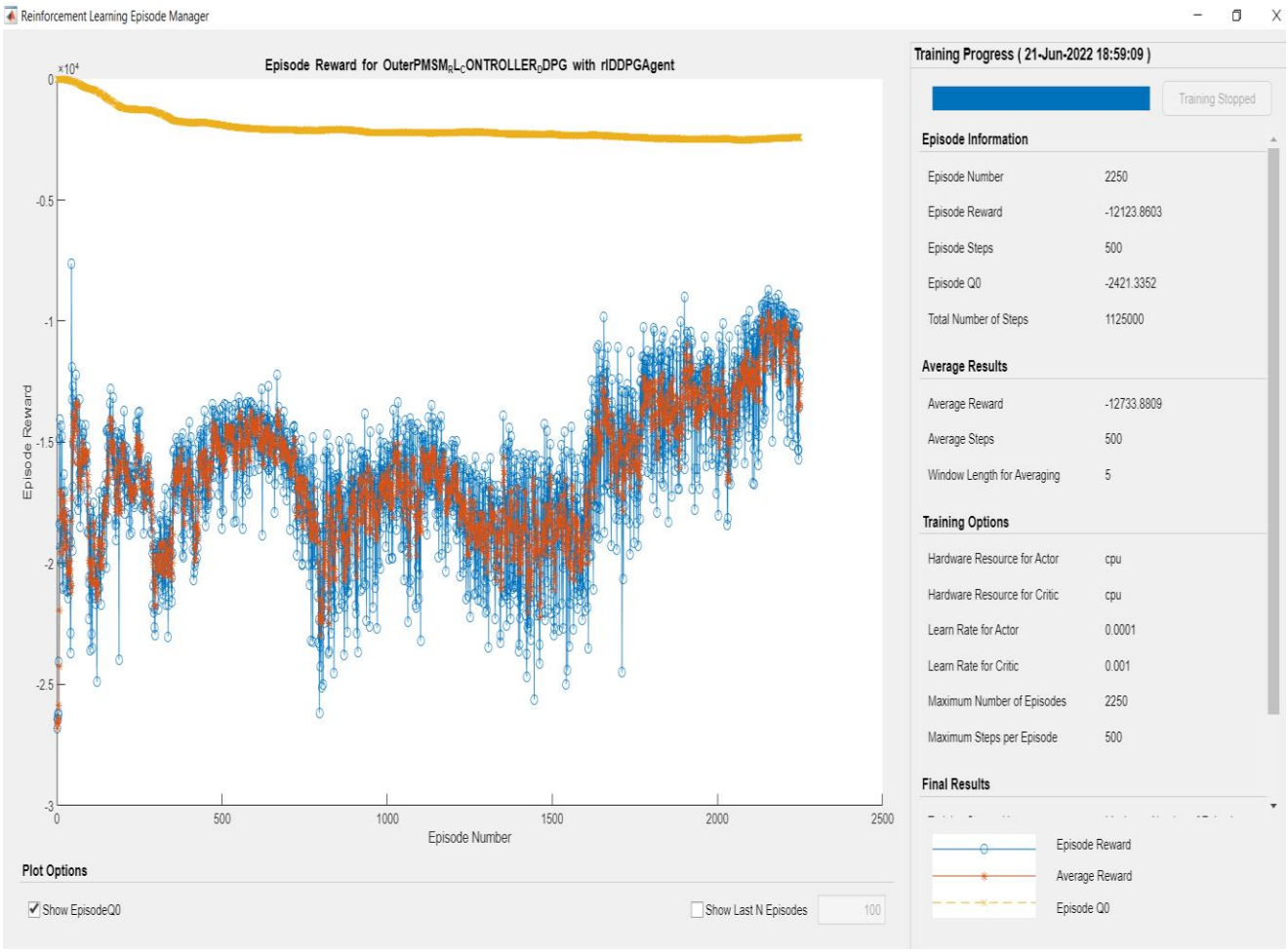
SampleTime	Ts=0.1
TargetSmoothFactor	1e-3
ExperienceBufferLength	1e6
DiscountFactor	0.99
MiniBatchSize	128
agentOptions.NoiseOptions.Variance	0.6
agentOptions.NoiseOptions.VarianceDecayRate	1e-5

Για την εκπαίδευση του ελεγκτή ορίσαμε συγκεκριμένες τιμές που αφορούν τις επιλογές τρόπου εκπαίδευσης του συστήματός μας. Στον παρακάτω πίνακα παρουσιάζονται οι τιμές, οι οποίες χρησιμοποιήθηκαν στον αλγόριθμο διαμόρφωσης χαρακτηριστικών της λειτουργίας εκπαίδευσης(rlTrainingOptions).

Ts	0.1
Tf	50
MaxEpisodes	2500
MaxStepsPerEpisode	$\frac{Tf}{Ts} = 500$
Verbose	false
StopTrainingCriteria	AverageReward
StopTrainingValue	-5
Plots	training-progress

Να σημειώσουμε εδώ , ότι δεν υπάρχουν τυποποιημένες τιμές για τα συγκεκριμένα λειτουργικά χαρακτηριστικά της διαδικασίας της εκπαίδευσης. Οι επιλογές έγιναν έπειτα από μια μακρά διαδικασία χρησιμοποίησης διάφορων τιμών και επιλογής εκείνων που απέδιδαν καλύτερα. Αυτό είναι κάτι απολύτως προφανές αν σκεφτούμε ότι η παραμικρή αλλαγή στις συγκεκριμένες τιμές δύναται να διαμορφώσει έναν διαφορετικό πράκτορα, δεδομένου ότι αφορούν τη δομή μοντελοποίησης του πράκτορα, ή μια διαφορετική διαδικασία εκπαίδευσης του συστήματος. Άρα γίνεται εύκολα κατανοητό, ότι η επιλογή των συγκεκριμένων τιμών παίζει καθοριστικό ρόλο τόσο στην διαμόρφωση όσο και στην απόδοση του ελεγκτικού μηχανισμού που θα χρησιμοποιηθεί για τον έλεγχο του συστήματος.

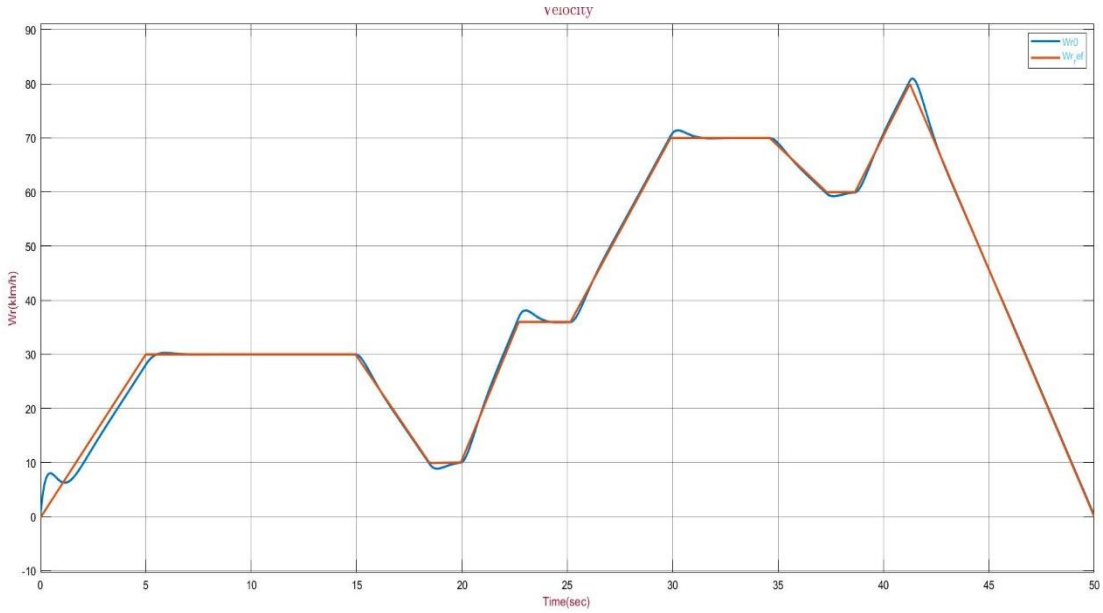
Στην παρακάτω εικόνα παρουσιάζεται εικονογραφημένη η διαδικασία εκπαίδευσης του πράκτορα. Κάθε κουκίδα (σημείο της γραφικής παράστασης ανά μονάδα χρόνου) αναπαριστά την τιμή της ανταμοιβής:



Η χρονική διάρκεια προκειμένου να ολοκληρωθεί όλη η διαδικασία είναι περίπου 8 ώρες. Γενικά, αυτό το οποίο παρατηρήθηκε κατά τη διάρκεια της εργασίας ήταν ότι η εκπαίδευση του ελεγκτή απαιτεί ένα μεγάλο χρονικό διάστημα. Το γεγονός αυτό σε συνδυασμό με την πληθώρα επιλογής εκπαιδευόμενων πρακτόρων ,καθιστά την όλη διαδικασία χρονοβόρα.

