Εθνικο Μετσοβιο Πολυτεχνειο
Τμημα Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης

Μελέτη ασύγχρονων μεθόδων βαθιάς ενισχυτικής μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Γεώργιου Χρήστου Τσιατσιάνη

Επιβλέπων: Ανδρέας Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2022

Εθνικό Μετσόβιο Πολυτεχνείο
Τμήμα Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και
Υπολογιστών
Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης
και Μάθησης

Μελέτη ασύγχρονων μεθόδων βαθιάς ενισχυτικής μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Γεώργιου Χρήστου Τσιατσιάνη

Επιβλέπων: Ανδρέας Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις 20 Σεπτεμβρίου 2022.

.....	.....	.....
Ανδρέας Σταφυλοπάτης	Γεώργιος Στάμου	Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2022

.....
Γεώργιος Χρήστος Τσιατσιάνης
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών
Ε.Μ.Π. και απόφοιτος του μεταπτυχιακού προγράμματος σπουδών Μαθηματική
Προτυποποίηση σε Σύγχρονες Τεχνολογίες και τη Χρηματοοικονομική Ε.Μ.Π.

επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Η βαθιά ενιχσυτική μάθηση αποτελεί ένα από τα πιο υποσχόμενα πεδία της μηχανικής μάθησης. Φαίνεται να επιλυεί αποδοτικά ένα ευρύ φάσμα προβλημάτων. Η ασυγχρονη ενισχυτική μάθηση αποτελεί εξέλιξη στο συγκεκριμένο τομέα. Επιχειρεί να επιλύσει τα προβλήματα ταχύτερα αποδοτικότερα και εξερευνώντας το περιβάλλον μάθησης του πράκτορα πολύ καλύτερα και παρέχοντας μεγαλύτερη ευστάθεια στο σύστημα.

Στα πρώτα κεφάλαια της παρούσας διπλωματικής εργασίας, εισάγουμε με αναλυτή μεθοδολογία τον αναγνώστη πάνω στις έννοιες της ενισχυτικής μάθησης. Στη συνέχεια, παρέχουμε στον αναγνώστη το απαραίτητο θεωρητικό για να κατανοήσει την αναγκαιότητα χρήσης της βαθιάς ενισχυτικής μάθησης. Εξηγούμε τα βασικότερα είδη βαθιάς ενισχυτικής μάθησης. Τέλος, εισάγουμε τον αναγνώστη στις έννοιες της ασύγχρονης ενισχυτικής μάθησης.

Στην παρούσα διπλωματική εργασία, βασιζόμενοι στην βιβλιογραφία [19] υλοποιήσαμε και εξετάσαμε τη συμπεριφορά των ασύγχρονων αλγορίθμων asynchronous one step Sarsa, asynchronous one step Q-learning, asynchronous n-step Q-learning, asynchronous advantage Actor-Critic. Επίσης, υλοποιήσαμε και τους αλγορίθμους βαθιάς ενισχυτικής μάθησης DQN, Double DQN, Dueling DQN. "Τρέξαμε" τα προγράμματα τύπου DQN σε κάρτα γραφικών (GPU), ενώ τους αλγόριθμους ασύγχρονης ενισχυτικής μάθησης σε απλή CPU. Συγκρίναμε τις επιδόσεις όλων των προγραμμάτων ενισχυτικής μάθησης πάνω στο παιχνίδι Cart-pole.

Λέξεις κλειδιά

βαθιά ενισχυτική μάθηση, ασύγχρονη, Asynchronous Advantage Actor-Critic, ανάλυση, υλοποίηση, μηχανική μάθηση

Abstract

Deep neural learning is one of the most promising fields of machine learning and appears to efficiently solve a range of problems. Asynchronous reinforcement learning is a new tendency in this field. It attempts to solve problems faster more efficiently and by exploring the agent's learning environment much better.

In the first chapters of this thesis, we introduce the reader to the concepts of reinforcement learning with an analytical methodology. Next, we provide the reader with the necessary theoretical background to understand the necessity of using deep reinforcement learning. We explain the main types of deep reinforcement learning. Finally, we introduce the reader to the concepts of asynchronous reinforcement learning.

In this thesis, based on the research [19] we implemented and examined the behavior of the asynchronous algorithms asynchronous one step Sarsa, asynchronous one step Q-learning, asynchronous n-step Q-learning, asynchronous advantage Actor-Critic. We also implemented the deep reinforcement learning algorithms DQN, Double DQN, Dueling DQN. We "ran" the DQN type programs on a graphics card, while the asynchronous reinforcement learning algorithms on a simple CPU. We compared the performance of all reinforcement learning programs on the Cart-pole game.

Keywords

deep reinforcement learning, asynchronous, Asynchronous Advantage Actor-Critic, analysis, implementation, machine learning

Ευχαριστίες

Κατ' αρχάς θα ήθελα να ευχαριστήσω όλο το προσωπικό του εργαστήριο ευφυών υπολογιστικών συστημάτων για το υψηλό επίπεδο γνώσεων και τα καινοτόμα μαθήματα, τα οποία παρέχει. Θέλω να ευχαριστήσω θερμά τον κ. Ανδρέα Σταφυλοπάτη για την υποστήριξή του καθ' όλη τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας. Επίσης ευχαριστώ τους κ. Νεκτάριο Κοζύρη, κ. Γεώργιο Γκούμα και κ. Νικόλαο Παπασπύρου για την πολύτιμη υποστήριξη που μου παρείχαν μετά το πέρας το προπτυχιακών μου σπουδών. Οφείλω να "πω", επίσης, ένα μεγάλο ευχαριστώ σε όλο το εκπαιδευτικό προσωπικό του μεταπτυχιακού της μαθηματικής προτυποποίησης για όσα διδάχτηκα τα τελευταία χρόνια. Ιδιαίτερες ευχαριστίες οφείλω για την υλοποίηση της παρούσας διπλωματικής και στο εργαστήριο Cslab.

Ευχαριστώ ιδιαίτερα τον κ. Γεώργιο Σιόλα για την αμέριστη βοήθειά του και για την πολύτιμη καθοδήγηση που μου προσέφερε για την επιτυχημένη εκπόνηση της παρούσας διπλωματικής εργασίας.

Ευχαριστώ προσωπικά την κ. Σοφία Λαμπροπούλου και τον κ. Νικόλαο Σταυρακάκη για τη συνεργασία που είχανε με όλους τους φοιτητές του μεταπτυχιακού.

Ευχαριστώ ξανά τον κ. Βασίλη Καρακώστα, ο οποίος σε προπτυχιακό επίπεδο με δίδαξε να εκπονώ διπλωματικές εργασίες.

Ευχαριστώ θερμά τη χώρα μου για τις ευκαιρίες μόρφωσης που μου παρείχε και το Εθνικό Μετσόβιο Πολυτεχνείο ειδικότερα.

Ευχαριστώ θερμά τη σχολή Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών που με δέχτηκε στο μεταπτυχιακό πρόγραμμα της μαθηματικής προτυποποίησης.

Τέλος, οφείλω ένα μεγάλο ευχαριστώ στην οικογένειά μου και στους γονείς μου ιδιαίτερα, για τη στήριξή τους καθ' όλη τη διάρκεια των σπουδών μου.

Περιεχόμενα

Ευρετήριο Εικόνων

Κεφάλαιο 1

Εισαγωγή

Στην παρούσα διπλωματική εργασία ασχολούμαστε με την μελέτη και την εφαρμογή ασύγχρονων μεθόδων ενισχυτικής αλλά και βαθιάς ενισχυτικής μάθησης. Έχουμε στηριχθεί σε μεγάλο βαθμό στην έρευνα [19]. Αφού αναλύσουμε τους αλγορίθμους DQN, Double DQN, Dueling DQN, τον ασύγχρονο αλγόριθμο ενός βήματος μάθησης-Q (asynchronous one step Q-learning), τον ασύγχρονο αλγόριθμο ενός βήματος Sarsa (asynchronous one-step Sarsa), τον ασύγχρονο αλγόριθμο n-βημάτων μάθησης-Q (asynchronous n-step Q-learning) και τον ασύγχρονο αλγόριθμο δράστη-κριτή με συνάρτηση πλεονεκτήματος asynchronous advantage actor-critic (A3C). Στη συνέχεια τους υλοποιούμε πάνω στο παιχνίδι Cart-pole για να εξετάσουμε κατά πόσο βελτιστοποιούν οι ασύγχρονοι αλγόριθμοι την απόδοση σε σχέση με τους κλασσικούς αλγορίθμους της βαθιάς ενισχυτικής μάθησης. Με βάση τα πειραματικά αποτελέσματα που λάβαμε αλλά και τα πειραματικά αποτλέσματα της έρευνας [19] επιχειρήσαμε να αναλύσουμε σε βάθος τα πλεονεκτήματα και τα μειονεκτήματα των ασύγχρονων αλγορίθμων ενισχυτικής μάθησης. Στη συνέχεια, συνέχεια προβαίνουμε στα συμπεράσματά μας για τους αλγορίθμους ασύγχρονης ενισχυτικής μάθησης, αλλά και στις προτάσεις μας για μελλοντική έρευνα πάνω στο συγκεκριμένο ερευνητικό πεδίο της τεχνητής νοημοσύνης.

1.1 Δομή Εργασίας

Πιο αναλυτικά, στο πρώτο κεφάλαιο της παρούσας διπλωματικής εργασίας πραγματοποιούμε μία περιγραφή της δομής και του περιεχομένου της παρούσας διπλωματικής εργασίας. Στο δεύτερο κεφάλαιο υλοποιούμε την ίδια περιγραφή της δομής και του περιεχομένου της εραγασίας μας στην αγγλική γλώσσα.

Από το τρίτο κεφάλαιο και μετά πραγματοποιούμε εισαγωγή στο κυρίως περιεχόμενο της εργασίας μας. Πιο συγκεκριμένα, στο τρίτο κεφάλαιο εισά-

γουμε τον αναγνώστη στα τρία είδη μηχνανικής μάθησης, στην επιβλεπόμενη , στη μη επιβλεπόμενη και στην ενισχυτική μάθηση. Στη συνέχεια του ίδιου κεφαλαίου αναλύουμε την έννοια του μαρκοβιανού συστήματος αποφάσεων. Στη συνέχεια, επίσης, του ίδιου κεφαλαίου περιγράφουμε τις μεθόδους δυναμικού προγραμματισμού, τι μεθόδους Monte Carlo, καθώς και τις μεθόδους εκμάθησης Temporal-Difference (TD). Στο τέταρτο κεφάλαιο συνεχίζουμε την εισαγωγή του αναγνώστη πάνω στις έννοιες της βαθιάς ενισχυτικής μάθησης. Εισάγουμε τον αναγνώστη πανω στη βαθιά ενισχυτική μάθηση. Αρχικά εισάγουμε τον αναγνώστη πανω στις value-based κλάσεις αλγορίθμων. Εστιάζουμε στους αλγορίθμους DQN, Double DQN και Dueling DQN. Στη συνέχεια του τετάρτου κεφαλαίου, αναλύουμε τις policy-based κλάσεις αλγορίθμων βαθιάς ενισχυτικής μάθησης. Συνδυασμός των value-based και policy-based αποτελεί ο αλγόριθμος A3C, τον οποίο και αναλύουμε διεξοδικά στο ίδιο κεφάλαιο. Τέλος, στο ίδιο κεφάλαιο πραγματοποιούμε και μία μικρή ενημερωτική αναφορά στις model-based μεθόδους ενισχυτικής μάθησης, τις οποίες, όμως, δε χρησιμοποιούμε στην παρούσα εργασία.

Στο πέμπτο κεφάλαιο περιγράφουμε κάποια κεφάλαια από την έρευνα [19]. Εστιάζουμε στους ασύγχρονους αλγόριθμους που περιγράφει η συγκεκριμένη έρευνα. Οι αλγόριθμοι αυτοί είναι ο ασύγχρονος αλγόριθμος ενός βήματος μάθησης-Q, ο ασύγχρονος αλγόριθμος ενός βήματος Sarsa, ο ασύγχρονος αλγόριθμο n-βημάτων μάθησης-Q και ο A3C. Αφού περιγράψουμε τους συγκεκριμένους αλγόριθμους στη συνέχεια αναφερόμαστε στα διαδικαστικά των πειραμάτικών μεθόδων μας. Υλοποιήσαμε και εκπαιδεύσαμε τους αλγόριθμους αλγορίθμους DQN, Double DQN, Dueling dqn, τον ασύγχρονο αλγόριθμο ενός βήματος μάθησης-Q, τον ασύγχρονο αλγόριθμο ενός βήματος Sarsa, τον ασύγχρονο αλγόριθμο n-βημάτων μάθησης-Q και τον αλγόριθμο A3C πάνω στο σχετικά απλό παιχνίδι Cart-pole για να εξετάσουμε κατά πόσο βελτιστοποιούν οι ασύγχρονοι αλγόριθμοι την απόδοση σε σχέση με τους κλασσικούς αλγορίθμους ενισχυτικής μάθησης.

Στο έκτο κεφάλαιο της παρούσας διπλωματικής, περιγράφουμε τα πειραματικά αποτελέσματα των μετρήσεών μας. Συγκρίνουμε την απόδοση των αλγορίθμων DQN, Double DQN Dueling DQN, ασύγχρονος αλγόριθμος ενός βήματος μάθησης-Q, ασύγχρονος αλγόριθμος ενός βήματος Sarsa, ασύγχρονος αλγόριθμο n-βημάτων μάθησης-Q και A3C. Επίσης, εξετάζουμε την απόδοση του αλγορίθμου A3C αλλάζοντας τις παραμέτρους εκμάθησής του. Σημειώνουμε ότι όλα τα πραγράμματα τα έχουμε αναπτύξει σε γλώσσα python, ενώ επίσης με χρήση γλώσσας python οπτικοποιούμε τα πειραματικά αποτελέσματα. Τέλος στο κεφάλαιο επτά προβαίνουμε στα συμπεράσματά μας για τους αλγορίθμους ασύγχρονης ενισχυτικής μάθησης, αλλά και στις προτάσεις μας για μελλοντική έρευνα πάνω στο συγκεκριμένο ερευνητικό πεδίο της τεχνητής νοημοσύνης.

1.2 Κίνητρο

Η βαθιά ενίσχυτική μάθηση αποτελεί το συνδυασμό ενίσχυτικής μάθησης (RL) και βαθιάς μηχανικής μάθηση. Αυτό το πεδίο έρευνας κατάφερε να επιλύσει ένα ευρύ φάσμα σύνθετων αποφάσεων-προβλημάτων ολοκληρώνοντας εργασίες που προηγουμένως δεν ήταν εφικτές για ένα μηχάνημα. Με αυτόν τον τρόπο, η βαθιά ενισχυτική μάθηση ανοίγει πολλές νέες εφαρμογές σε τομείς, όπως η υγειονομική περίθαλψη, η ρομποτική, τα έξυπνα δίκτυα, τα χρηματοοικονομικά καθώς και πολλούς άλλους. Παρά το γεγονός ότι η ενισχυτική μάθηση είχε πολλές επιτυχίες στο παρελθόν, εμφανίζει πολλά εγγενή προβλήματα, κάποια από τα οποία δε τα λύνει αποδοτικά ούτε η βαθιά ενισχυτική μάθηση. Σε αυτό το σημείο λύση στο πρόβλημα επιχειρούμε να δώσουμε μέσω της ασύγχρονης ενισχυτικής μάθησης και κυρίως της ασύγχρονης βαθιάς ενισχυτικής μάθησης. Μας ενδιαφέρει να κατανοήσουμε σε τι βαθμό βελτιώνουν την απόδοση του πράκτορα οι νέοι ασύγχρονοι αλγόριθμοι, καθώς και κατά πόσο χρόνο μας εξοικονομούν σε σχέση με του κλασσικούς μη κατανεμημένους αλγόριθμους.