Στο σημείο αυτό παρουσιάζονται οι αποκρίσεις αλλά και αποδόσεις του ελεγκτή ενισχυτικής μάθησης μας για διάφορες τιμές στους PI ελεγκτές του εξωτερικού βρόχου. Αυτό το κάνουμε για να δούμε το πόσο εύρωστο αλλά και αποδοτικό είναι το σύστημα ελέγχου που έχουμε σχεδιάσει.

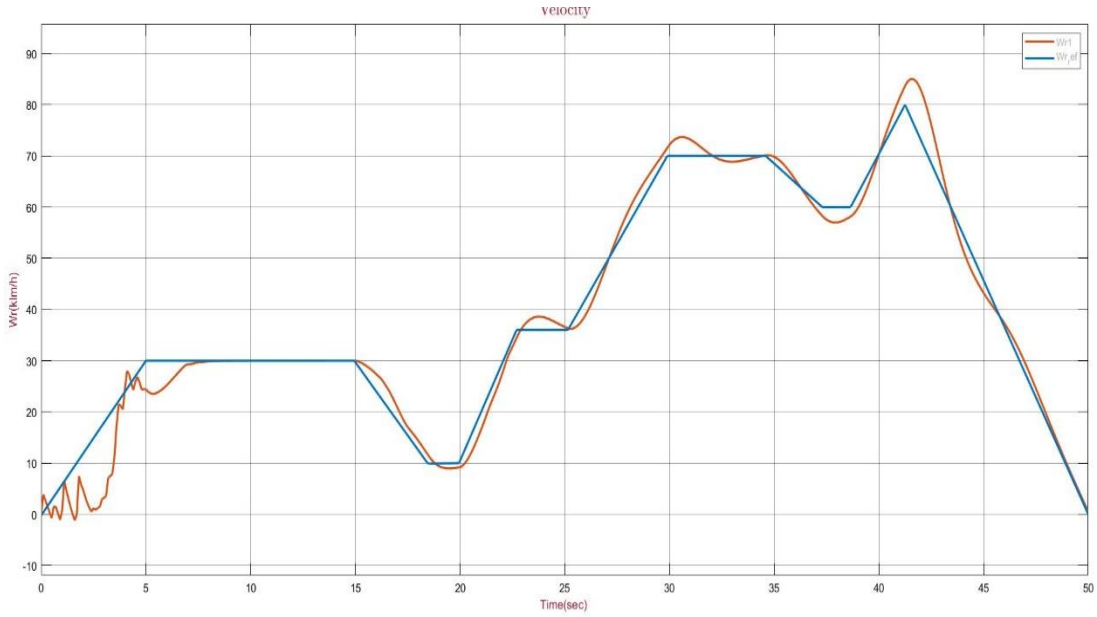
Αρχικά , στην πρώτη εικόνα αναπαρίσταται η βέλτιστη δυνατή απόκριση όπως αυτή προκύπτει από τη λειτουργία των αρχικών PI ελεγκτών τόσο στον εσωτερικό όσο και στον εξωτερικό βρόχο.



Αφού είδαμε προηγουμένως την βέλτιστη απόκριση με την εφαρμογή των PI ελεγκτών σε αυτό το στάδιο θα δοκιμάσουμε την εφαρμογή του ελεγκτή ενισχυτικής μάθησης μας για διάφορες τιμές στα βάρη των ελεγκτών στον εξωτερικό βρόχο ελέγχου. Στο σημείο αυτό πρέπει να αναφερθεί ότι εκτός της αναπαράστασης της απόκρισης των στροφών θα παρατίθεται και η απόκλιση τους σε σχέση με την βέλτιστη απόκριση κάθε φορά.

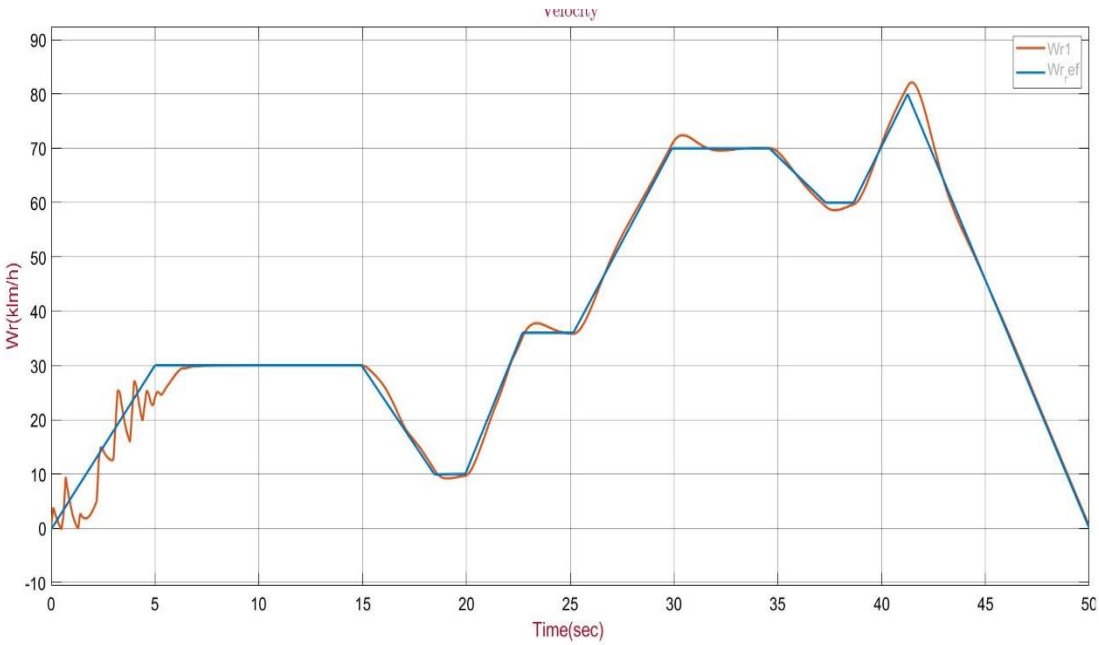
Στην παρακάτω φωτογραφία φαίνεται η απόκριση του συστήματος για τις εξής τιμές:

K_{P_ω}=7.5 και K_{I_ω}=15



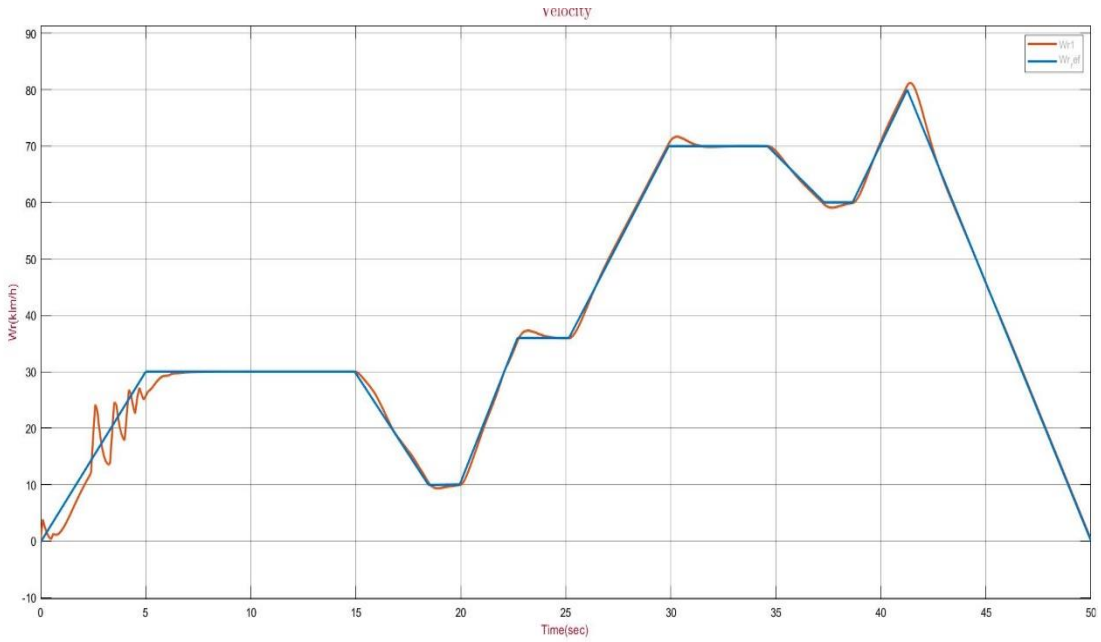
Η μέση απόκλιση είναι 2.2620

Παρακάτω φαίνεται η απόκριση για τις εξής τιμές: K_{P_ω}15 και K_{I_ω}=30



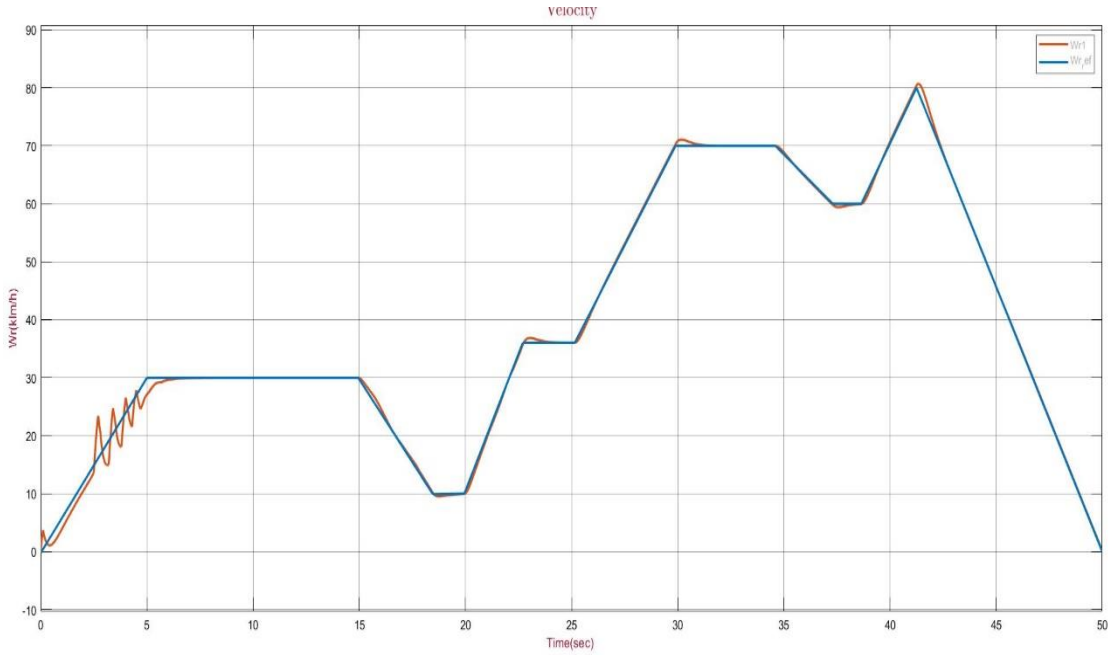
Η μέση απόκλιση είναι 1.1508

Παρακάτω φαίνεται η απόκριση για τις εξής τιμές: K_{P_ω}=25 και K_{I_ω} =50



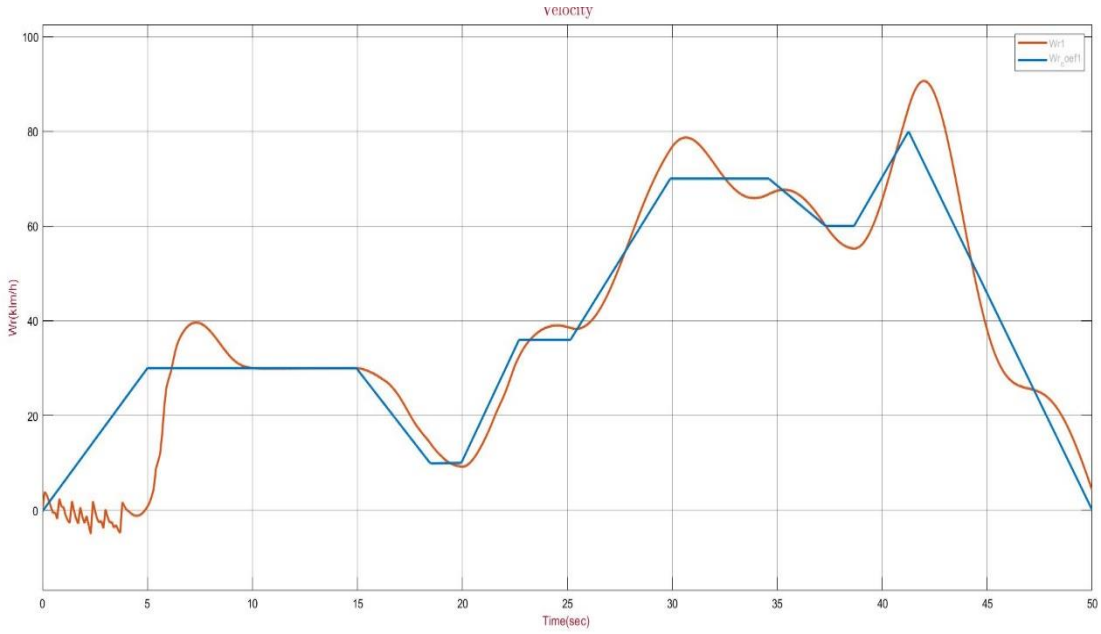
Η μέση απόκλιση είναι 0.7188

Παρακάτω φαίνεται η απόκριση για τις εξής τιμές: K_{P_ω}=45 και K_{I_ω}=85



Η μέση απόκλιση είναι 0.4392

Παρακάτω φαίνεται η απόκριση για τις εξής τιμές: K_{P_ω}=3.5 και K_{I_ω}=7.5



Η μέση απόκλιση είναι 5.5877

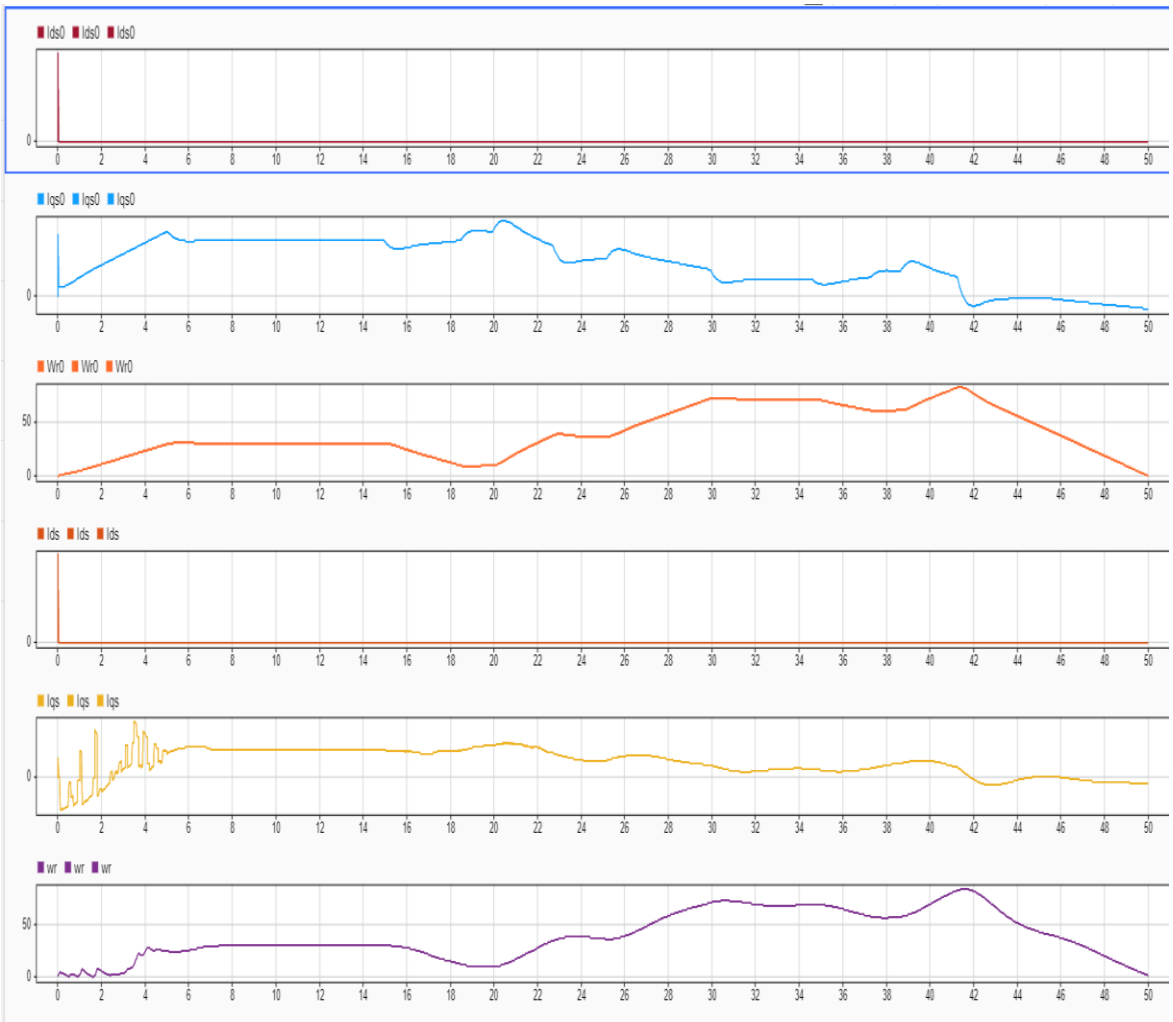
Τα αποτελέσματα είναι αναμενόμενα αφού όσο τείνουμε να δώσουμε στα βάρη των ελεγκτών τιμές κοντά στις βέλτιστες δυνατές τόσο καλύτερη γίνεται η απόκριση . Επίσης παρατηρούμε ότι ο ελεγκτής κρίνεται αποδοτικός αφού επιδρά θετικά στο σύστημα και τείνει να διορθώνει κάθε φορά τις αστοχίες που προκύπτουν από την χρησιμοποίηση διαφορετικών κερδών σε σχέση με αυτά που έχουν προκύψει από τη μαθηματικοποίηση του συστήματος και στον μετέπειτα προσδιορισμό των βέλτιστων δυνατών ελεγκτών PI.