Κεφάλαιο 2

Introduction

In this thesis, we deal with the study and application of asynchronous reinforcement and deep reinforcement learning methods. We have relied heavily on Asynchronous Methods for Deep Reinforcement Learning research. After analyzing dqn, double dqn, dueling dqn, asynchronous one step Q-learning, asynchronous one-step Sarsa, asynchronous n-step Q-learning and asynchronous advantage actor-critic algorithms (A3C). We then implement them on the Cart-pole game to examine whether asynchronous algorithms optimize performance over classical deep reinforcement learning algorithms. Based on the experimental results we received as well as the experimental results of the [19] research we attempted to analyze in depth the advantages and disadvantages of asynchronous reinforcement learning algorithms. Then, we proceed to our conclusions about the asynchronous reinforcement learning algorithms, but also to our proposals for future research on the specific research field of artificial intelligence.

2.1 Thesis Structure

In more detail, in the first chapter of this thesis we provide a description of the structure and content of this thesis. In the second chapter we implement the same description of the structure and content of our essay in the english language.

From the third chapter onwards we introduce the main content of our work. More specifically, in the third chapter we introduce the reader to the three types of machine learning, supervised, unsupervised and reinforcement learning. Later in the same chapter we analyze the concept of the Markovian decision system. Also later in the same chapter we describe dynamic programming methods, Monte Carlo methods, as well as Temporal-Difference

(TD) learning methods. In the fourth chapter we continue to introduce the reader to the concepts of deep reinforcement learning. We introduce the reader above to deep reinforcement learning. First we introduce the reader above to the value-based classes of algorithms. We focus on dqn, double and dueling dqn algorithms. In the continuation of the fourth chapter, we analyze the policy-based classes of deep reinforcement learning algorithms. A combination of value-based and policy-based is the asynchronous advantage actor-critic algorithm, which we analyze thoroughly in the same chapter. Finally, in the same chapter we make an informative reference to model-based reinforcement learning methods, which, however, we do not use in this work.

In the fifth chapter we describe some chapter from the [19] research. We focus on the asynchronous algorithms that this research describes. We refer to the asynchronous one step q learning, asynchronous one-step Sarsa, asynchronous n-step Q-learning and A3C algorithms. After describing the specific algorithms, we then refer to the procedures of our experimental methods. We implemented and trained the dqn, duel dqn, asynchronous one step q learning, asynchronous one-step Sarsa, asynchronous n-step Q-learning and A3C algorithms on the Cart-pole game to examine whether they optimize asynchronous algorithms performance over classical reinforcement learning algorithms.

In the sixth chapter of this dissertation, we describe the experimental results of our measurements. We compare the performance of dqn, duel dqn, asynchronous one step Q-learning, asynchronous one-step Sarsa, asynchronous n-step Q-learning and A3C algorithms. We also examine the performance of the A3C algorithm by varying its learning parameters. We note that we have developed all the programs in python language. We use, also, python language to visualize the experimental results. Finally, in chapter seven we proceed to our conclusions about the asynchronous reinforcement learning algorithms, but also to our proposals for future research on the specific research field of artificial intelligence.

2.2 Motivation

Deep reinforcement learning is the combination of reinforcement learning (RL) and deep learning. This field of research has succeeded in solving a wide range of complex decision-problems by completing tasks previously unfeasible for a machine. In this way, deep reinforcement learning opens up many new applications in areas such as healthcare, robotics, smart networks, finance and many others. Despite the fact that reinforcement learning has had many successes in the past, it exhibits many inherent problems, some of which

even deep reinforcement learning does not solve efficiently. At this point we are trying to provide a solution to the problem through asynchronous reinforcement learning and especially asynchronous deep reinforcement learning. We are interested in understanding how much the new asynchronous algorithms improve the agent's performance, as well as how much time they save compared to the classical non-distributed algorithms.

Κεφάλαιο 3

Θεωρητικό Υπόβαθρο και κίνητρο

3.1 Μηχανική Μάθηση

Με τον όρο μηχανική μάθηση εννοούμε την επιστημονική μελέτη αλγορίθμων υπολογιστών, οι οποίοι επιχειρούν να επιλύσουν ένα ζητημα χωρίς τη χρήση προκαθορισμένων οδηγιών, αλλά βασιζόμενοι σε μοτίβα και τεκμήρια. Οι αλγόριθμοι μηχανικής μάθησης χτίζουν ένα μοντέλο βασιζόμενοι στα δεδομένα εκπαίδευσης (training data), με στόχο το υπολογιστικό σύστημα να πραγματοποιήσει προβλέψεις ή να λάβει αποφάσεις πάνω σε θέματα στα οποία δεν έχουν ρητά προγραμματιστεί. Η μηχανική μάθηση αποτελεί κλάδο της τεχνητής νοημοσύνης.[3]

3.1.1 Κατηγορίες Μηχανικής Μάθησης

Η μηχανική μάθηση χωρίζεται σε τρεις ευρείες κατηγορίες:

Επιβλεπόμενη μάθηση (Supervised Learning): Η επιβλεπόμενη μάθηση αποτελεί μία κατηγορία μηχανικής μάθησης στην οποία επιχειρείται να εκπαιδευτεί μία συνάρτηση μάθησης ώστε να αντιστοιχίζει την είσοδο με την έξοδο, βασιζόμενη κάποια ήδη υπάρχοντα δείγματα εισόδου-εξόδου. Στην επιβλεπόμενη μάθηση κάθε παράδειγμα αποτελείται από ένα ζευγάρι, το οποίο περιέχει ένα αντικείμνο εισόδου (συνήθως ένα διάνυσμα) και ένα αντικείμενο επιθυμητής εξόδου (λέγεται και supervisory signal). Ένας αλγόριθμος επιβλεπόμενης μηχανικής μάθησης αναλύει τα δεδομένα εκπαίδευσης (training data) και παράγει μία "εκπαιδευμένη" συνάρτηση, η οποία μπορεί να χρησιμοποιηθεί ως συνάρτηση εισόδου για νέα παραδείγματα και να προβλέψει τις αντίστοιχες εξόδους. Ιδανικά αλγόριθμος θα μπορεί να προβλέψει σωστά τις ετικέτες κλάσης για άγνωστες εισόδους. Γι' αυτόν τον λόγο είναι απαραίτητο

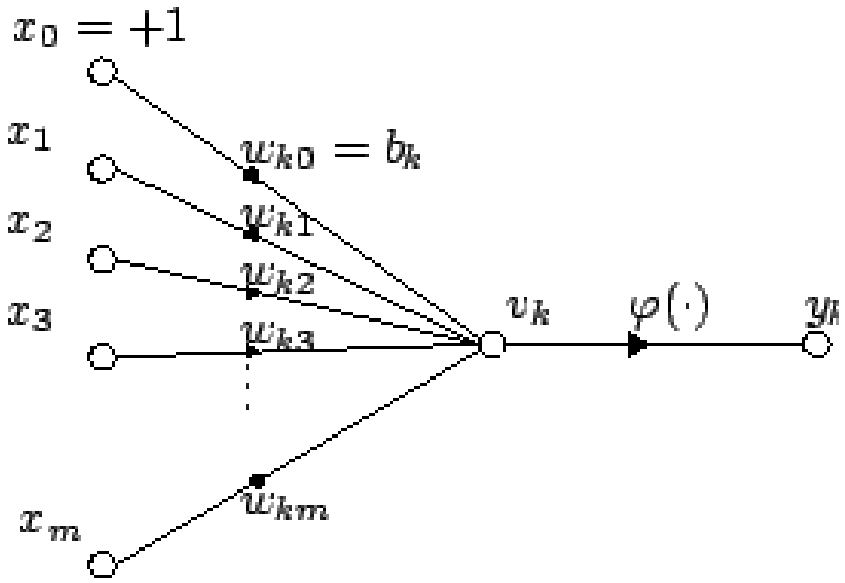
ο αλγόριθμος εκμάθησης να γενικεύεται από τα δεδομένα εκπαίδευσης σε άγνωστες καταστάσεις με "λογικό" τρόπο.[5]

Μη-επιβλεπόμενη μάθηση (Unsupervised Learning): Η μη-επιβλεπόμενη μάθηση αποτελεί ένα είδος αλγορίθμων, οι οποίοι αναγνωρίζουν από δεδομένα χωρίς ετικέτες (unlabaled data). Το ενδιαφέρον είναι ότι μέσω της μίμησης, η μηχανή αναγκάζεται να χτίσει μια συμπαγή εσωτερική αναπαράσταση του κόσμου. Σε αντίθεση με την εποπτευόμενη μάθηση (Supervised Learning) όπου τα δεδομένα επισημαίνονται από έναν άνθρωπο, π.χ. ως "αυτοκίνητο" ή ως "ψάρι" κ.λπ., οι αλγόριθμοι μη-επιβλεπόμενης μάθησης εμφανίζουν αυτοοργάνωση που αναγνωρίζει μοτίβα ως νευρωνικές προεπιλογές ή πυκνότητες πιθανότητας. Γενικότερα μπορούμε να πούμε ότι στόχος της είναι στόχος της οποίας είναι η ανακάλυψη πιθανής δομής που μπορεί να κρύβεται πίσω από μη χαρακτηρισμένα δεδομένα. Γι' αυτόν το λόγο τα δεδομένα που εκπαίδευσης που παρέχονται σε αλγορίθμους αυτής της κατηγορίας πρέπει να μην περιέχουν λάθη και να έχουν περιορισμένο θόρυβο. [7] (θέλει δουλειά πάλι)

Ενισχυτική Μάθηση (Reinforcement Learning): Η ενισχυτική μάθηση είναι ένας τομέας της μηχανικής μάθησης που ασχολείται με τον τρόπο με τον οποίο οι ευφυείς πράκτορες πρέπει να κάνουν ενέργειες σε ένα περιβάλλον προκειμένου να μεγιστοποιήσουν την τιμή της συσωρευτικής ανταμοιβής. Θα αναφερθούμε πιο διεξοδικά στην ενισχυτική μάθηση σε θεωρητικό επίπεδο στο επόμενο υποκεφάλαιο, αλλά και σε όλο το υπόλοιπο κείμενο της παρούσας διπλωματικής. Αποτελεί, άλλωστε, το βασικό θέμα της.

3.2 Τεχνητά Νευρωνικά Δίκτυα Artificial neural networks

Τα τεχνητά νευρωνικά δίκτυα (Artificial neural networks-ANN), είναι υπολογιστικά συστήματα εμπνευσμένα από την αρχιτεκτονική των βιολογικών νευρώνων, όπως ο ανθρώπινος εγκεφάλους. Ένα τεχνητό νευρωνικό δίκτυο βασίζεται σε μια συλλογή συνδεδεμένων μονάδων ή κόμβων που ονομάζονται τεχνητοί νευρώνες, οι οποίοι μοντελοποιούν νευρώνες σε έναν εγκέφαλο. Κάθε σύνδεση, όπως οι συνάψεις σε έναν βιολογικό εγκέφαλο, μπορεί να μεταδώσει ένα σήμα σε άλλους νευρώνες. Ένας τεχνητός νευρώνας που λαμβάνει ένα σήμα στη συνέχεια το επεξεργάζεται και μπορεί να σηματοδοτήσει νευρώνες που συνδέονται με αυτό. Το "σήμα" σε μια σύνδεση είναι ένας πραγματικός αριθμός και η έξοδος κάθε νευρώνα υπολογίζεται από κάποια μη γραμμική συνάρτηση του αθροίσματος των εισόδων του. Οι συνδέσεις ονομάζονται ακμές. Οι νευρώνες και οι ακμές έχουν συνήθως ένα βάρος που προσαρμόζεται καθώς προχωρά η μάθηση. Το βάρος αυξάνει ή μειώνει την ισχύ του σήματος σε

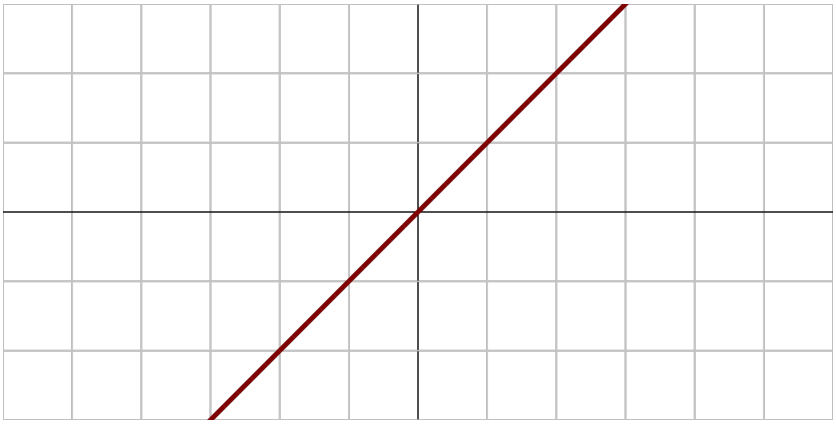


Εικόνα 3.1: Δομή τεχνητού νευρώνα

μια σύνδεση. Οι νευρώνες μπορεί να έχουν ένα κατώφλι έτσι ώστε ένα σήμα να αποστέλλεται μόνο εάν το συνολικό σήμα διασχίζει αυτό το όριο. Συνήθως, οι νευρώνες κατανέμονται σε στρώματα. Διαφορετικά επίπεδα μπορεί να εκτελούν διαφορετικούς μετασχηματισμούς στις εισόδους τους. Τα σήματα μετακινούνται από το πρώτο στρώμα (το στρώμα εισόδου), στο τελευταίο στρώμα (το επίπεδο εξόδου).[1]

3.2.1 Τεχνητός Νευρώνας

Ο τεχνητός νευρώνας, αποτελεί δομικό συστατικό ενός νευρωνικού δικτύου, και προσπαθεί ναπροσομοιώσει την λειτουργία ενός βιολογικού νευρώνα του ανθρωπίνου εγκεφάλου.[2] Οι τεχνητοί νευρώνες είναι στοιχειώδεις μονάδες σε ένα τεχνητό νευρωνικό δίκτυο. Ο τεχνητός νευρώνας λαμβάνει μία ή περισσότερες εισόδους και τους αθροίζει για να παράγει έξοδο. Συνήθως κάθε είσοδος σταθμίζεται ξεχωριστά και το άθροισμα περνά από μια μη γραμμική συνάρτηση γνωστή ως συνάρτηση ενεργοποίησης. Οι συναρτήσεις μεταφοράς συνήθως έχουν σιγμοειδές σχήμα, αλλά μπορεί επίσης να έχουν τη μορφή άλλων μη γραμμικών συναρτήσεων, γραμμικών κατά τμήματα συναρτήσεων, ή βηματικής συνάρτησης. Επίσης, αυξάνονται μονοτονικά, συνεχείς, διαφοροποιούνται και οριοθετούνται. Πιο αναλυτικά παρουσιάζουμε τη δομή του τεχνητού νευρώνα:



Εικόνα 3.2: identity activation function

3.2.2 Συναρτήσεις ενεργοποίησης

Οι συναρτήσεις ενεργοποίησης, γνωστές και ως συναρτήσεις μεταφοράς, χρησιμοποιούνται για την αντιστοίχηση κόμβων εισόδου σε κόμβους εξόδου με συγκεκριμένο τρόπο. Χρησιμοποιούνται για να προσδώσουν μη γραμμικότητα στο δίκτυο. Υπάρχουν πολλές συναρτήσεις ενεργοποίησης που χρησιμοποιούνται στη μηχανική μάθηση. Παραθέτουμε κάποιες από αυτές τις συναρτήσεις ενεργοποίησης:

Identity:

$$f(x) = x$$

Binary Step:

$$f(x) = 0 if x < 0 else f(x) = 1$$

Logistic

$$f(x) = 1/(1 + e^{-x})$$

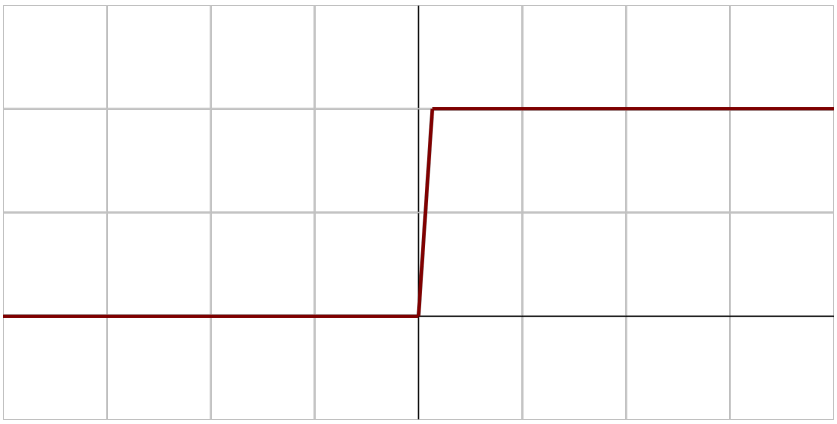
tanh

$$f(x) = tanh(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x})$$

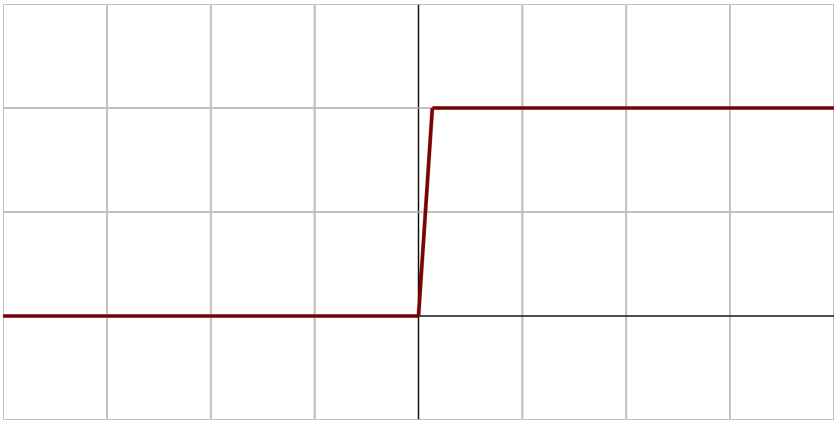
Relu

$$f(x) = \begin{cases} 1 & 0 \geq x \\ 0 & 0 > x \end{cases}$$

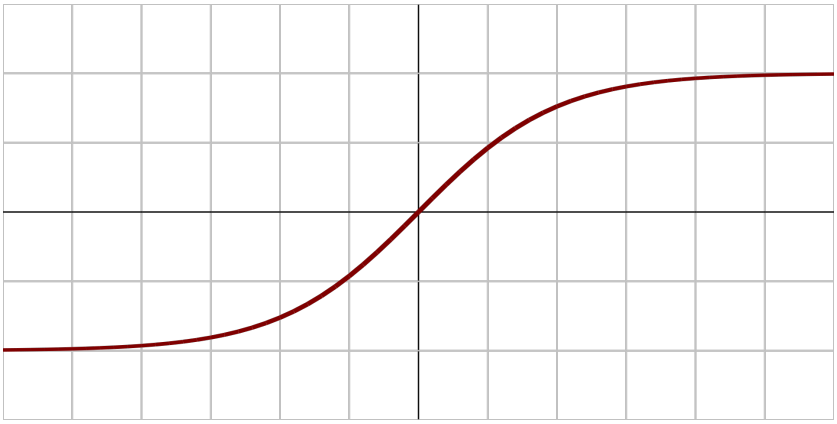
Softplus



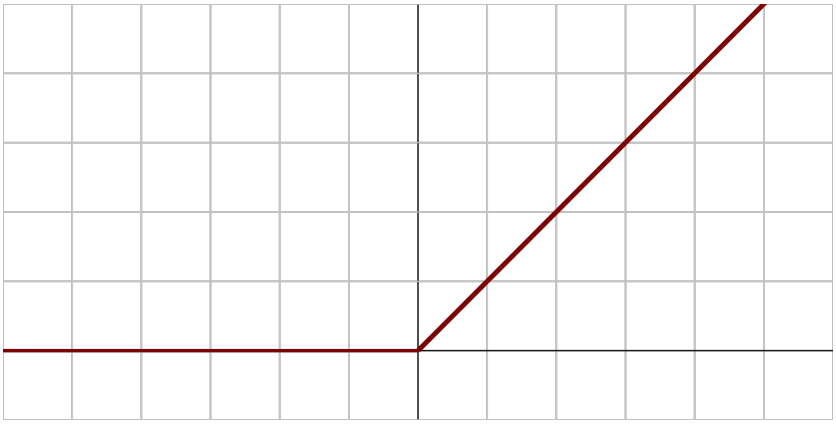
Εικόνα 3.3: Binary Step activation function



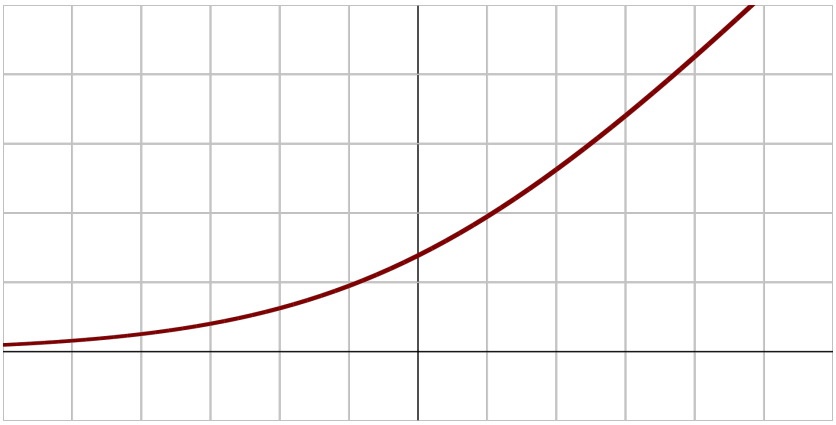
Εικόνα 3.4: Logistic activation function



Εικόνα 3.5: tanh activation function



Εικόνα 3.6: Relu activation function



Εικόνα 3.7: Softplus activation function

$$f(x) = ln(1 + e^x)$$

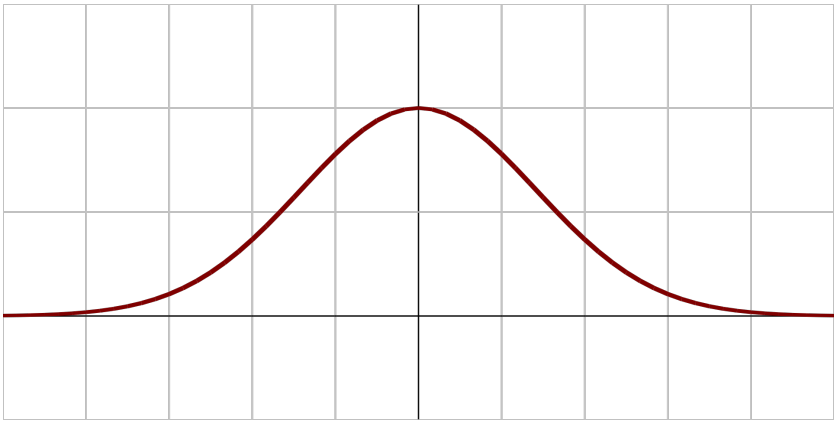
Gaussian

$$f(x) = e^{-x^2}$$

Βασικά χαρακτηριστικά συναρτήσεων ενεργοποίησης:

1. Μη γραμμικότητα

Ο σκοπός της συνάρτησης ενεργοποίησης είναι να εισαγάγει τη μη γραμμικότητα στο δίκτυο. Η μη γραμμικότητα με τη σειρά της μας επιτρέπει να μοντελοποιήσουμε μια μεταβλητή απόκρισης-εξόδου (ονομάζεται συχνά και ως μεταβλητή στόχου ή και ετικέτα κλάσης) που διαφέρει μη γραμμικά με τις επεξηγηματικές μεταβλητές της (είσοδοι). Υπενθυμίζουμε ότι η μη γραμμικότητα



Εικόνα 3.8: Gaussian activation function

σημαίνει ότι η έξοδος δεν μπορεί να αναπαραχθεί από έναν γραμμικό συνδυασμό των εισόδων.

2.Συνεχώς διαφορίσιμες συναρτήσεις

Αυτή η ιδιότητα είναι απαραίτητη για την ενεργοποίηση των gradient-descent μεθόδων βελτιστοποίησης.

3.Εύρος τιμών συνάρτησης

Όταν το εύρος τιμών της συνάρτησης τιμών είναι πεπερασμένο, τότε οι gradient base training μέθοδοι είναι πιο ευσταθείς. Αντίθετα όταν το εύρος τιμών της συνάρτησης τιμών είναι άπειρο, η εκπαίδευση των βαρών του νευρώνα είναι πιο αποτελεσματική. Σε τέτοιες περιπτώσεις, όπου η συνάρτηση ενεργοποίησης εμφανίζει άπειρο εύρος τιμών, οι απεικονίσεις των εισόδων επηρεάζουν όλα τα βάρη του νευρώνα. Γι' αυτόν το λόγο κατά την εκπαίδευση δικτύων με συναρτήσεις ενεργοποίησης με άπειρα εύρη τιμών, ρυθμίζουμε τις μεταβλητές εκμάθησης να λαμβάνουν χαμηλές τιμές.

3.2.3 Multi-Layer Perceptron

Η χρήση αλγορίθμων της κατηγορίας artificial neural networks είναι ιδιαίτερα δημοφιλής σήμερα. Χρησιμοποιούνται κυρίως για την πρόβλεψη τιμής στόχου μη γραμμικών προβλημάτων. Οι συσχετίσεις που συμπεραίνονται από τα συγκεκριμένα δίκτυα θεωρούνται ιδιαίτερα περίπλοκες. Θα περιγράψουμε σε αυτό το σημείο τη δομή ενός multilayer perceptron (fully feedforward ANN). Το νευρωνικό δίκτυο δέχεται ως είσοδο ένα διάνυσμα N διαστάσεων $x_1, ..., x_n$. Από το επίπεδο εισόδου κάθε x_i συνδέεται με κάθε μονάδα του πρώτου κρυφού επιπέδου. Κάθε είσοδος x_i πολλαπλασιάζεται με το αντίστοιχο βάρος $w_{i,j}$ πριν την είσοδό του στην αντίστοιχη μονάδα j του πρώτου κρυφού επιπέδου. Σε κάθε μονάδα του πρώτου κρυφού επιπέδου παράγεται η τιμή:

$$z_j = (x_1 w_{1,j} + ... + x_n w_{n,j})$$

Η παραγόμενη τιμή εισέρχεται ως είσοδος σε μία συνάρτηση ενεργοποίησης, όπως αυτές που αναφέραμε σε προηγούμενο υποκεφάλαιο.

Σε κάθε κρυφό επίπεδο L το output της συνάρτησης ενεργοποίησης εισέρχεται ως είσοδος σε κάθε μονάδα του επιπέδου (L+1), πολλαπλασιαζόμενο πάντα από το αντίστοιχο βάρος. Το αποτέλεσμα βασίζεται στις εξόδους των μονάδων του τελευταίου επιπέδου, χωρίς να περνάει απαραίτητα από την ίδια συνάρτηση ενεργοποίησης. Κριτήριο επιτυχίας θεωρείται το αποτέλεσμα της συνάρτησης απωλειών κατά την έξοδο.

Για την εκπαίδευση η ίδια διαδικασία επαναλαμβάνεται για κάθε είσοδο, αφού ανανεωθούν οι τιμές των βαρών από την έξοδο προς την είσοδο (Back-Propagation Algorithm). Τα επιμέρους βάρη ανανεώνονται με βάση μία συγκεκριμένη συνάρτηση. Η μερική παράγωγος της συνάρτησης απωλειών, ως προς το κάθε βάρος ξεχωριστά, πολλαπλασιάζεται με την υπερπαράμετρο α και προστίθεται στην παλαιά τιμή του βάρους, ώστε να προκύψει το νέο ανανεωμένο βάρος.

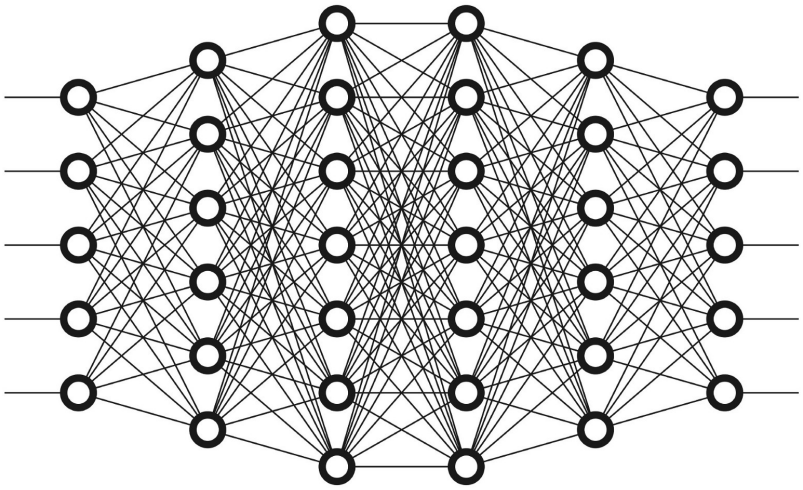
Παραθέτουμε τη σχέση αναλυτικά:

$$w_{i,j_{new}} = w_{i,j} + a * \frac{\partial J(w_{i,j})}{\partial w_{i,j}}$$

όπου J η συνάρτηση απωλειών.

Η διαδικασία σταματάει όταν υπάρξει σύγκλιση, δηλαδή όταν το αποτέλεσμα της συνάρτησης απωλειών, κατά την έξοδο είναι μηδέν, ή όταν η έξοδος της συνάρτησης απωλειών αρχίζει σταθερά να αυξάνεται. Οι υπερπαράμετροι του δικτύου είναι το πλήθος των μονάδων κάθε επιπέδου, το πλήθος των επιπέδων, η συνάρτηση ενεργοποίησης, η συνάρτηση απωλειών, ο αριθμός των δειγμάτων εισόδου σε κάθε επανάληψη (batch size), το learning rate και οι όροι ομαλοποίησης (regularazation terms).

Το δίκτυο Multi-Layer Perceptron θεωρείται ένα είδος ANN. Περιέχει τουλάχιστον τρία επίπεδα κόμβων, το επίπεδο εισόδου, το επίπεδο εξόδου και τουλάχιστον ένα ενδιάμεσο επίπεδο. Χρησιμοποιεί μη γραμμική συνάρτηση ενεργοποίησης σε κάθε κόμβο. Για την εκπαίδευση του δείγματος χρησιμοποιείται η τεχνική backpropagation, την οποία περιγράψαμε στις προηγούμενες παραγράφους.[39] Για καλύτερη κατανόηση έχουμε παρθέσει και μία εικόνα ενός Multi-Layer Perceptron, βλέπε εικόνα 3.9.



Εικόνα 3.9: Δομή Multi-Layer Perceptron

3.2.4 Overfitting και Underfitting στα Τεχνητά Νευρωνικά Δίκτυα

Με την έννοια της υπερεκπαίδευσης (overfitting) αναφερόμαστε στην περίπτωση όπου ένα μοντέλο μηχανικής μάθησης δεν μπορεί να γενικεύσει και να πραγματοποιήσει προβλέψεις με υψηλή ακρίβεια σε ένα αγνωστο σύνολο δεδομένων, ενώ στο σύνολο πάνω στο οποίο έχει εκπαιδεύτεί εμφανίζει πολύ υψηλά ποσοστά επιτυχίας. Ένα σαφές σημάδι της υπερβολικής προσαρμογής ενός μοντέλου μηχανικής μάθησης πάνω σε ένα σύνολο δεδομένων, είναι όταν η ακρίβεια πρόβλεψεων του μοντέλου πάνω στο testing ή validation dataset είναι πολύ χαμηλότερη από την αντίστοιχη στο σύνολο δεδομένων εκπαίδευσης.

Στη στατιστική με τον όρο υπερεκπαίδευση overfitting αναφερόμαστε στο σφάλμα μοντελοποίησης που εμφανίζεται όταν μια συνάρτηση έχει προσαρμόσει τις παραμέτρους της πολύ κοντά σε ένα σύνολο δεδομένων. Η υπερεκπαίδευση της συνάρτησης ενδέχεται να επηρεάσει αρνητικά την ακρίβεια προβλέψεών της σε μελλοντικές παρατηρήσεις.