Να σημειωθεί επίσης ότι δώσαμε στα κέρδη μια μεγάλη γκάμα τιμών ούτως ώστε να δοκιμάσουμε τον ελεγκτή μας για μια πληθώρα καταστάσεων. Έτσι επιλέξαμε από πολύ μικρές τιμές για να προσομοιώσουμε την κατάσταση που δεν θα έχουμε εξ' αρχής ουσιαστικό εξωτερικό έλεγχο μέχρι και αρκετά μεγαλύτερες από τις βέλτιστες προκειμένου να δούμε πως αποκρίνεται και εκεί το σύστημά μας.

Εν συνεχεία παρουσιάζονται οι αποκρίσεις των επιμέρους χαρακτηριστικών του συστήματός μας. Αυτά τα χαρακτηριστικά είναι τα προσανατολισμένα στους δύο άξονες d και q ρεύματα , η τάση στον πυκνωτή και προφανώς οι στροφές του κινητήρα.

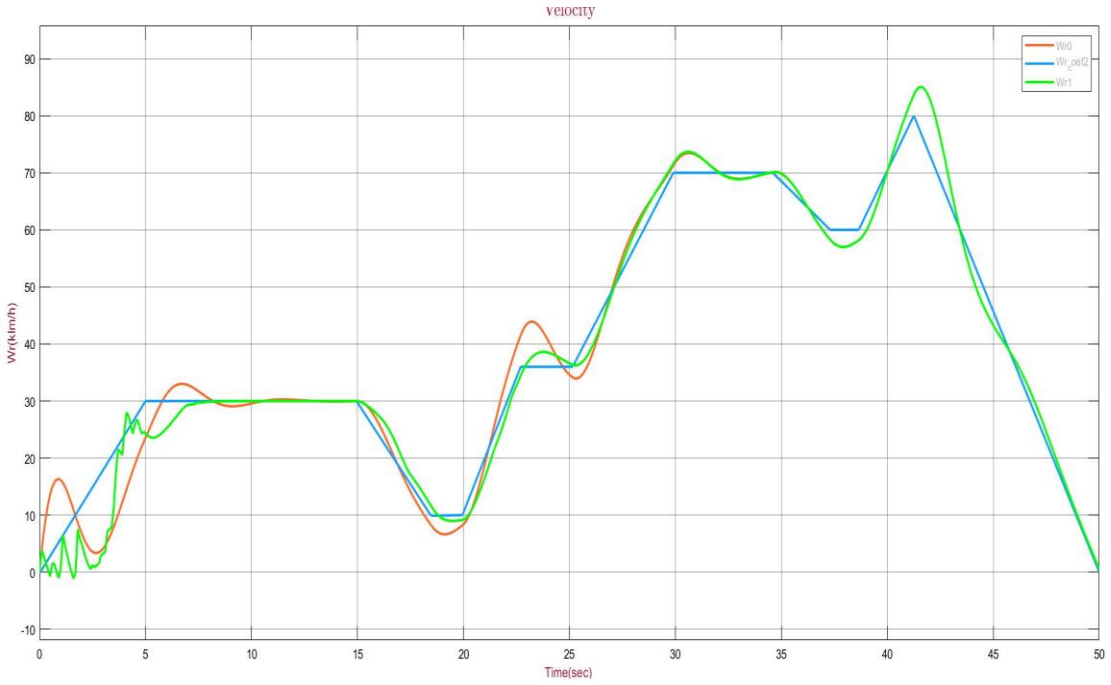
Για να γίνεται η σύγκριση με το αρχικό σύστημα , το οποίο όπως είπαμε προκύπτει από τη χρήση των βέλτιστων κερδών στον εξωτερικό έλεγχο, σε κάθε γράφημα θα απεικονίζονται τόσο οι αποκρίσεις του συστήματος με ενσωματωμένο του RL ελεγκτή όσο και του αρχικού συστήματος ελέγχου.

Στο παρακάτω γράφημα παρουσιάζονται οι αποκρίσεις των ρευμάτων και των στροφών. Οι όροι I_{ds_0}, I_{qs_0} και W_{r_0} αναφέρονται στο αρχικό σύστημα και αντίστοιχα οι I_{ds}, I_{qs} και W_r στο RL μοντέλο ελέγχου. Να σημειώσουμε ότι τα βάρη στον εξωτερικό έλεγχο επιλέχθηκαν 7.5 και 15.



Τώρα, προκειμένου να διαπιστώσουμε το αν και πόσο αποδοτικός είναι ο ελεγκτής μας , θα αντιπαραθέσουμε εικόνες αποκρίσεων χωρίς την χρησιμοποίηση του ελεγκτή σε αντιδιαστολή με εικόνες αφότου έχει συνδράμει στο σύστημά μας.

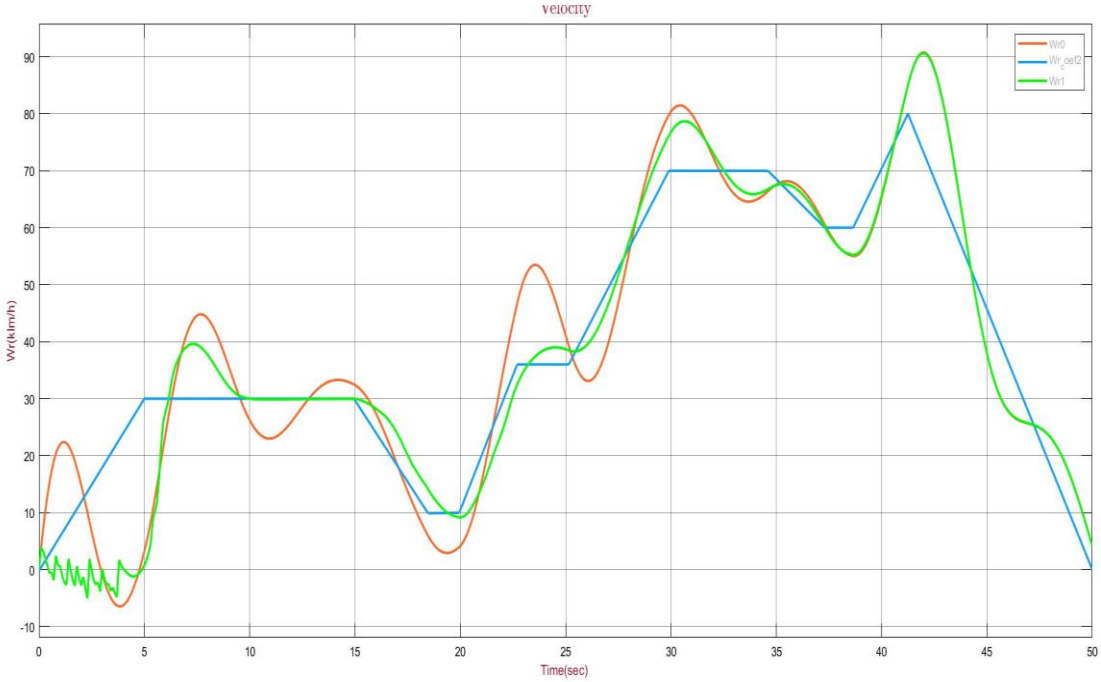
Αρχικά, θα δώσουμε τιμές K_{P_ω}=7.5 και K_{I_ω}=15 στα κέρδη του ελεγκτή εξωτερικού βρόχου και θα δούμε στο παρακάτω γράφημα τις διαφορές στην απόκριση των στροφών μεταξύ των δύο συστημάτων.



Με το μπλέ χρώμα απεικονίζεται το σήμα αναφοράς του συστήματος , με το πράσινο χρώμα απεικονίζονται οι στροφές υπό τη χρησιμοποίηση του ελεγκτή και με το πορτοκαλί χρώμα απεικονίζονται οι στροφές του συστήματος χωρίς τη λειτουργία του ελεγκτή μας. Είναι φανερό ότι η χρησιμοποίηση του ελεγκτή βελτιώνει αισθητά την απόκριση του συστήματός μας.

Όπως είδαμε προηγουμένως η μέση απόκλιση με τη χρησιμοποίηση του ελεγκτή για τη συγκεκριμένη επιλογή κερδών είναι 2.2620.Χωρίς τη χρησιμοποίηση του ελεγκτή είναι 2.7308.Αυτό σημαίνει πρακτικά ότι το σύστημά μας βελτιώνεται σε ποσοστό 17,17%.

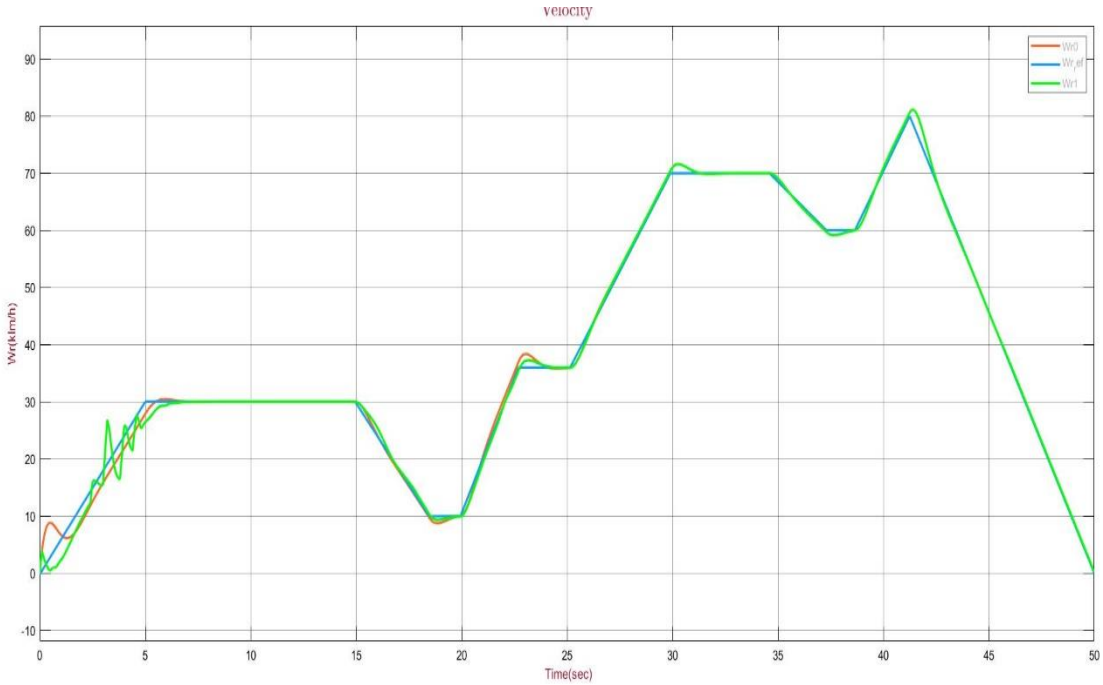
Θα ακολουθήσουμε την ίδια διαδικασία για τις εξής τιμές στα βάρη του εξωτερικού βρόχου ελέγχου : K_{P_ω}3.5 και K_{I_ω}=7.5



Και αυτή τη φορά με το μπλέ χρώμα απεικονίζεται το σήμα αναφοράς του συστήματος , με το πράσινο χρώμα απεικονίζονται οι στροφές υπό τη χρησιμοποίηση του ελεγκτή και με το πορτοκαλί χρώμα απεικονίζονται οι στροφές του συστήματος χωρίς τη λειτουργία του ελεγκτή μας.

Η μέση απόκλιση με τη χρησιμοποίηση του ελεγκτή είναι 5.5877. Χωρίς την επίδραση του ελεγκτή είναι 7.4385. Αυτό σημαίνει πρακτικά ότι το σύστημά μας βελτιώνεται σε ποσοστό 24,88%.Παρατηρούμε ότι το ποσοστό βελτίωσης είναι μεγαλύτερο όσο απομακρυνόμαστε από τις βέλτιστες τιμές των κερδών του εξωτερικού βρόχου ελέγχου.