Η υποεκπαίδευση (underfitting) αναφέρεται σε ένα μοντέλο που δεν μπορεί να μοντελοποιήσει το σύνολο δεδομένων εκπαίδευσης του ούτε μπορεί να γενικεύσει σε νέο σύνολο δεδομένων. Ένα υποεκπαιδευμένο μοντέλο μηχανικής μάθησης δεν είναι κατάλληλο μοντέλο και θα έχει κακή απόδοση προβλέψεων ακόμα και στο σύνολο δεδομένων που έχει εκπαίδευτεί.[6]

3.3 Ενισχυτική Μάθηση

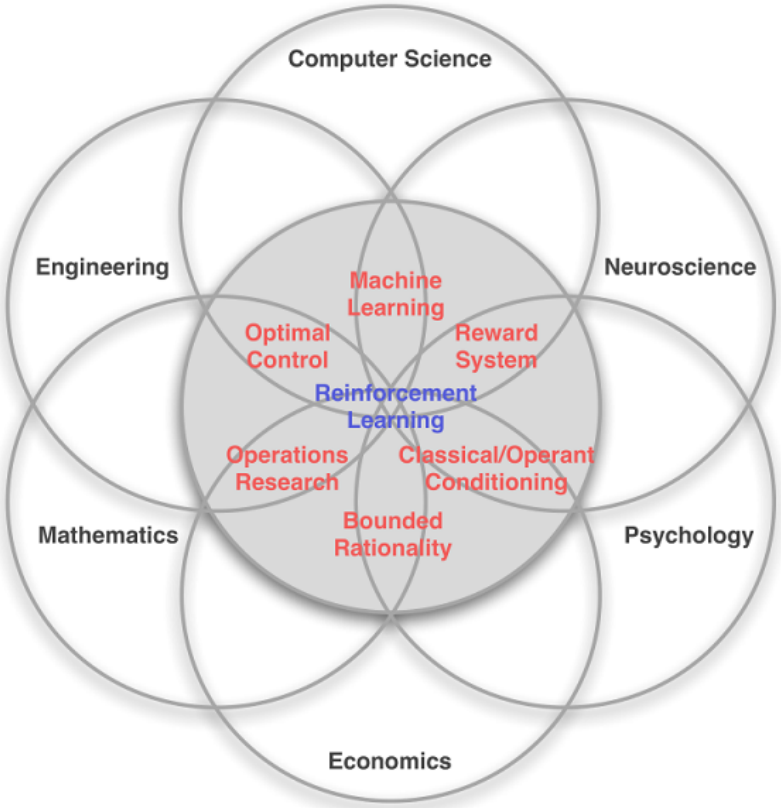
3.3.1 Εισαγωγή

Η ιδέα της εκμάθησης του ανθρώπου μέσω της αλληλεπίδρασης με το περιβάλλον του είναι η πρώτη που μας έρχεται στο μυαλό όταν αναλογιζόμαστε για τη φύση της εκμάθησης. Όταν ένα νεογνό παίζει, κινεί τους ώμους του ή παρατηρεί το περιβάλλον, δεν έχει κάποιο άμεσο δάσκαλο να το καθοδηγήσει, αλλά διαθέτει συνδέσεις με το περιβάλλον του μέσω των φυσικών αισθητηρίων του. Με την εκπαίδευση των συνδέσεων παράγεται μία πληθώρα πληροφοριών για τη σχέση αιτίου-αποτελέσματος, για τις συνέπειες κάθε δράσεως, καθώς και για τη σειρά με την οποία πρέπει να δράσει ο άνθρωπος (σειρά δράσεων) για να πετύχει το στόχο του. Στην καθημερινή ζωή του τέτοιου είδους αλληλεπιδράσεις αποτελούν βασική πηγή γνώσης του ανθρώπου για το περιβάλλον και για τον ίδιο του τον εαυτό. Όταν επιχειρούμε να μάθουμε οδήγηση είτε να εξασκήσουμε κάποια κοινωνική συμπεριφορά (π.χ. να βελτιωθούμε ως συνομιλητές), τότε είμαστε ενήμεροι εάν το περιβάλλον ανταποκρίνεται στις δράσεις μας. Η ιδέα της εκμάθησης μέσω της αλληλεπίδρασης αποτελεί θεμελιώδη μέθοδο εκμάθησης για το ανθρώπινο είδος και όχι μόνο.

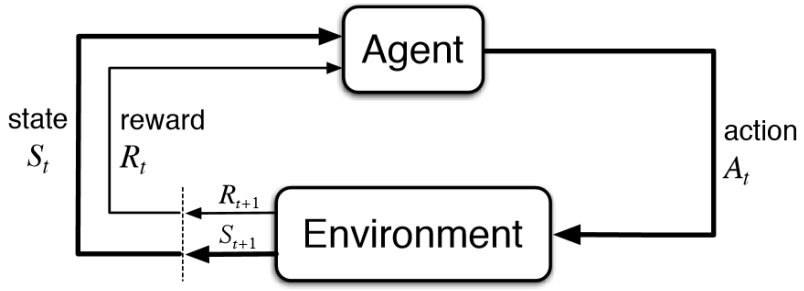
Στη συνέχεια του υποκεφαλαίου, αλλά και στα επόμενα κεφάλαια, θα προσεγγίσουμε υπολογιστικά τις μεθόδους εκμάθησης μέσω αλληλεπίδρασης. Αντί να επιχειρήσουμε να μοντελοποιήσουμε με ποιο τρόπο μαθαίνει ο άνθρωπος οι ερευνητές ακολούθησαν μία διαφορετική προσέγγιση. Πάνω σε ιδανικά περιβάλλοντα μάθησης εκτιμάται η απόδοση διαφόρων μεθόδων μάθησης, οι οποίες υποκαθιστούν τη φυσική εκμάθηση μέσω αλληλεπίδρασης. Η ενισχυτική μάθηση αποτελεί πεδίο της μηχανικής μάθησης, το οποίο ασχολείται με το πώς αλγοριθμικοί πράκτορες θα πρέπει να πράττουν σε ένα περιβάλλον, με στόχο να μεγιστοποιούν κάποιο κέρδος.[37] Η συγκεκριμένη προσέγγιση της εκμάθησης μέσω της εξερεύνησης, ονομάζεται ενισχυτική μάθηση. Επικεντρώνεται περισσότερο στην goal-directed μάθηση από οποιαδήποτε άλλη μέθοδο μηχανικής μάθησης. Το πεδίο της ενισχυτικής μάθησης αποτελεί συνδυασμό πολλών επιστημονικών τομέων, όπως φαίνεται και στην εικόνα 3.10.[27]

3.3.2 Μοντελοποίηση του προβλήματος

Στη standard μέθοδο ενισχυτικής μάθησης ένας πράκτορας είναι συνδεδεμένος με το περιβάλλον μέσω της κατάστασης που βρίσκεται και των δράσεων που μπορεί να πραγματοποιήσει. Σε κάθε βήμα αυτής της διαδικασίας αλληλεπίδρασης ο πράκτορας λαμβάνει μία είσοδο i, γνωρίζει σε ποια κατάσταση, s, βρίσκεται και διαλέγει μία δράση a, η οποία και αποτελεί την έξοδο (output). Η δράση αυτή αλλάζει την κατάσταση, s, του πράκτορα και η ανταμοιβή αυτής



Εικόνα 3.10: Επιστημονικά πεδία που συνθέτουν τις έννοιες της ενισχυτικής μάθησης [26]



Εικόνα 3.11: Βήμα ενισχυτικής μάθησης

της μετάβασης στέλνεται στον πράκτορα, μέσω σήματος ενίσχυσης. Στόχος του πράκτορα είναι η επιλογή πράξεων που στοχεύουν στην αύξηση του μακροπρόθεσμου αθροίσματος των ανταμοιβών. Η εκπαίδευση επιτυγχάνεται μέσω προσπάθειας - λάθους για αρκετά βήματα, κατευθυνόμενη από διάφορους αλγόριθμους και μεθόδους που θα αναλυθούν στη συνέχεια.

Πιο αναλυτικά, ο πράκτορας και το περιβάλλον αλληλεπιδρούν, σε κάθε βήμα από μία ακολουθία διακριτών βημάτων t=0,1,2...n . Σε σε κάθε βήμα ο πράκτορας λαμβάνει αναπαράσταση της κατάστασης του περιβάλλοντος$S_t \in S$, όπου με S συμβολίζουμε το σύνολο δυνατών καταστάσεων. Με βάση τη συγκεκριμένη κατάσταση S_t επιλέγει τη δράση που θα ακολουθήσει $A_t(S_t) \in A$, όπου $A_t(S_t)$ είναι το σύνολο των δράσεων που μπορεί να ακολουθήσει ο πράκτορας όταν βρίσκεται στην κατάσταση S_t. Ένα βήμα αργότερα, σα συνέπεια της πράξης του ο πράκτορας λαμβάνει μία αριθμητική ανταμοιβή, $R_{t+1} \in R$ και $R \subset \mathbf{R}$ και μεταφέρεται στην κατάσταση S_{t+1}. Στην εικόνα 3.11 παρουσιάζεται η συγκεκριμενή διαδικασία.

Σε κάθε χρονικό βήμα, ο πράκτορας πραγματοποιεί αντιστοίχιση από την κατάσταση που βρίσκεται κάθε φορά στην πιθανότητα να επιλέξει την κάθε διαθέσιμη δράση-επιλογή. Αυτή η αντιστοίχιση ονομάζεται ως "πολιτική" του πράκτορα και συμβολίζεται ως π_t, όπου $\pi(s|a)$ είναι η πιθανότητα ο πράκτορας να επιλέξει τη δράση α ,$A_t = a$, δεδομένου ότι βρίσκεται στη κατάσταση $S_t = s$. Οι μέθοδοι ενισχυτικής μάθησης καθορίζουν με ποιο τρόπο οι πράκτορες αλλάζουν την πολιτική τους με βάση την εμπειρία που λαμβάνουν. Ο τελικός στόχος του πράκτορα είναι να μεγιστοποιήσει το συνολικό ποσό των ανταμοιβών που λαμβάνει μακροπρόθεσμα.[27]

Στόχοι και ανταμοιβές

Στην ενισχυτική μάθηση ο στόχος του πράκτορα τυποποιείται σε ένα ειδικό σήμα ενίσχυσης-ανταμοιβής, το οποίο στέλνεται από το περιβάλλον στον πράκτορα. Σε κάθε χρονικό βήμα, η ανταμοιβή είναι ένας απλός αριθμός $R_t \in \mathbf{R}$.

Ο στόχος του πράκτορα είναι να μεγιστοποιήσει το συνολικό ποσό των ανταμοιβών που δέχεται. Αυτό σημαίνει ότι ο στόχος είναι η μεγιστοποίηση της συνολικής ανταμοιβής και όχι της άμεσης ανταμοιβής. Μπορούμε να θεωρήσουμε τη συγκεκριμένη ιδέα ως την υπόθεση ανταμοιβής:

Μπορούμε να θεωρήσουμε το στόχο του πράκτορα ως τη μεγιστοποίηση της αναμενόμενης τιμής της συσσωρευμένου αθροίσματος των βαθμωτών σημάτων ενίσχυσης. [27]

Επιστροφές

Μέχρι αυτό το σημείο έχουμε αναφερθεί στο στόχο της μάθησης και στις ανταμοιβές μη υπολογιστικά. Σε αυτό το σημείο θα επιχειρήσουμε να ορίσουμε και υπολογιστικά την έννοια της ανταμοιβής. Αρχικά, θεωρούμε την ακολουθία ανταμοιβών μετά το χρονικό βήμα t και τη συμβολίζουμε ως $R_{t+1}, R_{t+2}, R_{t+3}, ...$ Γενικά, στην ενισχυτική μάθηση επιχειρούμε να μεγιστοποιήσουμε την αναμενόμενη επιστρφοφή, όπου τη συμβολίζουμε με G_t. Στην πιο απλή της μορφή, η επιστροφή G_t, μπορεί να γραφεί ως άθροισμα των όλων των ανταμοιβών που έχει λάβει ο πράκτορας:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + ... + R_T$$

όπου T είναι το τελευταίο χρονικό βήμα. Η συγκεκριμένη μοντελοποίηση της επιστροφής G_t έχει νόημα μόνο σε περιπτώσεις όπου το τελευταί χρονικό βήμα έχει φυσικό νόημα. Σε αυτήν την περίπτωση η αλληλεπίδραση ανθρώπου-περιβάλλοντος "σπάει" σε υπακολουθίες, οι οποίες ονομάζονται συνήθως ως επεισόδια (episodes). Τέτοιου είδους αλληλεπιδράσεις μπορούν να παρατηρηθούν στα παιχνίδια, όπου ένα επεισόδιο είναι η παρτίδα, αλλά και σε κάθε είδους επαναλαμβανόμενες διαδιακασίες σχετικά μικρού μήκους. Κάθε επεισόδιο τερματίζει σε μία κατάσταση τερματισμού. Μετά από τον τερματισμό πραγματοποιείται επανεκκίνηση (reset) στη σχέση πράκτορα-περιβάλλοντος. Ο πράκτορα συνήθως εκκινεί είτε από μία προκαθορισμένη κατάσταση επανεκκίνησης είτε από μία ή περισσότερες καταστάσεις επανεκκίνησης, όπου η επιλογή τους εμφανίζει κάποια προκαθορισμένη τυχαιότητα.

Από την άλλη πλευρά, σε πολλές περιπτώσεις η διαδικασία αλληλεπίδρασης πράκτορα-περιβάλλοντος δεν μπορεί να χωριστεί με φυσικό τρόπο σε επεισόδια. Τέτοιου είδους διαδιακασίες ονομάζονται συνεχείς διαδικασίες. Συνήθως αυτές οι διαδικασίες θεωρούνται προβληματικές καθώς το τελευταίο χρονικό βήμα είναι για $T = \infty$, με αποτέλεσμα πολύ εύκολα να απειρίζεται η τιμή της επιστροφής T. Γι' αυτόν το λόγο στη συνέχεια θα χρησιμοποιήσουμε μία διαφορετική μοντελοποίηση της τιμής της επιστροφής G_t.

Στο νέο υπολογιστικό ορισμό της επιστροφή ενσωματώνουμε την έννοια της έκπτωσης. Σύμφωνα και με αυτήν την προσέγγιση ο πράκτορας επιχειρεί να

επιλέξει σε κάθε βήμα τη δράση που μεγιστοποιεί μακροπρόθεσμα την τιμή της επιστροφής με έκπτωση. Πιο ειδικά, επιλέγει σε κάθε χρονικό βήμα μία δράση A_t, η οποία επιχειρεί να μεγιστοποισει την τιμή της ανανομενης επιστροφής με έκπτωση G_t:

$$G_t = R_{t+1} + \gamma * R_{t+2} + \gamma^2 * R_{t+3} + ... = \sum_{t=0}^{\infty} \gamma^t * R_t$$

όπου γ είναι ο εκπτωτικός όρος με τιμές στο στο δίαστημα $0 \leq \gamma \leq 1$

Η συγκεκριμένη παράμετρος γ καθορίζει το πόσο εξαρτάται η τιμή της επιστροφής από μελλοντικές ανταμοιβές. Για την ακρίβεια, μία ανταμοιβή που λαμβάνεται μελλοντικά μετά από κ χρονικά βήματα, πολλαπλασιάζεται με έναν εκπτωτικό παράγοντα $\gamma^{\kappa-1}$. Από τη στιγμή, μάλιστα, που ο παράγοντας γ είναι μικρότερος της μονάδας, $\gamma < 1$, τότε ισχύει ότι κάθε όρος $R_{t+k} * \gamma^k - 1$ είναι φραγμένος. Άρα και η τιμής της επιστροφής είναι φραγμένη σε αυτήν την περίπτωση. Τέλος, στην περίπτωση που γ=0, τότε ο πράκτορας ενδιαφέρεται μόνο για την άμεση ανταμοιβή, καθώς όλες οι υπόλοιπες ανταμοιβές του αθροίσματος των ανταμοιβών μηδενίζονται, όπως φαίνεται εύκολα από τη σχέση $G_t = R_{t+1} + \gamma * R_{t+2} + \gamma^2 * R_{t+3} + ... = R_{t+1}$. [27]

3.3.3 Μαρκοβιανές Διαδικασίες αποφάσεων

Ένα βασικό ζητήμα στην ενισχυτική μάθηση είναι με ποιό τρόπο προτυποποιούμε μαθηματικά το περιβάλλον κίνησης του πράκτορα. Σε αυτό το σημείο έρχεται να δώσει λύση η ιδέα της Μαρκοβιανής αλυσίδας αποφάσεων (Markov Decision Process - MDP). Πριν, όμως, αναφερθούμε στη Μαρκοβιανή αλυσίδα αποφάσεων θα πρέπει να δώσουμε κάποιους επεξηγηματικούς ορισμούς.

Μαρκοβιανή Ιδιότητα: Το μέλλον δεν εξαρτάται από το παρελθόν δοθέντος του παρόντος.

Μαθηματικοποιημένα η συγκεκριμένη δήλωση σημαίνει:

$$P[s_{t+1}|s_t] = P[s_{t+1}|s_1, s_2, ..., s_t]$$

όπου το σύμβολο s_{t+1} συμβολίζει την επόμενη κατάσταση του πράκτορα και το σύμβολο s_t συμβολίζει την τρέχουσα κατάσταση του πράκτορα. Δηλαδή, σύμφωνα με την Μαρκοβιανή ιδιότητα η επόμενη κατάσταση του πράκτορα εξαρτάται αποκλειστικά από την τρέχουσα κατάστασή του και όχι από τις παρελθοντικές καταστάσεις.

Πίνακας πιθανοτήτων μετάβασης (State Transition Probability Matrix) Για κάθε μετάβαση από την κατάσταση s στην s', όπου s,s' δύο οποιεσδήποτε τυχαίες καταστάσεις λαμβάνουμε:

$$P_{s,s'} = P[s_{t+1} = s'|s_t = s]$$

και ονομάζουμε τη συγκεκριμένη ποσότητα ως πιθανότητα μετάβασης. Με αυτόν τον τρόπο προκύπτει κάθε στοχείο του πίνακα πιθανοτήτων μετάβασης. P=

$$\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}$$

όπου κάθε στοιχείο του πίνακα προκύπτει με βάση τη δεσμευμένη πιθανότητα μετάβασης.[12]

Πλέον, είμαστε σε θέση να όρίσουμε την έννοια της Μαρκοβιανής αλυσίδας αποφάσεων. Σύμφωνα με τη θεωρία πιθανοτήτων, μία στοχαστική διαδικασία X_n λέγεται μαρκοβιανή αλυσίδα αποφάσεων αν για κάθε $n \in \mathbf{N}$, η δεσμευμένη κατανομή της X_{n+1} δοθέντων των $X_0, .., X_n$, ταυτίζεται με τη δεσμευμένη κατανομή της X_{n+1} με μόνη δοθείσα τη X_n. Με άλλα λόγια μία μαρκοβιανή αλυσίδα αποφάσεων αποτελεί ένα στοχαστικό μοντέλο το οποίο περιγράφει μια αλληλουχία από δυνατά γεγονότα, στα οποία η πιθανότητα εμφάνισης του καθενός, εξαρτάται μόνο από την κατάσταση που πραγματοποιήθηκε στο προηγούμενο γεγονός. Στον τομέα της ενισχυτικής μάθησης θεωρούμε την μαρκοβιανή αλυσίδα αποφάσεων ως μία αλυσίδα αποφάσεων χωρίς μνήμη. Παραθέτουμε παρακάτω τον ορισμό:

Ορισμός

Μαρκοβιανή αλυσίδα αποφάσεων

Μια Μαρκοβιανή Διαδικασία (Αλυσίδα) ορίζεται ως ένα σύνολο $< S, P >$

1.Το S είναι ένα σύνολο πεπερασμένων καταστάσεων.

2.Το P είναι ένας πίνακας πιθανοτήτων μετάβασης, με στοιχεία που δίνονται από τη σχέση
$P_{s,s'} = P[s_{t+1} = s' | s_t = s]$
όπου όπου s,s' δύο οποιεσδήποτε τυχαίες καταστάσεις του S.

Αν προσθέσουμε την έννοια της ανταμοιβής μετά από την μετακίνηση του πράκτορα από μία κατάσταση στην επόμενη τότε δημιουργούμε μία Μαρκοβιανή διαδικασία ανταμοιβής. Παρουσιάζουμε παρακάτω:

Ορισμός

Μαρκοβιανή διαδικασία αποφάσεων

Μια Μαρκοβιανή Διαδικασία Αποφάσεων ορίζεται ως ένα σύνολο $< S, P, R, \gamma >$

1.Το S είναι ένα σύνολο πεπερασμένων καταστάσεων.

2.Το P είναι ένας πίνακας πιθανοτήτων μετάβασης, με στοιχεία που δίνονται από τη σχέση
$P_{s,s'} = P[s_{t+1} = s' | s_t = s]$
όπου όπου s,s' δύο οποιεσδήποτε τυχαίες καταστάσεις του S.

3.Ο όρος R αποτελεί μια συνάρτηση ανταμοιβής,$R_{s,a} = E[R_{t+1} | S_t = s, A_t = a]$

4.Ο όρος γ αποτελεί έναν παράγοντα έκπτωσης, όπου $\gamma \in [0, 1]$

Μία διαδικασία ενισχυτικής μάθησης, η οποία ικανοποιεί την Μαρκοβιανή ιδιότητα ονομάζεται Μαρκοβιανή διαδικασία αποφάσεων, MDP. Αν το σύνολο καταστάσεων του περιβάλλοντος και το σύνολο των δυνατών δράσεων του πράκτορα είναι πεπερασμένο τότε η διαδικασία ονομάζεται πεπερασμένη Μαρκοβιανή διαδικασία αποφάσεων finite MDP. Η έννοια της πεπερασμένης, Μαρκοβιανής διαδικασίας αποφάσεων είναι βασική στη θεωρία της ενισχυτικής μάθησης. Αποτελεί βασικό στοιχείο θεωρία για να κατανοήσει κάποιος το 90% της σύγχρονης ενισχυτικής μάθησης. Μία συγκεκριμένη πεπερασμένη Μαρκοβιανή διαδικασία αποφάσεων ορίζεται από το σύνολο καταστάσεων του περιβάλλοντος, από το σύνολο δράσεων του περιβάλλοντος και από τις δυναμικές ενός βήματος του περιβάλλοντος. Με τον όρο δυναμικές ενός βήματος του περιβάλλοντος εννοούμε τον πίνακα πιθανοτήτων μετάβασης που ορίσαμε, αλλά και ττις ανταμοιβές μετάβασης.

Παρακάτω παραθέτουμε τον ορισμό της Μαρκοβιανής διαδικασία αποφάσεων MDP:

Ορισμός

Μαρκοβιανή διαδικασία αποφάσεων

Μια Μαρκοβιανή Διαδικασία Αποφάσεων ορίζεται ως ένα σύνολο $< S, A, P, R, \gamma >$
1.Το S είναι ένα σύνολο πεπερασμένων καταστάσεων.

2.Το σύνολο A αποτελεί ένα σύνολο από δυνατές δράσεις που μπορεί να χρησιμοποιήσει ο πράκτορας.

3.Το P είναι ένας πίνακας πιθανοτήτων μετάβασης, με στοιχεία που δίνονται από τη σχέση
$P_{s,s'} = P[s_{t+1} = s'|s_t = s]$
όπου όπου s,s' δύο οποιεσδήποτε τυχαίες καταστάσεις του S.

4.Ο όρος R αποτελεί μια συνάρτηση ανταμοιβής,$R_{s,a} = E[R_{t+1}|S_t = s, A_t = a]$

5.Ο όρος γ αποτελεί έναν παράγοντα έκπτωσης, όπου $\gamma \in [0, 1]$

[27],[37]

3.3.4 Δυναμικός Προγραμματισμός

Στην ενισχυτική μάθηση με τον όρο δυναμικός προγραμματισμός αναφερόμαστε σε ένα σύνολο από αλγορίθμους, οι οποίοι μπορούν να χρησιμοποιηθούν για τον υπολογισμό των βέλτιστων πολιτικών δοθέντος ενός τέλειου περιβάλλοντος με τη μορφή MDP - Markov Decision Process(Μαρκοβιανή Διαδικασία αποφάσεων). Γιατί, όμως, δε χρησιμοποιούνται εκτεταμένα στην πράξη οι αλγόριθμοι δυναμικού προγραμματισμού; Κλασσικοί αλγόριθμοι δυναμικού προγραμματισμού δε χρησιμοποιούνται στην πράξη γιατί:

1. Δεν υπάρχει τελειο μοντέλο που να περιγράφει το περιβάλλον.
2. Έχουν μεγάλο υπολογιστικό κόστος.

Όμως, από θεωρητική άποψη οι συγκεκριμένοι αλγόριθμοι εμφανίζουν μεγάλη χρησιμότητα καθώς παρέχουν το απαραίτητο υπόβαθρο για την κατανόηση της πλειονότητας των πρακτικά εφαρμόσιμων μεθόδων ενισχυτικής μάθησης. Η κύρια ιδέα του δυναμικού προγραμματισμού θεωρείται η χρήση των συναρτήσεων τιμών για την οργάνωση και δόμηση του χώρου των πολιτικών, με σκοπό την αποδοτική αναζήτηση των βέλτιστων εξ αυτών. Στο δυναμικό προγραμματισμό συνήθως υποθέτουμε ότι υπάρχει ένα τέλειο Μαρκοβιανό μοντέλο αποφάσεων που περιγράφει το περιβάλλον κίνησης του πράκτορα. Θεωρούμε πάντα ότι τα σύνολα καταστάσεων, δράσεων και ανταμοιβών, S, A(s), R αντίστοιχα είναι πεπερασμένα. Επίσης, θεωρούμε ότι η δυναμική τους δίνεται από ένα σύνολο πιθανοτήτων $p(s', r|s, a)$ για όλα τα $s, s' \in S$, $a \in A$ και $r \in \mathbf{R}$. Οι αλγόριθμοι δυναμικού προγραμματισμού μπορούν να εφαρμοστούν σε εφαρμογές

με συνεχές πεδίο καταστάσεων και δράσεων, ποσοτικοποιώντας, κανοντάς τα πεπερασμένα, τα πεδία των καταστάσεων και των δράσεων. Αξίζει να σημειώσουμε ότι σε τέτοιου είδους περιπτώσεις λαμβάνουμε, όμως, προσεγγιστικές λύσεις. [27]

Οι βασικοί όροι στο δυναμικό προγραμματισμό είναι η αξιολόγηση πολιτικής (policy evaluation) και η βελτίωση πολιτικής (policy improvement).

Αξιολόγηση πολιτικής (policy evaluation)

Αρχικά εξετάζουμε με ποιόν τρόπο μπορούμε να υπολογίσουμε τη συνάρτηση αξίας κατάστασης υ_π για μια αυθαίρετη πολιτική π. Η συγκεκριμένη στρατηγική, στη βιβλιογραφία, ονομάζεται αξιολόγηση πολιτικής (policy evaluation). Αλλιώς ονομάζεται και ως πρόβλημα πρόβλεψης. Γνωρίζουμε ότι για κάθε $s \in S$:

$$u_{\pi(s)} = E[R_{t+1} + \gamma * R_{t+2} + \gamma^2 * R_{t+3} + ...|S_t = s] =$$

$$= E[R_{t+1} + \gamma * u_\pi(S_{t+1})|S_t = s] =$$
$$= \sum_{\pi(a|s)} \sum_{(s',r)} p(s', r|s, a) * [r + \gamma * V(s')]$$

όπου $\pi(\alpha|s)$ είναι η πιθανότητα ο πράκτορας να λάβει τη δράση α ακολουθώντας τη πολιτική π. Επίσης οι αναμενόμενες τιμές Ε έχουν δείκτη π για να δειχθεί ότι κινούνται σύμφωνα με την πολιτική π. Η ύπαρξη και η μοναδικότητα της λύσης u_π εξασφαλίζεται όταν είτε η σταθερα γ είναι μικρότερη της μονάδας, $\gamma < 1$, είτε όταν ο τερματισμός της διαδρομής του πράκτορα εξασφαλίζεται από όλες τις καταστάσεις, $s \in S$, στις οποίες μπορεί να βρεθεί ο πράκτορας. Θεωρούμε μία ακολουθία από προσεγγιστικές συναρτήσεις τιμών $u_0, u_1, u_2, ...$, καθεμία αντιστοιχίζεται από το S^+ στο R. Η αρχική προσέγγιση, u_0, επιλέγεται αυθαίρετα, με εξαίρεση την τελική κατάσταση όπου θέτουμε πάντα την τιμή 0. Για κάθε διαδοχική προσέγγιση επιλέγεται η εξίσωση Bellman:

$$u_{\kappa+1}(s) = E[R_{t+1} + \gamma * u_\kappa(S_{t+1})|S_t = s] =$$

$$= \sum_{\pi(\alpha|s)} \sum_{(s',r)} *p(s', r|s, a) * [r + \gamma * V(s')]$$

Η σχέση ισχύει για όλα τα $s \in S$. Επιπλέον, γνωρίζουμε ότι στη γενική περίπτωση η ακολουθία συναρτήσεων u_k συγκλίνει στη u_π, όταν κ $\to$ inf, κάτω από τις ίδιες υποθέσεις που εξασφαλίζουν την ύπαρξη της u_π. Ο αλγόριθμος, τον οποίο μόλις περιγράψαμε, ονομάζεται αλγόριθμος αξιολόγησης

πολιτικής policy evaluation. Παραθέτουμε παρακάτω σε μορφή ψευδοκώδικα τη λειτουργία του αλγορίθμου:

Ψευδοκώδικας αξιολόγησης πολιτικής

Input: π the policy to be evaluated
repeat
    Δ=0
    For each $s \in S$ u=V(s)
    $V(s) = \sum_{\pi(a|S)} \sum_{(s',r)} p(s', r|s, a) * [r + \gamma * V(s')]$
    $\Delta = max(\Delta, |u - V(s)|)$
until($\Delta \leq \theta$) όπου ϑ πολύ μικρή σταθερά

[27]

Για να παραχθεί κάθε διαδοχική εκτίμηση καταστάσεων u_{k+1} από τη u_κ, ο αλγόριθμος αξιολόγησης πολιτικής εφαρμόζει την ίδια λειτουργία σε κάθε κατάσταση $s \in S$. Ειδικότερα, αντικαθιστά τις παλιές τιμές των καταστάσεων s, με τις νέες (επανάληψης κ+1), οι οποίες έχουν προκύψει από τις παλαιές τιμές (επανάληψης κ) των διαδοχικών καταστάσεων της s. Με τον όρο διαδοχικές καταστάσεις εννοούμε τις γειτονικές καταστάσεις της κατάστασης s, one step transition, κάτω από την πολιτική που ακολουθείται.