Παρακάτω ακολουθείται η ίδια διαδικασία με πριν με μόνη διαφορά την αλλαγή των βαρών του ελεγκτή. Αυτή τη φορά θα επιλέξουμε K_{P_ω}=25 και K_{I_ω}=55.Το γράφημα παρουσιάζεται ακριβώς παρακάτω:



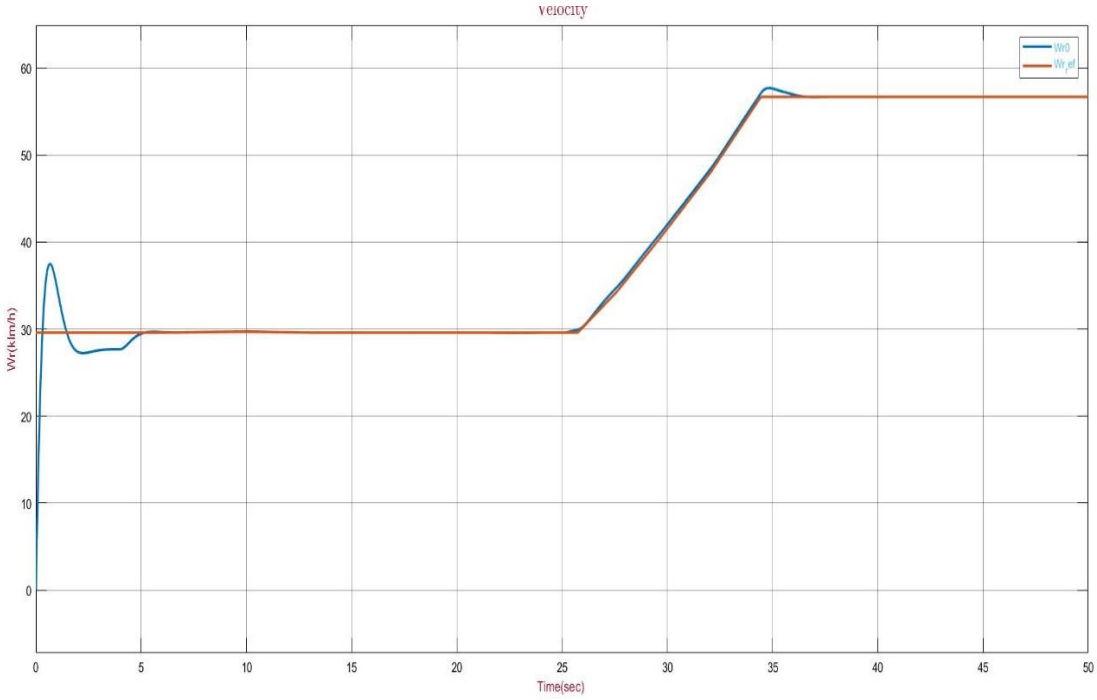
Όπως και πριν έτσι και τώρα με το μπλέ χρώμα απεικονίζεται το σήμα αναφοράς του συστήματος , με το πράσινο χρώμα απεικονίζονται οι στροφές υπό τη χρησιμοποίηση του ελεγκτή και με το πορτοκαλί χρώμα απεικονίζονται οι στροφές του συστήματος χωρίς τη λειτουργία του ελεγκτή μας.

Η μέση απόκλιση με τη χρησιμοποίηση του ελεγκτή είναι 0.6399. Χωρίς την επίδραση του ελεγκτή είναι 0.6322.Αυτό πρακτικά σημαίνει ότι για τιμές που σχεδόν προσεγγίζουν στο ακέραιο τις βέλτιστες δυνατές τιμές του ελεγκτή ,ο ελεγκτής μειώνει κατά 1,22% το ποσοστό επιτυχημένης απόκρισης του συστήματος.

Αν παρατηρήσει κανείς προσεκτικά το γράφημα , διακρίνεται ότι το βασικό ποσοστό αποτυχίας του ελεγκτή ως προς την καλυτέρευση της απόδοσης του συστήματος βρίσκεται στην αρχή της διαδικασίας. Στη συνέχεια όχι μόνο δε δυσχεραίνει την απόδοση του συστήματος αλλά τη βελτιώνει κιόλας.

Για να αποκτήσουμε όμως μια πιο σφαιρική άποψη για την απόδοση του ελεγκτή μας σε μια ευρεία κλίμακα θα πρέπει να αξιολογήσουμε την λειτουργικότητά του σε διαφορετικών τύπων σημάτων εισόδου. Για αυτό το λόγο θα επαναλάβουμε την διαδικασία που ακολουθήθηκε παραπάνω προκειμένου να λάβουμε μια πιο πλήρη εικόνα της αποδοτικότητας του συστήματος ελέγχου. Ως εκ τούτου θα αλλάξουμε τη μορφή του σήματος εισόδου , αναπαριστώντας αυτή τη φορά ένα σήμα που θα προσεγγίζει βηματικές εισόδους.

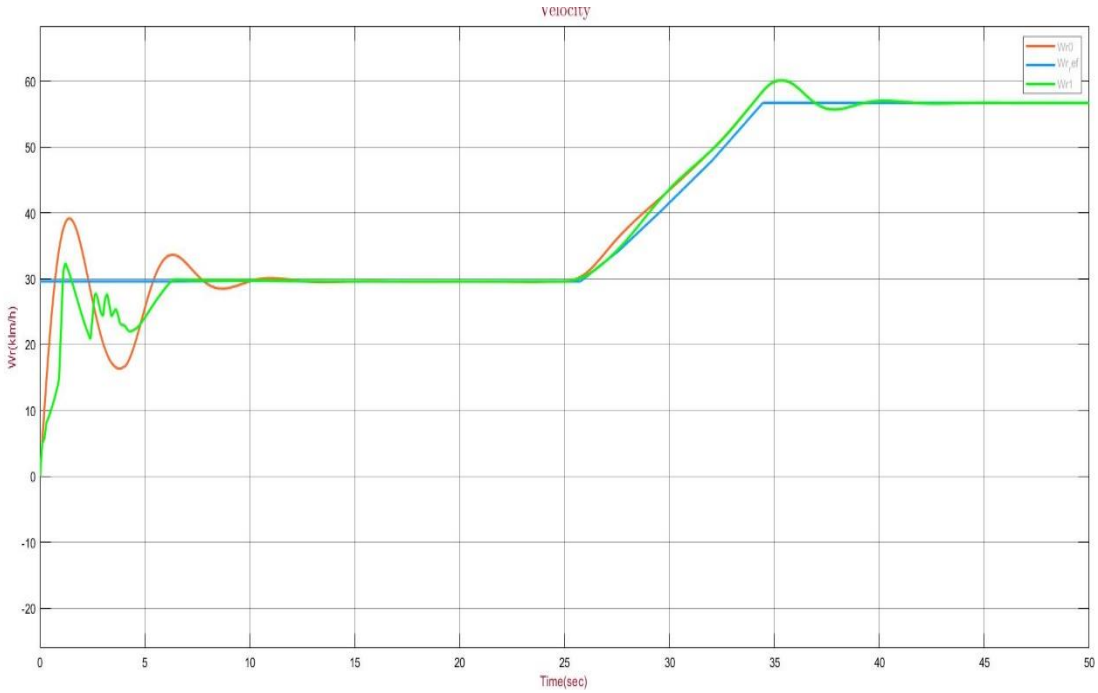
Αρχικά στο παρακάτω γράφημα βλέπουμε την βέλτιστη απόκριση του συστήματος ύστερα από την χρησιμοποίηση των κερδών στον εξωτερικό βρόχο ελέγχου.



Όπως και πριν έτσι και τώρα θα ακολουθήσουμε τη διαδικασία ανάλυσης και παρουσίασης των επιμέρους αποκρίσεων του συστήματός μας υπό την χρησιμοποίηση του RL

ελεγκτή για διάφορες τιμές στα κέρδη του εξωτερικού βρόχου ελέγχου. Αυτή τη φορά θα παρουσιάζεται σε κοινό γράφημα τόσο η απόκριση του συστήματος με τον ελεγκτή όσο και χωρίς αυτόν. Έπειτα θα αναγράφονται οι αποκλίσεις κάθε φορά και θα ελέγχεται με αυτόν τον τρόπο η απόδοση του συστήματος ελέγχου που εφαρμόζουμε.