Βελτίωση πολιτικής (policy improvment)

Υπολογίζουμε, κυρίως, τη συνάρτηση τιμής για μία πολιτική, ώστε μέσω της υπολογισμένης συνάρτησης τιμής να μπορέσουμε να υπολογίσουμε μία νέα καλύτερη πολιτική. Θεωρούμε ότι έχουμε επιλέξει τη συνάρτηση τιμών value function για κάποια αυθαίρετη πολιτική π. Για κάθε κατάσταση s θα θέλουμε να γνωρίζουμε αν πρέπει να αλλάξει η πολιτική και όταν ο πράκτορας βρίσκεται στην κατάσταση s να επιλέγει ντετερμινιστικά μία δράση $a^1 \neq a$. Προκύπτει με αυτόν τρόπο ένα βασικό ζήτημα. Σε ποιά περίπτωση πρέπει να επιλέξουμε μία νέα πολιτική π' και σε ποιά περίπτωση πρέπει να συνεχίσουμε να ακολουθούμε την υπάρχουσα π. Ένας τρόπος για να επιλύσουμε το ζήτημα είναι να επιλέξουμε τη δράση a^1 αντί της a όταν βρεθούμε στην κατάσταση s και στη συνέχεια να ακολουθήσουμε πάλι την κατάσταση s'. Παραθέτουμε παρακάτω:

$$q_\pi(s, a^1) = E_\pi[R_{t+1} + \gamma * \upsilon_\pi(S_{t+1})|S_t = s, A_t = a^1] =$$

$$= \sum_{\alpha^1, s} p(s', r|s, a^1) * [r + \gamma * \upsilon_\pi(S_{t+1})]$$

Μας ενδιαφέρει να μάθουμε αν η ποσότητα $q_\pi(s, a^1)$ είναι αυστηρά μεγαλύτερη από την $\upsilon_\pi(s)$. Σε περίπτωση που η ποσότητα $q_\pi(s, a^1)$ είναι αυστηρά μεγαλύτερη, αυτό σημαίνει ότι όταν ο πράκτορας βρεθεί στην κατάσταση s θα πρέπει να επιλέξει τη δράση α^1. Εύκολα κάποιος παρατηρητής θα ανέμενε κάθε φορά που ο πράκτορας βρεθεί στην κατάσταση s να πρέπει να επιλέξει τη δράση α^1. Μα αυτή η στρατηγική αποτελεί καινούρια πολιτική. Το γεγονός ότι ισχύει η προγηγούμενη παρατήρηση είναι αποτέλεσμα ενός γενικότερου αποτελέσματος, το οποίο ονομάζεται θεώρημα βελτίωσης πολιτικής (Policy Improvment theorem). Θεωρούμε οποιοδήποτε ζευγάρι ντετερμινιστικών πολιτικών π και π', για τις οποίες ισχύει για κάθε $s \in S$:

$$q_\pi(s, a) \geq \upsilon_\pi(s)$$

Αυτό σημαίνει ότι η πολιτική π' πρέπει να είναι ίση ή καλύτερη από την π. Σε αυτήν την περίπτωση πρέπει να ισχύει:

$$\upsilon_{\pi'}(s) \geq \upsilon_\pi(s)$$

για κάθε $s \in S$

Η ιδέα πίσω από την απόδειξη του θεωρήματος βελτίωσης πολιτικής (Policy Improvment theorem) είναι αρκετά απλή, αλλά δεν είναι ο σκοπός μας να πραγματοποιήσουμε τη συγκεκριμένη ανάλυση. Ο σκοπός μας στην παρούσα διπλωματική είναι να εισαγάγουμε τον αναγνώστη στις έννοιες της βαθιάς ενισχυτικής μάθησης. Γι' αυτόν το λογο δε θα προχωρήσουμε σε περαιτέρω ανάλυση και επεξήγηση της απόδειξης του θεωρήματος βελτίωσης πολιτικής.

Ως αυτό το σημείο δείξαμε με ποιό τρόπο, δοθείσης μίας πολίτικής π και μίας συνάρτησης τιμών, μπορούμε να εκτιμήσουμε τι βελτιώσεις θα επιφέρει, μία αλλαγή στην επιλογή δράσης πάνω σε μία συγκεκριμένη κατάσταση s στην υπάρχουσα πολιτική π. Είναι φυσική επέκταση να σκεφτούμε ότι μπορούμε να αλλάξουμε τις επιλογές δράσεων σε κάθε κατάσταση $s \in S$ με τέτοιο τρόπο, ώστε σε κάθε κατάσταση $s \in S$ να επιλέγεται η δράση a που μεγιστοποιεί την ποσότητα $q_{\pi(s,a)}$. Με άλλα λόγια υπολογίζουμε τη νέα άπληστη,greedy, πολιτική π' ως εξής:

$$\pi'(s) = argmax_a q_\pi(s, a) = E_\pi[R_{t+1} + \gamma * \upsilon_\pi(S_{t+1})|S_t = s, A_t = a] =$$
$$= \sum_{(s,r)} p(s', r|s, a) * [r + \gamma * \upsilon_\pi(S_{t+1})]$$

Όπου με τον όρο argmax εννοούμε τη δράση a που μεγιστοποιεί την ποσότητα argmax. Η άπληστη (greedy) διαδικασία δημιουργίας νέας πολιτικής π', η οποία βελτιώνει την αρχική πολιτική π χρησιμοποιώντας τη συνάρτηση τιμών της πολιτική π, ονομάζεται πολιτική βελτίωσης Policy Improvment.[27]

Επαναλήψεις πολιτικών Policy Iteration

1. Initialization
V (s) $\in R$ and π(s) $\in$ A(s) arbitrarily for all $s \in S$

2. Policy Evaluation Input:
π the policy to be evaluated
repeat
    Δ=0
    For each $s \in S$
    u=V(s)
    $V(s) = \sum_{\pi(a|s)} * \sum_{(s',r)} p(s', r|s, a) * [r + \gamma * V(s')]$
    $\Delta = max(\Delta, |u - V(s)|)$
until($\Delta < \theta$) όπου ϑ πολύ μικρή σταθερά

3. Policy Improvment
policy-stable = true
For each $s \in S$:
    a = π(s)
    π(s) =$argmax_a \Sigma_{s,r} p(s', r|s, a) * [r + \gamma * \upsilon_\pi(s_{t+1})]$
    If $a \neq$ π(s), then policy-stable = false

[27] Υπάρχουν περαιτέρω βελτιστοποιήσεις του αλγορίθμου επανάληψη πολιτικής στις οποίες, όμως, δε θα επεκταθούμε στην παρούσα διπλωματική. Για περισσότερη εμβάθυνση στο ζήτημα μπορεί ο αναγνώστης να να αναζητήσει πληροφορίες στη σχετική βιβλιογραφία. Κατά την προσωπική άποψη του συγγραφέα πολύ καλά περιγράφονται τα συγκεκριμένα ζητήματα στο βιβλίο [27], το οποίο πραγματεύεται τις βασικές αλλά και πιο προχωρημένες έννοιες ενισχυτικής μάθησης.

3.3.5 Μέθοδοι Monte Carlo

Στην υποενότητα του δυναμικού προγραμματισμού ασχοληθήκαμε κυρίως με τη σχεδίαση αλγορίθμων, οι οποίοι βασίζονται στη γνώση του προβλήματος ή με άλλα λόγια στη Μαρκοβιανή Διαδικασία Αποφάσεων. Στις επόμενες υποενότητες του παρόντος κεφαλαίου θα ασχοληθούμε με την αξιολόγηση πολιτικών για άγνωστα μοντέλα. Οι μέθοδοι Monte Carlo (MC) αποτελούν ένα υποσύνολο υπολογιστικών αλγορίθμων που χρησιμοποιούν τη διαδικασία επαναλαμ-

βανόμενης τυχαίας δειγματοληψίας για να πραγματοποιήσουν αριθμητικές εκτιμήσεις άγνωστων παραμέτρων. Επιτρέπουν τη μοντελοποίηση πολύπλοκων καταστάσεων όπου εμπλέκονται πολλές τυχαίες μεταβλητές και αξιολογείται η επίδραση του κινδύνου. Οι χρήσεις του MC είναι απίστευτα ευρείας εμβέλειας και έχουν οδηγήσει σε μια σειρά πρωτοποριακών ανακαλύψεων στους τομείς της φυσικής, της θεωρίας παιγνίων και της χρηματοδότησης.

Οι μέθοδοι MC αποτελούν μεθόδους, οι οποίες μπορούν να επιλύσουν το πρόβλημα της ενισχυτικής μάθησης βασιζόμενες στο μέσο όρο των δειγματοληπτικών επιστροφών. Για να εξασφαλίσουμε ότι οι επιστροφές θα είναι καλά ορισμένες, στη συγκεκριμένη εργασία ορίζουμε τις μεθόδους MC μόνο για διεργασίες, tasks, που αποτελούνται από επεισόδια. Με αυτό το σκεπτικό θεωρούμε ότι η εμπειρία τις διεργασίας χωρίζεται σε επεισόδια και ότι κάθε επεισόδιο θα τερματίσει, ανεξαρτήτως των καταστάσεων που μπορεί να βρεθεί ο πράκτορας. Μόνο μετά την ολοκλήρωση ενός επεισοδίου πραγματοποιείται υπολογισμός των τιμών που μας ενδιαφέρουν. Επίσης, αυστηρά μετά την ολοκλήρωση ενός επεισοδίου μπορεί να πραγματοποιηθεί αλλαγή στην πολιτική του πράκτορα. Ουσιαστικά οι μέθοδοι MC βασίζονται στη λογική της ενημέρωσης των τιμών-στοιχείων ανά επεισόδιο episode by episode mode. Δε θεωρούνται online αλγόριθμοι, καθώς δεν εμφανίζουν step by step ενημερώσεις. Όρος Monte Carlo (MC) συνήθως αναφέρεται σε οποιαδήποτε μέθοδο εκτίμησης, της οποίας η εκτέλεσή της βασίζεται στην τυχαιότητα. [27] Σε αυτήν τη διπλωματική αναφερόμαστε, όπως και στη βιβλιογραφία [27] με τον όρο μέθοδοι MC αναφερόμαστε στις μεθόδους που λαμβάνουν τον μέσο όρο των επιστροφών από κάθε επεισόδιο.

Θα ξεκινήσουμε την ανάλυση των μεθόδων MC, με έναν από τους πιο στοιχειώδεις αλγορίθμους MC, ώστε να μπορέσει να κατανοήσει καλύτερα και πιο ομαλά ο αναγνώστης τη λειτουργία των MC αλγορίθμων. Αυτός ο αλγόριθμος είναι η μέθδος MC πρώτης επίσκεψης first visit. Με τη χρήση του συγκεκριμένου αλγορίθμου επιχειρούμε να προσεγγίσουμε τις τιμές της συνάρτησης τιμών value function. Πιο συγκεκριμένα, επιχειρούμε να εκτιμήσουμε την τιμή της $\upsilon_\pi(s)$, δηλαδή την τιμή της συνάρτηση τιμών για την κατάσταση s όταν ακολουθείται η πολιτική π. Όποτε εμφανίζεται η κατάσταση s θεωρούμε ότι πραγματοποιείται επίσκεψη στην κατάσταση s. Προφανώς, κατά τη διάρκεια ενός επεισοδίου ένας πράκτορα μπορεί να επισκεφτεί πολλές φορές την κατάσταση s. Εμείς, όμως, στον αλγόριθμο MC πρώτης επίσκεψης ενδιαφερόμαστε για την πρώτη εμφάνιση της κατάστασης s κατά τη διάρκεια ενός επεισοδίου. Σημειώνουμε ότι υπάρχει και η μέθοδος every visit MC, όπου σε αυτήν την περίπτωση ενδιαφερόμαστε για κάθε εμφάνιση της κατάστασης s κατά τη διάρκεια ενός επεισοδίου. Οι μέθοδοι every visit MC και first visit MC εμφανίζουν πολλά οφθαλμοφανή κοινά μεταξύ τους, αλλά εμφανίζουν διαφορετικές θεωρητικές ιδιότητες.[27]

Παρουσιάζουμε παρακάτω με μορφή πίνακα τον αλγόριθμο first visit MC:

first visit Monte Carlo method
Initialize: π ← policy to be evaluated
V an arbitrary state-value function
Returns(s) ← an empty list, for all $s \in S$

Repeat forever:
    Generate an episode using π
    For each state s appearing in the episode:
    G ← return following the first occurrence of s
    Append G to Returns(s)
    V(s) ← average(Returns(s))

[27]

Στον παραπάνω πίνακα απεικονίζεται η first visit MC μέθοδος. Παρατηρούμε ότι η χρησιμοποιείται το κεφαλαίο γράμμα V για την απεικόνιση της συνάρτησης τιμών, γιατί μετά την τυχαία αρχικοποίησή της θεωρείται μία τυχαία μεταβλητή. Κατά τη διάρκεια κάθε επεισοδίου υπολογίζεται η τιμή της συνάρτησης τιμών για κάθε κατάσταση s, ως ο μέσος όρος των επιστροφών Returns(s) που λαμβάνουμε από την πρώτη εμφάνιση της κατάστασης s σε όλα τα επεισόδια που έχουν διαδρματιστεί μέχρι εκείνη την ολοκλήρωση εκείνου του επεισοδίου. Αξίζει να σημειώσουμε ότι τόσο η μέθοδος μέθοδοι first visit MC όσο και η every visit MC συγκλίνουν στην τιμή της $\upsilon_\pi(s)$ για κάθε κατάσταση s, καθώς ο αριθμός των ολοκληρωμένων επεισοδίων τείνει στο άπειρο.

Monte Carlo εκτίμηση τιμών κατά την επιλογή ζεύγους κατατάστασης-δράσης

Αν δεν έχουμε διαθέσιμο ένα μοντέλο πλοήγησης στο περιβάλλον τότε είναι χρήσιμο να γνωρίζουμε τις τιμές της συνάρτησης κατατάστασης-δράσης. Δηλαδή, αν επιλέξουμε ένα ζευγάρι κατατάστασης-δράσης να μπορούμε να προβλέψουμε την τιμή που θα μας αποδώσει. Όταν γνωρίζουμε μόνο τις τιμές της συνάρτησης τιμών μας είναι αδύνατο να προβλέψουμε τις βέλτιστες δράσεις, με αποτελέσμα να μην μπορούμε να επιλέξουμε πολιτική. Κάποιος πρέπει να εκτιμήσει τι τιμή θα μας αποδώσει κάθε ζευγάρι κατατάστασης-δράσης αν επιλεγεί. Γι' αυτόν το λόγο ένα από τα κύρια ζητήματα των μεθόδων MC είναι να μπορέσουν να προβλέψουν τις τιμές της συνάρτησης κατατάστασης-δράσης, q_*. Για να επιλύσουμε το ζήτημα αντιμετωπίζουμε το πρόβλημα της εκτίμησης των τιμών της συνάρτησης κατατάστασης-δράσης.

Το πρόβλημα της εκτίμησης των τιμών της συνάρτησης κατατάστασης-δράσης είναι να εκτιμήσουμε τις τιμές της $q_\pi(s, a)$, δηλαδή την επιστροφή που

αναμένουμε να λάβουμε όταν βρεθούμε στην τυχαία κατάσταση s και πραγματοποιήσουμε τη δράση a και στη συνέχεια ακολουθήσουμε την πολιτική π. Οι μέθοδοι MC είναι βασικά οι ίδιες με αυτές που ασχολούνται με την εύρεση μόνο των συναρτήσεων τιμών, με τη μόνη διαφορά ότι ασχολούμαστε, πλέον, με τον προσδιορισμό της συνάρτησης ζεύγους τιμών και όχι με τον προσδιορισμό της συναρτησης τιμών. Θεωρούμε ότι ένα ζεύγος συνάρτησης τιμών, s, a, έχει επισκεφθεί κατά τη διάρκεια ενός επεισοδίου αν ο πράκτορας έχει επισκεφτεί την κατάσταση s και πραγματοποίησε, όταν βρέθηκε εκεί, τη δράση a. Η μέθοδος first visit MC λαμβάνει το μέσο όρο των επιστροφών που ακολουθούν από την πρώτη επίσκεψη του πράκτορα σε μία κάτασταση s όταν πραγματοποιεί τη δράση a. Με αυτόν τον τρόπο από μία σειρά επεισοδίων λαμβάνει το μέσο όρο των επιστροφών από το κάθε ζευγάρι s,a και υπολογίζει τη συνάρτηση $q_\pi(s, a)$. Αυτού του τύπου οι μέθοδοι συγκλίνουν με τετραγωνικό ρυθμό στις πραγματικές αναμενόμενες τιμές, καθώς ο αριθμός των επισκέψεων σε κάθε ζευγάρι s,a πρσεγγίζει το άπειρο.[27]

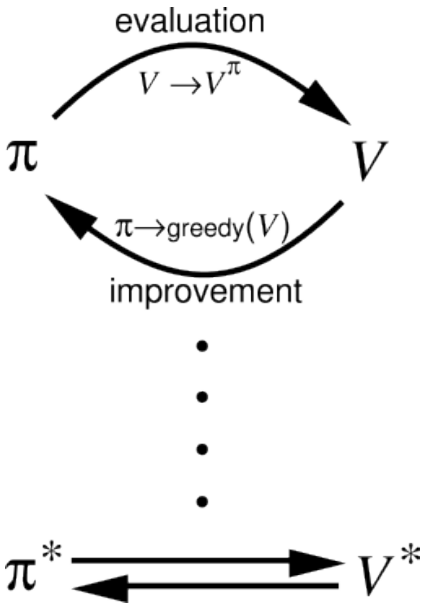
Έλεγχος Monte Carlo

Είμαστε τώρα έτοιμοι να εξετάσουμε πώς μπορεί να χρησιμοποιηθεί η εκτίμηση του Monte Carlo στον έλεγχο, δηλαδή για την προσέγγιση των βέλτιστων πολιτικών. Η γενική ιδέα είναι να χρησιμοποιήσουμε την ίδια λόγική με αυτή που χρησιμοποιούμε και στο δυναμικό προγραμματισμό για την εύρεση βέλτιστων πολιτικών. Αυτή η λογική ονομάζεται γενικευμένη επανάληψη πολιτικής, generalized policy iteration (GPI). Στη μέθοδο GPI διατηρούμε μία προσεγγιστική πολιτική και μία προσεγγιστική συνάρτηση τιμών. Η συνάρτηση τιμών συνεχώς διαφοροποιείται, ώστε να προσεγγίζει καλύτερη την πραγματική συνάρτηση τιμών. Η πολιτική βελτιώνεται με βάση την τρέχουσα συνάρτηση τιμών. Παραθέτουμε και σχηματικά το συγκεκριμένο αλγόριθμο GPI, βλέπε εικόνα 3.12.

Αυτά τα δύο είδη αλλαγών εναλλάσονται μεταξύ τους, καθώς το ένα δημιουργεί ένα κινούμενο στόχο στο άλλο. Από τη μία η ανανεωμένη συνάρτηση τιμών παρέχεται στον αλγόριθμο προσέγγισης πολιτικής για την εύρεση βελτιωμένης πολιτικής. Από την άλλη η βελτιωμένη πολιτική παρέχεται στον αλγόριθμο προσέγγισης συνάρτησης τιμών για την βελτίωση της συνάρτηση τιμών. Η διαρκής εναλλαγή μεταξύ των δύο προσεγγιστικών αλγορίθμων μας οδηγεί στο βέλτιστο αποτέλεσμα.

Για να ξεκινήσουμε την ανάλυσή μας, ας σκεφτούμε μία μέθοδο MC για την κλασική μέθοδο επανάληψης πολιτικής. Σε αυτού του είδους την μεθοδολογία, πραγματοποιούνται, εναλλακτικά, πλήρη βήματα των αλγορίθμων αξιολόγησης πολιτικής και βελτίωσης πολιτικής. Ξεκινάει η διαδικασία με μία αρχική πολιτική π_0 και ολοκληρώνεται με τη λήψη της βέλτιστης πολιτικής π_* και της βέλτιστης συνάρτησης κατάστασης-δράσης:

$$\pi_0 \xrightarrow{\text{E}} q_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} q_{\pi_1} \xrightarrow{\text{I}} \ldots \xrightarrow{\text{E}} \pi_n$$



Εικόνα 3.12: generalized policy iteration (GPI)

όπου το βέλος $\xrightarrow{\text{E}}$ συμβολίζει μία πλήρη εκτίμηση της συνάρτησης κατάστασης-δράσης, ενώ το βέλος $\xrightarrow{\text{I}}$ συμβολίζει μία βελτίωση πολιτικής. Η εκτίμηση της συνάρτησης τιμών πραγματοποιείται, όπως ακριβώς, έχουμε περιγράψει σε προηγούμενη παράγραφο για τις μεθόδους MC. Λαμβάνει ο πράκτορας εμπειρία από πολλά επεισόδια και μέσω της προσεγγιστικής συνάρτησης κατάστασης-δράσης προσεγγίζει την πραγματική συνάρτηση κατάστασης-δράσης. Για να εξασφαλίσουμε ότι οι μέθοδοι MC θα υπολογίσουν τη συνάρτηση q_{π_κ} για οποιαδήποτε πολιτική π_κ, πραγματοποιούμε δύο υποθέσεις. Πρώτον, θεωρούμε ότι για την υλοποίση της μεθόδου εκτίμηση της συνάρτησης κατάστασης-δράσης για μία δεδομένη πολιτική, θα έχουμε στη διάθεσή μας άπειρο αριθμό επεισοδίων. Δεύτερον, θεωρούμε κάθε επεισόδιο παράγεται με τη μέθοδο εξερεύνησης διαθέσιμων ζευγών καταστάσης-δράσης exploring starts. Με τον όρο exploring starts εννοούμε ότι κάθε ζευγάρι καταστάσης-δράσης εμφανίζει μη μηδενική πιθανότητα να επιλεγεί ως θέση εκκίνησης ενός επεισοδίου.

Η μέθοδος βελτίωσης πολιτικής υλοποιείται ακολουθώντας μία άπληστη πολιτική κινήσεων, η οποία βασίζεται στην τρέχουσα υπολογισμένη συνάρτηση τιμών. Σε αυτήν την περίπτωση, από τη στιγμή που διαθέτουμε μία τρέχουσα υπολογισμένη συνάρτηση κατάστασης-δράσης, κανένα μοντέλο δεν απαιτείται για τον υπολογισμό της μίας άπληστης πολιτικής. Για οποιοδήποτε ζεύγος κατάστασης-δράσης, a, s, θέσουμε στη συνάρτηση q(a,s), η αντίστοιχη άπληστη πολιτική είναι αυτή που για κάθε $s \in S$, ντετερμινιστικά επιλέγει τη δράση, a, που μεγιστοποιεί την ποσότητα q(a,s):

$$\pi(s) = argmax_a q(a, s)$$

Με αυτόν τον τρόπο η μέθοδος της βελτίωσης πολιτικής μπορεί να ολοκληρωθεί κατασκευάζοντας την πολιτική $\pi_{\kappa+1}$ σαν έναν άπληστο αλγόριθμο, με χρήση της συνάρτησης q_{π_κ}. Το θεώρημα βελτίωσης πολιτικής εφαρμόζεται στις πολιτικές π_κ και $\pi_{\kappa+1}$ για κάθε $s \in S$:

$$q_{\pi_\kappa}(s, \pi_{\kappa+1}(s)) = q_{\pi_\kappa}(s, argmax_a q_{\pi_\kappa}(s, a)) =$$

$$= \max_a q_{\pi_\kappa}(s, a) \geq q_{\pi_\kappa}(s, \pi_\kappa(s)) = u_{\pi_\kappa}(s)$$

Το θεώρημα της βελτίωσης πολιτικής μας εξασφαλίζει ότι κάθε πολιτική $\pi_{\kappa+1}$ είναι ομοιόμορφα καλύτερη από την πολιτική π_κ, ή στη χειρότερη περίπτωση είναι το ίδιο καλή με την πολιτική π_κ. Σε αυτήν την περίπτωση είναι και οι δύο πολιτικές βέλτιστες πολιτικές. Με αυτόν τον τρόπο εξασφαλίζεται ότι η συνολική διαδικασία συγκλίνει στη βέλτιστη πολιτική και στη βέλτιστη συνάρτηση τιμής. [27]

Μέθοδοι Monte Carlo χωρίς την υπόθεση exploring starts

Η διασφάλιση της υπόθεσης exploring starts στην πράξη δεν είναι πάντα εφικτή. Γι' αυτόν το λόγο, στις επόμενες παραγράφους του παρόντος υποκεφαλαίου θα αναφέρθούμε σε μεθόδους MC, οι οποίες δεν χρησιμοποιούν τη μη ρεαλιστική υπόθεση exploring starts. Για να επιτύχουμε το σκοπό μας χρησιμοποιούμε μία νέα γενική ιδέα. Προσπαθούμε να εξασφαλίσουμε ότι ο πράκτορας θα συνεχίσει να επιλέγει όλες τις δυνατές δράσεις επ' άπειρον. Δύο είναι οι μέθοδοι που μας εξασφαλίζουν το ζητούμενο η on policy μέθοδος και η off policy μέθοδος. Η on policy μέθοδος επιχειρεί να εκτιμήσει ή να βελτιώσει την πολιτική που χρησιμοποιεί ο αλγόριθμος για να λαμβάνει αποφάσεις. Αντίθετα, η off policy μέθοδος επιχειρεί να εκτιμήσει ή να βελτιώσει μία πολιτική διαφορετική από εκείνη που έχει "γεννήσει" τα δεδομένα.

On policy μέθοδος

Στις on policy μεθόδους η πολιτική που ακολουθείται είναι γενικά soft, αυτό σημαίνει ότι η πιθανότητα $\pi(a|s)$ είναι μεγαλύτερη του μηδενός, $\pi(a|s) > 0$, για κάθε για κάθε $s \in S$ και κάθε $a \in A$. Ταυτόχρονα η ίδια on policy μέθοδος συγκλινει στη βέλτιστη πολιτική. Η on policy μέθοδος που παρουσιάζουμε στην τρέχουσα παράγραφο χρησιμοποιεί ε-greedy πολιτικές. Αυτό σημαίνει ότι τις περισσότερες φορές ο αλγόριθμος επιλέγει τη δράση που μεγιστοποιεί τη συνάρτηση κατάστασης-δράσης. Όμως, με πιθανότητα ε ο πράκτορας μπορεί να επιλέξει μία τυχαία δράση, δηλαδή να επιλεγεί ένα τυχαίο ζευγάρι κατάστασης-δράσης. Με αυτόν τον τρόπο όλες οι μη βέλτστες δράσεις έχουν μία μικρή πι-

θανότητα να επιλεγούν ίση με, $\varepsilon/A(s)$. Το υπολοιπόμενο ποσοστό αντιστοιχεί στη δράση που μεγιστοποιεί την τιμή της συνάρτησης κατάστασης-δράσης και εμφανίζει πιθανότητα ίση με $1-\varepsilon+\varepsilon/A(s)$.

Για καλύτερη κατανόηση παραθέτουμε παρακάτω τον ψευδοκώδικα του αλγορίθμου first visit MC control algorithm for ε-soft policies:

Initialize for all $s \in S, a \in A(s)$:
Q(s,a)=arbitrary
Returns(s,a)=empty list
$\pi(a|s)$=an arbitrary ε-soft policy

Repeat forever:
a)Generate an episode using policy π
b) For each s,a appearing in the episode:
G=return following the first occurence of s,a
Append G to Returns(s,a)
Q(s,a)=average(Returns(s,a))
c) For each s in episode:
$a_* = argmax_a Q(s,a)$
For all a in A(s):
if a=a_**then* :
π(a—s)=1-ε+ε/A(s)
elseif a $\neq a_*$: $\pi(a|s) = \varepsilon/A(s)$

[27]

Off policy μέθοδος

Στην προηγούμενη παράγραφο δείξαμε με ποιό τρόπο οι on policy μέθοδοι λειτουργούν. Στις on policy μέθοδους οι συναρτήσεις τιμών εκτιμώνται, μέσα από έναν άπειρο αριθμό επεισοδίων τα οποία έχουν παραχθεί από την ίδια μέθοδο. Ας προχωρήσουμε σε αυτό το σημείο στις off policy μεθόδους. Υποθέτουμε ότι όλα τα επεισόδια έχουν δημιουργηθεί από διαφορετικές πολιτικές. Επιχειρούμε να εκτιμήσουμε τις συναρτήσεις υ_π ή q_π, αλλά στη διάθεσή μας έχουμε επεισόδια τα οποία ακολουθούν μία πολιτική μ, όπου $\mu \neq \pi$. Θεωρούμε την πολιτική π, ως την πολιτική στόχο, γιατί ο στόχος της διαδικασίας μάθησης είναι είναι η εκμάθηση της συνάρτησης τιμών κάτω από την πολιτική π. Καλούμε την πολιτική μ ως πολιτική συμπεριφοράς γιατί είναι η πολιτική που ελέγχει τον πράκτορα και "γεννά" τη συμπεριφορά του. Το συνολικό πρόβλημα ονομάζεται off policy μάθηση γιατί ο στόχος μας είναι η εκμάθηση μίας πολιτικής, ενώ έχουμε στη διάθεσή μας εμπειρία που δεν ακολουθεί τη συγκεκριμένη πολιτική,off policy παραχθέντα επεισόδια. Έχουμε στη διάθεσή μας επεισόδια που

ακολουθούν διαφορετική πολιτική. [27]

Για να εκτιμήσουμε τιμές πάνω στην πολιτική π, πρέπει να εξασφαλίσουμε ότι μέσω των επειδοσίων από την πολιτική μ μπορούν να παραχθούν οι εκτιμήσεις μας. Γι' αυτόν το λόγο ότι κάθε δράση που μπορεί να επιλεχθεί κάτω από την πολιτική π μπορεί να επιλεχθεί και κάτω από την πολιτική μ. Πρέπει δηλαδή να εξασφαλίσουμε ότι αν $\pi(a|s) > 0$, τότε και $\mu(a|s) > 0$. Η πολιτική μ πρέπει να είναι στοχαστική, ενώ η πολιτική π μπορεί να είναι είτε στοχαστική είτε ντετερμινιστική. Τυπικά η πολιτική στόχος π πρέπει να είναι ντετερμινιστική και να χρησιμοποιεί την τρέχουσα εκτίμηση της συνάρτησης κατάστασης-δράσης. Αντίθετα, η πολιτική μ είναι μία στοχαστική πολιτική, όπως η ε-greedy πολιτική.

3.3.6 Πολιτικές Χρονικών Διαφορών (Temporal-Difference (TD) Learning)

Οι μέθοδοι εκμάθησης Temporal-Difference (TD) αποτελούν ένα συνδυασμό των μεθόδων μάθησης Monte Carlo και Δυναμικού Προγραμματισμού. Όπως οι μέθοδοι MC, έτσι και οι μέθοδοι Temporal-Difference μπορούν να μάθουν κατευθείαν από την εμπειρία χωρίς να βασίζονται σε κάποιο μοντέλο περιβάλλοντος. Όπως, ακριβώς, και οι μέθοδοι Δυναμικού Προγραμματισμού, οι μέθοδοι εκμάθησης TD πραγματοποιούν εκτιμήσεις, οι οποίες στηρίζονται σε ήδη γνωστές εκτιμήσεις, χωρίς να αναμένουν τα τελικά αποτελέσματα. Αποτελούν σημαντικό κομμάτι του ερευνητικού πεδίου της ενισχυτικής μάθησης.