Αρχικά, θα δώσουμε τιμές K_{P_ω}7.5 και K_{I_ω}=15 στα κέρδη του ελεγκτή. Τα αποτελέσματα φαίνονται στο παρακάτω γράφημα:

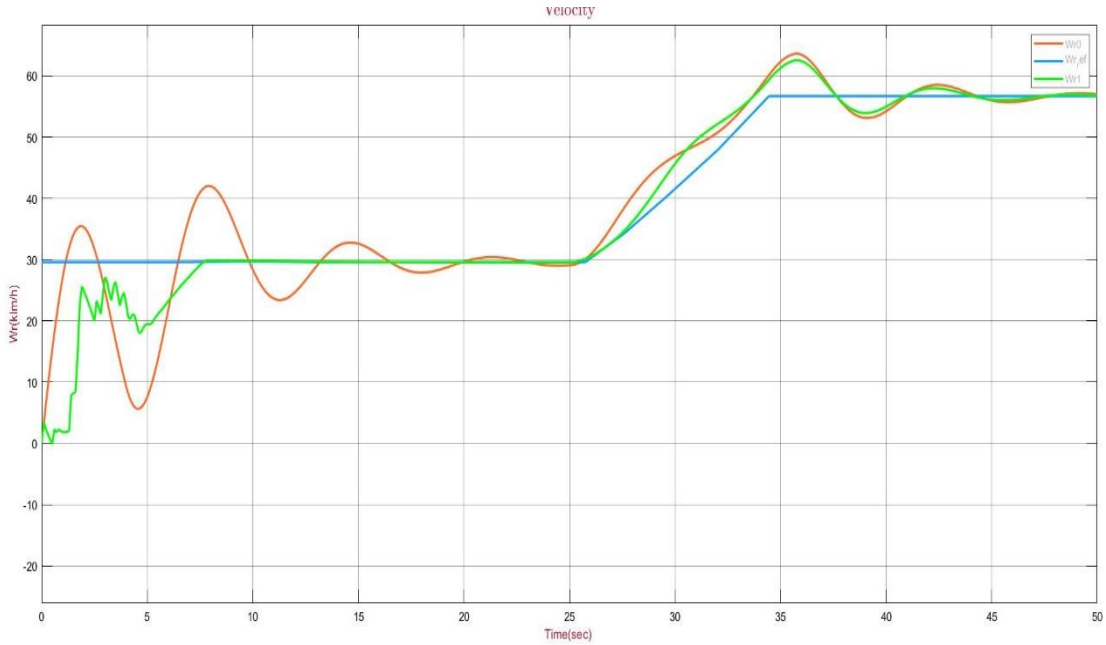


Όπως και πριν έτσι και τώρα με το μπλέ χρώμα απεικονίζεται το σήμα αναφοράς του συστήματος , με το πράσινο χρώμα απεικονίζονται οι στροφές υπό τη χρησιμοποίηση του ελεγκτή και με το πορτοκαλί χρώμα απεικονίζονται οι στροφές του συστήματος χωρίς τη λειτουργία του ελεγκτή μας.

Η απόκλιση με τη χρήση του ελεγκτή είναι 1.2332 και χωρίς 1.5353. Άρα έχουμε ένα ποσοστό βελτίωσης της τάξης του 19,68%.

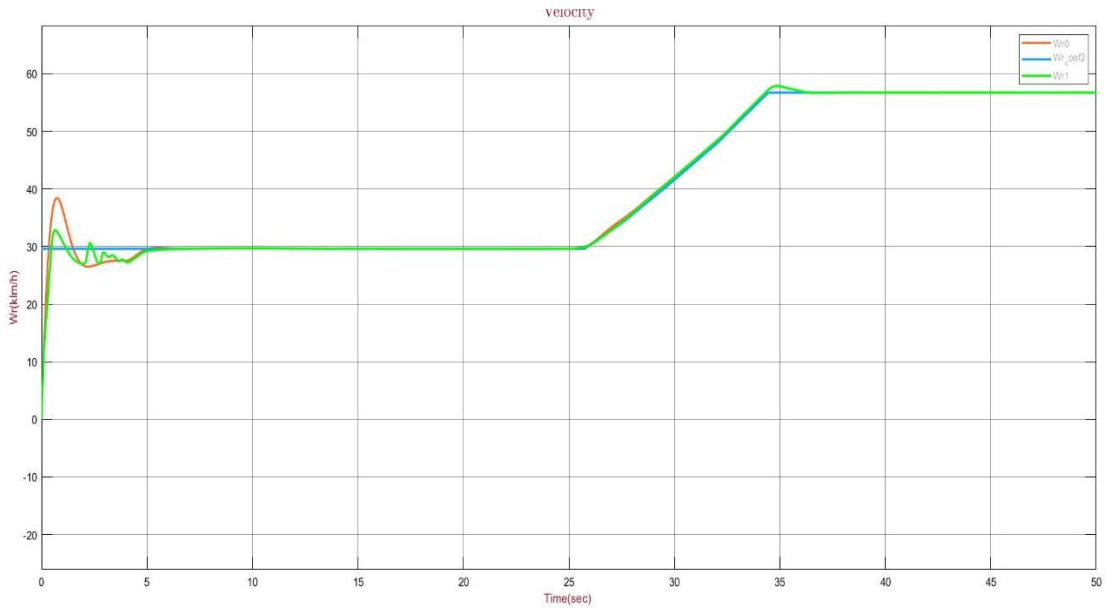
Θα ακολουθήσουμε την ίδια διαδικασία για τις εξής τιμές στα βάρη του εξωτερικού βρόχου ελέγχου : K_{P_ω}=3.5 και K_{I_ω}7.5

Τα αποτελέσματα φαίνονται στο παρακάτω γράφημα:



Η απόκλιση με τη χρήση του ελεγκτή είναι 2.5617 και χωρίς 3.3160. Άρα έχουμε ένα ποσοστό βελτίωσης της τάξης του 22,75%.

Εδώ ακολουθείται η ίδια διαδικασία με πριν με μόνη διαφορά την αλλαγή των βαρών του ελεγκτή. Αυτή τη φορά θα επιλέξουμε K_{P_ω}=25 και K_{I_ω}=55.Το γράφημα παρουσιάζεται ακριβώς παρακάτω:



Η απόκλιση με τη χρήση του ελεγκτή είναι 0.3570 και χωρίς 0.4640. Άρα έχουμε ένα ποσοστό βελτίωσης της τάξης του 23,06%.

ΚΕΦΑΛΑΙΟ 7ο

Συμπεράσματα

Το αντικείμενο της συγκεκριμένης διπλωματικής εργασίας ήταν η διαμόρφωση και η παρουσίαση ενός ελεγκτικού μηχανισμού με βάση τη συνέργεια PI ελεγκτών και ενισχυτικής μάθησης , για τη ρύθμιση του κινητήρα ενός ηλεκτρικού οχήματος. Κύριος στόχος ήταν να διερευνηθεί το κατά πόσο είναι δυνατόν RL ελεγκτές να συνδράμουν στην βελτίωση της απόδοσης ενός αρχικού συστήματος PI ελεγκτών.

Η σκέψη αυτή προήλθε από το γεγονός ότι πολλές φορές κρίνεται αδύνατη ή ιδιαίτερα δύσκολη η λεπτομερής μαθηματική προτυποποίηση της δυναμικής, που δυσχεραίνει το σχεδιασμό μηχανισμών αυτόματου ελέγχου.

Τα αποτελέσματα της εργασίας έδειξαν ότι η χρησιμοποίηση της ενισχυτικής μάθησης οδήγησε σε αποδοτική ρύθμιση του συστήματος και ενθαρρύνουν τη περαιτέρω μελέτη και αξιοποίησή της. Τα αποτελέσματα προήλθαν έπειτα από εκτεταμένη ανάλυση προσομοιώσεων και γραφικών απεικονίσεων των επιμέρους χαρακτηριστικών του συστήματος, με βάση τα οποία η απόδοση του ελέγχου βελτιώθηκε σε ποσοστά 17% έως 25%.Σημειώνεται ότι η απόδοση του υβριδικού PI-RL μηχανισμού ελέγχου μετρήθηκε και αξιολογήθηκε για δύο διαφορετικά σενάρια σημάτων εισόδου-αναφοράς (reference signals) αλλά και για μια πληθώρα επιλογών βαρών στους PI ελεγκτές του εξωτερικού βρόχου, κάτι που πρακτικά καταδεικνύει την εύρωστη απόδοση του προτεινόμενου σχήματος ελέγχου.

Σημειώνεται ότι για να καταλήξουμε σε αυτόν τον αποδοτικό σχεδιασμό, χρειάστηκε να διερευνήσουμε και άλλες αρχιτεκτονικές ενισχυτικής μάθησης. Αρχικά δοκιμάστηκε ο αλγόριθμος DQN ,η χρησιμοποίηση του οποίου δεν έφερε τα προσδοκόμενα αποτελέσματα. Έτσι καταλήξαμε στον DDPG αλγόριθμο , ο οποίος ,όπως αναφέραμε και προηγουμένως , κάλυψε σε σημαντικό βαθμό τους στόχους που θέσαμε κατά την έναρξη της εργασίας. Οι επιλογές των επιμέρους χαρακτηριστικών του αλγόριθμου έγιναν με κριτήρια την απόδοσή του ως προς το προαναφερθέν σύστημα και η εύρεση αυτών προέκυψε μετά από εκτεταμένες δοκιμές και αξιολογήσεις.

Η επιλογή του λογισμικού του Simulink της Matlab έγινε γιατί συνδέεται με μια εξειδικευμένη εργαλειοθήκη ενισχυτικής μάθησης συνδυασμού και ολοκλήρωσης με άλλα συστήματα (όπως το μοντέλο του κινητήρα και οι ελεγκτές PI) που επιτρέπουν την πλήρη αναπαράσταση και προσομοίωση του συνολικού συστήματος. Επίσης παρέχει αυτοματοποιημένα εργαλεία για το σχεδιασμό των βαθιών νευρωνικών δικτύων που αντιστοιχούν στον πράκτορα και τον κριτή.

Συνοψίζοντας, η ενισχυτική μάθηση είναι μια τεχνολογία που βασίζεται στην τεχνητή νοημοσύνη και μπορεί να βελτιώσει σημαντικά την απόδοση των συμβατικών συστημάτων αυτομάτου ελέγχου. Σημειώνεται όμως ότι ο σχεδιασμός, η διαμόρφωση και η εφαρμογή ενός συστήματος ελέγχου βασισμένο στην ενισχυτική μάθηση απαιτεί χρόνο και χρήζει εμπειρίας και καλής γνώσης του συστήματος. Επομένως αποτελεί μια υποσχόμενη, ανοικτή και εξαιρετικά ενδιαφέρουσα κατεύθυνση έρευνας και περαιτέρω μελέτης και ανάπτυξης στην περιοχή των συστημάτων ελέγχου.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Sutton, Richard S.; Barto, Andrew G.” Reinforcement Learning: An Introduction”. ISBN 10: 0262193981 ISBN 13: 9780262193986

[2] Bellman, R. “A Markovian Decision Process.” In: *Indiana University Mathematics Journal* 6.5 (1957), pp. 679–684. issn: 0022-2518. url: https://www.jstor.org/stable/24900506

[3] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. “OpenAI Gym.” 2016. arxiv: 1606.01540.

[4] Barto, A. G., Sutton, R. S., and Anderson, C. W. “Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems.” In: *IEEE Transactions on Systems, Man, & Cybernetics* 13.5 (1983), pp. 834–846. url: https://ieeexplore.ieee.org/abstract/document/6313077.

[5] Williams, R. J. “Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning.” In: *Machine Learning* 8.3–4 (May 1992), pp. 229–256. issn: 0885-6125. doi: 10.1007/BF00992696.

[6] Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. “Policy Gradient Methods for Reinforcement Learning with Function Approximation.” In: *NeurIPS’99 Proceedings of the 12th International Conference on Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 1999, pp. 1057–1063. url: http://dl.acm.org/citation.cfm?id=3009657.3009806.

[7] Rummery, G. A. and Niranjan, M. *On-Line Q-Learning Using Connectionist Systems*. Tech. rep. University of Cambridge, 1994.

[8] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. A. “Playing Atari with Deep Reinforcement Learning.” 2013. arxiv: 1312.5602.

[9] van Hasselt, H., Guez, A., and Silver, D. “Deep Reinforcement Learning with Double Q-Learning.” 2015. arxiv: 1509.06461.

[10] Schaul, T., Quan, J., Antonoglou, I., and Silver, D. “Prioritized Experience Replay.” 2015. arxiv: 1511.05952.

[11] Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., et al. “QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation.” 2018. arxiv: 1806.10293.

[12] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. “Mastering the Game of Go with Deep Neural Networks and Tree Search.” In: *Nature* 529.7587 (2016), pp. 484–489.

[13] Li, W. and Todorov, E. “Iterative Linear Quadratic Regulator Design for Nonlinear Biological Movement Systems.” In: *Proceedings of the 1st International Conference on Informatics in Control, Automation and Robotics (ICINCO 1)*. Ed. by Araújo, H., Vieira, A., Braz, J., Encarnação, B., and Carvalho, M. INSTICC Press, 2004, pp. 222–229. isbn: 972-8865-12-0.

[14] Schulman, J., Levine, S., Moritz, P., Jordan, M. I., and Abbeel, P. “Trust Region Policy Optimization.” 2015. arxiv: 1502.05477.

[15] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. “Proximal Policy Optimization Algorithms.” 2017. arxiv: 1707.06347.

[16] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. “Continuous Control with Deep Reinforcement Learning.” 2015. arxiv: 1509.02971.

[17] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor.” 2018. arxiv: 1801.01290.

[18] Sutton, R. S. “Dyna, an Integrated Architecture for Learning, Planning, and Reacting.” In: *ACM SIGART Bulletin* 2.4 (July 1991), pp. 160–163. issn: 0163-5719. doi: 10.1145/122344.122377.

[19] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. Journal of Physiology, 160:106–154, 1962.

[20] F. Fukushima. A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetic, 36:193–202, 1980.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems,pages 1097–1105, 2012.

[22] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. nature, 521(7553):436,2015.

[23] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. Neural networks, 2(5):359–366, 1989.

[24] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016.
http://www.deeplearningbook.org.

[25] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda,Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. Knowledge and information systems, 14(1):1–37, 2008.

[26] Steven L. Brunton, J. Nathan Kutz.” Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control “.ISBN-10 : 1108422098. ISBN-13 : 978-1108422093

[27] Williams, R. J. “Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning.” In: *Machine Learning* 8.3–4 (May 1992), pp. 229–256. issn: 0885-6125. doi: 10.1007/BF00992696.

[28] Barto, A. G., Sutton, R. S., and Anderson, C. W. “Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems.” In: *IEEE Transactions on Systems, Man, & Cybernetics* 13.5 (1983), pp. 834–846. url: https://ieeexplore.ieee.org/abstract/document/6313077.

[29] Schulman, J., Moritz, P., Levine, S., Jordan, M. I., and Abbeel, P. “High-Dimensional Continuous Control Using Generalized Advantage Estimation.” 2015. arxiv: 1506.02438.

[30] Εμμανουηλίδης Γ. (2011) ΗΛΕΚΤΡΙΚΑ ΑΥΤΟΚΙΝΗΤΑ . [Online]. Available: https://en.calameo.com/read/005021287abec2ddebdb2

[31] Reinforcement Learning Toolbox™ User's Guide © COPYRIGHT 2019–2022 by The MathWorks, Inc.url:
https://www.mathworks.com/help/pdf_doc/reinforcement-learning/rl_ug.pdf