TD μέθοδοι προβλέψεων

Τόσο η μέθοδος TD, όσο και η μέθοδος Monte Carlo χρησιμοποιούν την εμπειρία για να επιλύσουν το πρόβλημα των εκτιμήσεων. Δοθείσης κάποιας εμπειρίας που ακολουθεί την πολιτική π, οι μέθοδοι TD και Monte Carlo ενημερώνουν τις εκτιμήσεις υ της v_π για τις μη τερματικές καταστάσεις που συναντά ο πράκτορας σε αυτήν την εμπειρία. Εν αντιθέσει, με τις μεθόδους Monte Carlo όπου ο αναμένουμε την ολοκλήρωση του εκάστοτε επεισοδίου για να αποφασίσουμε την τιμή της προσεγγιστικής συνάρτησης τιμών $V(s_t)$, όταν δηλαδή η τιμή της G_t μας είναι γνωστή, οι μέθοδοι TD αναμένουν μόνο μέχρι το επόμενο βήμα. Αναμένουν να πληροφορηθούν την τιμή της ανταμοιβής R_{t+1}. Στη χρονική στιγμή $t+1$ επειχειρούν να ανανεώσουν την τιμή της πρσεγγιστικής συνάρτησης $V(s_t)$. Αυτή η μέθοδος είναι η πιο απλή και ονομάζεται TD(0). Η βασική εξίσωση ενημέρωσής σε αυτόν τον αλγόριθμο είναι η εξής:

$$V(S_t) = V(S_t) + a * [R_{t+1} + \gamma * V(S_{t+1}) - V(S_t)]$$

Όπως παρατηρούμε οι μέθοδοι Monte Carlo ενημερώνουν τη συνάρτηση τιμών μέσω της επιστροφής G_t, ενώ οι μέθοδοι TD ενημερώνουν τη συνάρτηση τιμών μέσω της σχέσης $R_{t+1} + \gamma * V(S_{t+1}) - V(S_t)$ [27]

Ψευδικώδικας μεθόδου TD(0) για την εκτίμηση της συνάρτησης υ_π:

Input the policy π to be evaluated:
Initialize V(s) arbitrarily (e.g. V(s)=0 for every s in S_+)

Repeat for each episode:
    A=action given by π for S
    Initialize S
    Repeat (for each step of episode):
        A=action given by π for S
        Take action A; observe reward, R and next state, S'
        $V(S) = V(S') + a * [R + \gamma * V(S') - V(S)]$
        S=S';
    until S is terminal

[27]

Sarsa: On-policy TD Control

Σε αυτό το σημείο παρουσιάζουμε μία on-policy TD Control μέθοδο. Στη μέθοδο Sarsa επιχειρούμε να μάθουμε τη συνάρτηση κατάστασης-δράσης και όχι την προσεγγιστική συνάρτηση τιμών, όπως συμβαίνει στη μέθοδο TD(0). Πιο συγκεκριμένα, στη μέθοδο Sarsa πρέπει να εκτιμήσουμε τη συνάρτηση $q_\pi(s, a)$ για την τρέχουσα πολιτική συμπεριφοράς π και για όλες τις καταστάσεις s και για όλες τις δράσεις a. Ο στόχος αυτός μπορεί να επιτευχθεί χρησιμοποιώντας βασικά την μέθοδο TD που περιγράψαμε στη θεωρία της μεθόδου TD(0). Στη μέθοδο TD(0) ασχοληθήκαμε με μεταβάσεις από κατάσταση σε κατάσταση. Στον αλγόριθμο Sarsa ασχολούμαστε με μεταβάσεις ζευγαριών κατάστασης δράσης και επιζητείται η μάθηση των συναρτήσεων τιμών κατάστασης δράσης. Η κύρια σχέση ενημέρωσης του αλγορίθμου που ακολουθείται σε κάθε βήμα κάθε επεισοδίου είναι η εξής:

$$Q(S, A) = Q(S, A) + a * [R + \gamma * Q(S', A') - Q(S, A)]$$

Όπως παρατηρούμε στη παραπάνω σχέση, η ενημέρωση των τιμών συνάρτησης $Q(S, A)$ πραγματοποιείται μετά από κάθε μετάβαση του πράκτορα σε μία μη τερματική κατάσταση S_t. Αν η κατάσταση S_t είναι τερματική τότε $Q(S_{t+1}, A_{t+1}) = 0$. Η παραπάνω σχέση ισχύει για κάθε δυνατή μετάβαση $(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1})$.

Παρακάτω παραθέτουμε αναλυτικά και τον ψευδοκώδικα της on-policy μεθόδου Sarsa:

Ψευδικώδικας μεθόδου Sarsa

Initialize Q(s,a) for every s in S and a in A(s), arbitrarily, and Q(terminal-state,.)=0

Repeat for each episode:
    Initialize S
    Choose A from S using policy derived from Q (e.g.,ε-greedy)
    Repeat (for each step of episode):
        Take acition A, observe R,S'
        Choose A' from S' using policy derived from Q(e.g.,ε-greedy)
        $Q(S,A) = Q(S,A) + a * [R + \gamma * Q(S',A') - Q(S,A)]$
        S=S'; A=A';
    until S is terminal

Μάθηση-Q: Off-policy TD Control

Μια από τις σημαντικότερες ανακαλύψεις στην ενισχυτική μάθηση, αφορά έναν off - policy αλγόριθμο ελέγχου. Ο αλγόριθμος αυτός ονομάζεται μάθηση-Q (Q-learning). Η πιο απλή του μορφή ονομάζεται μάθηση Q ενός βήματος και ορίζεται παρακάτω: $Q(S_t, A_t) = Q(S_t, A_t) + a * [R_{t+1} + \gamma * \max_a Q(S_{t+1}, a) - Q(S_t, A_t)]$

Σε αυτήν την περίπτωση, η προσεγγιστική συνάρτηση δράσης-κατάστασης Q, προσεγγίζει κατευθείαν την ιδανική συνάρτηση Q , ανεξαρτήτως της πολιτικής που ακολουθείται. Η συγκεκριμένη προσέγγιση, απλοποιεί δραματικά την ανάλυση του αλγορίθμου. Η πολιτική εξακολουθεί να έχει επίδραση, καθώς επιλέγει ποια ζεύγη δράσης-κατάστασης θα ενημερωθούν. Παρόλα αυτά, αυτό που απαιτείται για ορθή σύγκλιση, είναι όλα τα ζευγάρια να συνεχίζουν να ανανεώνονται

Παρακάτω παραθέτουμε αναλυτικά και τον ψευδοκώδικα της off-policy μεθόδου μάθησης-Q:

Ψευδοκώδικας μεθόδου μάθησης-Q

Initialize Q(s,a) for every s in S and a in A(s), arbitrarily, and Q(terminal-state,.)=0

Repeat for each episode:
Initialize S Choose A from S using policy derived from Q (e.g.,ε-greedy)
    Initialize S
    Repeat (for each step of episode):
        Take acition A, observe R,S'
        Choose A' from S' using policy derived from Q(e.g.,ε-greedy)
        $Q(S,A) = Q(S,A) + a * [R + \gamma * \max_a Q(S',a) - Q(S,A)]$
        S=S'
    until S is terminal

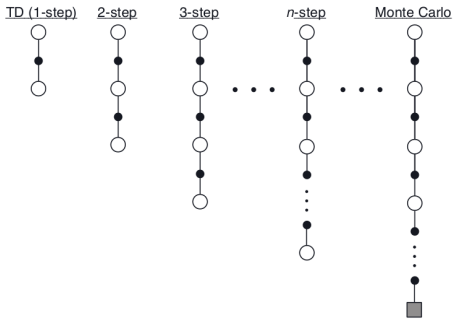
[27]

3.3.7 Πολιτικές Χρονικών Διαφορών n βημάτων (n-step Temporal-Difference (TD) Learning)

Η μέθοδος TD n βημάτων (TD(n)) αποτελεί μία μέθοδο της κατηγορίας, η οποία περιέχει την ιδιαίτερα διαδεδομένη τεχνική στην ενισχυτική μάθηση, την τεχνική των ιχνών επιλεξιμότητας eligibility traces. Στον αλγόριθμο TD(n), η παράμετρος n αποτελεί το ίχνος επιλεξιμότητας. Σχεδόν κάθε μέθοδος χρονικής διαφοράς (TD), όπως η μάθηση Q ή η μέθοδος Sarsa, μπορεί να συνδυαστεί με ίχνη καταλληλότητας για να ληφθεί μια γενικότερη μέθοδος που μπορεί να μάθει πιο αποτελεσματικά.[27]

Στην παρούσα υποενότητα γεφυρώνουμε την απόσταση μεταξύ των μεθόδων (TD) και MC μέσω της τεχνικής των ιχνών επιλεξιμότητας, βλέπε εικόνα 3.13. Συνήθως, τόσο οι μέθοδοι (TD) όσο και MC δεν παρέχουν την καλύτερη δυνατή επίλυση του προβλήματος. Σε αυτό το σημείο λύση στο πρόβλημα έρχεται να δώσει ο αλγόριθμος TD(n). Στις επόμενες παραγράφους της παρούσας υποενότητας θα τον παρουσιάσουμε αναλυτικά. Οι μέθοδοι TD(n) γενικεύουν και τις δύο μεθόδους έτσι ώστε να μπορεί κάποιος να μεταβαίνει σταδιακά από την μία στην άλλη όπως απαιτείται για να ικανοποιήσει τις προσαρμοσμένες απαιτήσεις μιας συγκεκριμένης εργασίας.

Ας δούμε σε αυτό το σημείο πιο αναλυτικά. Οι μέθοδοι MC διατηρούν ένα αντίγραφο ασφαλείας (backup) της συνάρτησης κατάστασης για κάθε κατάσταση με βάση ολόκληρη τη σειρά των παρατηρούμενων ανταμοιβών από αυτήν



Εικόνα 3.13: Μετάβαση από τις TD μεθόδους στις MC μεθόδους [27]

την εκάστοτε τρέχουσα κατάσταση μέχρι το τέλος του επεισοδίου. Το αντίγραφο ασφαλείας των απλών μεθόδων TD, από την άλλη πλευρά, βασίζεται μόνο στη μία επόμενη ανταμοιβή, χρησιμοποιώντας την τιμή της συνάρτηση κατάστασης ένα βήμα αργότερα ως βοηθητική συνάρτηση για τις υπόλοιπες ανταμοιβές. Ένα είδος ενδιάμεσης μεθόδου αποτελεί ο αλγόριθμος TD(n), ο οποίος διατηρεί ένα αντίγραφο ασφαλείας με βάση έναν ενδιάμεσο αριθμό ανταμοιβών: περισσότερες από μία, αλλά λιγότερες από όλες μέχρι τον τερματισμό.[27]

Οι μέθοδοι που χρησιμοποιούν αντίγραφα ασφαλείας n-βημάτων εξακολουθούν να είναι μέθοδοι TD επειδή εξακολουθούν να αλλάζουν μια προηγούμενη εκτίμηση με βάση το πώς διαφέρει από μια μεταγενέστερη εκτίμηση. Τώρα η μεταγενέστερη εκτίμηση δεν είναι ένα βήμα αργότερα, αλλά n βήματα μεταγενέστερα. Οι μέθοδοι στις οποίες η χρονική διαφορά εκτείνεται σε n βήματα ονομάζονται μέθοδοι TD n-βημάτων. Οι μέθοδοι TD που εισήχθησαν στο προηγούμενο κεφάλαιο χρησιμοποιούν όλες αντίγραφα ασφαλείας ενός βήματος και γι'αυτό το λόγο τις ονομάζουμε μεθόδους TD ενός βήματος. Οι ενημερώσεις που πραγματοποιούνται στις εξισώσεις TD(n) πραγματοποιούνται μέσω της παρακάτω σχέσης:

$$G_{t:t+n} = R_{t+1} + \gamma * R_{t+2} + \gamma^2 * R_{t+3} + ... + \gamma^n 1 * R_{t+n} + \gamma^n * V_{t+n1}(S_{t+n}) \forall n \geq 1$$

Ονομάζουμε τη συγκεκριμένη ποσότητα ως στόχο της ενημέρωσης. Η ποσότητα $G_{t:t+n}$ θεωρείται μια αποκομμένη επιστροφή για το χρόνο t χρησιμοποιώντας ανταμοιβές μέχρι τη χρονική στιγμή t+n. Με απλά λόγια αυτή είναι η βασική ιδέα πίσω από τους αλγορίθμους TD(n)[38]. Αξίζει να σημειώσουμε ότι μπορούν να υπάρξουν πολλές παραλλαγές του συγκεκριμένου αλγορίθμου, αλλά δε θα επεκταθούμε στην παρούσα διπλωματική.

[27]

3.3.8 Συμπεράσματα πάνω στις μεθόδους ενισχυτικής μάθησης που περιγράψαμε

Με όλες αυτές τις μεθόδους που περιγράψαμε, με σχετική συντομία, στο παρόν υποκεφάλαιο θεωρούμε ότι παρέχουμε στον αναγνώστη μία ολοκληρωμένη εισαγωγή στις έννοιες τις ενισχυτικής μάθησης. Προφανώς κάθε μέθοδος ενισχυτικής μάθησης μπορεί να αναλυθεί μεμονωμένα σε μεγαλύτερο βάθος. Επίσης, πρέπει να αναφέρουμε ότι υπάρχουν και άλλοι αλγόριθμοι ενισχυτικής μάθησης, τους οποίους σκόπιμα δεν αναφέρουμε, καθώς δεν τους έχουμε χρησιμοποιήσει στο πρακτικό κομμάτι της παρούσας διπλωματικής. Γι' αυτό το λόγο παραπέμπουμε τους ενδιαφερόμενους στη σχετική βιβλιογραφία, αλλά και σε σχετικές δημοσιεύσεις στο διαδίκτυο.

Κεφάλαιο 4

Εισαγωγή στη Βαθιά Ενισχυτική Μάθηση

Στο προηγούμενο κεφάλαιο περιγράψαμε αναλυτικά κάποιους βασικούς ορισμούς πάνω στις έννοιες μηχανική μάθηση. Στη συνέχεια, παραθέσαμε μία εκτενή εισαγωγή πάνω στις έννοιες της βαθιάς μηχανικής μάθησης. Στο συγκεκριμένο κεφάλαιο εισάγουμε τον αναγνώστη πάνω στο αντικείμενο της βαθιάς ενισχυτικής μάθησης Reinforcement Learning (RL). Η βαθιά ενισχυτική μάθηση αποτελεί το συνδυασμό ενιχυτικής μάθησης (RL) και βαθιάς μάθησης. Αυτό το πεδίο έρευνας κατάφερε να επιλύσει ένα ευρύ φάσμα σύνθετων αποφάσεων ή και προβλημάτων γενικότερα, ολοκληρώνοντας εργασίες που προηγουμένως δεν ήταν εφικτές για ένα μηχάνημα. Με αυτόν τον τρόπο, η βαθιά ενισχυτική μάθηση ανοίγει πολλές νέες εφαρμογές σε τομείς, όπως η υγειονομική περίθαλψη, η ρομποτική, τα έξυπνα δίκτυα, τα χρηματοοικονομικά καθώς και πολλούς άλλους.

4.1 Βαθιά Ενισχυτική Μάθηση

Παρόλο που η ενισχυτική μάθηση (RL) είχε κάποιες επιτυχίες στο παρελθόν, οι προηγούμενες προσεγγίσεις δεν είχαν δυνατότητα κλιμάκωσης και περιορίζονταν εγγενώς σε προβλήματα αρκετά χαμηλών διαστάσεων. Αυτοί οι περιορισμοί υπάρχουν επειδή οι αλγόριθμοι RL μοιράζονται τα ίδια προβλήματα πολυπλοκότητας με άλλους αλγόριθμους: πολυπλοκότητα μνήμης, υπολογιστική πολυπλοκότητα και, στην περίπτωση αλγορίθμων μηχανικής μάθησης, πολυπλοκότητα δειγμάτων. Τα τελευταία χρόνια παρατηρούμε την ανάδυση της βαθιάς μάθησης, η οποία στηρίζεται στις ισχυρές συναρτήσεις προσέγγισης και αναπαράστασης των μαθησιακών ιδιοτήτων που παρέχει η τεχνολογία των βαθιών νευρωνικών δικτύων. Η χρήση της συγκεκριμένης τενχολογίας φαίνε-

ται να επιλύει τα συγκεκριμένα προβλήματα. Ταυτόχρονα, όμως, δημιουργεί τεράστια ηθικά ζητήματα.

Η έλευση της βαθιάς μάθησης είχε σημαντικό αντίκτυπο σε πολλούς τομείς της μηχανικής μάθησης, βελτιώνοντας σημαντικά state of the art εργαλεία της τεχνητής νοημοσύνης, όπως η ανίχνευση αντικειμένων, η αναγνώριση ομιλίας και η μετάφραση ανθρώπινης γλώσσας. Η πιο σημαντική ιδιότητα της βαθιάς μάθησης είναι ότι τα βαθιά νευρωνικά δίκτυα μπορούν να βρουν αυτόματα συμπαγείς χαμηλών διαστάσεων αναπαραστάσεις (χαρακτηριστικά) δεδομένων υψηλής διάστασης (π.χ. εικόνες, κείμενο και ήχο). Η βαθιά μάθηση έχει επίσης επιταχύνει την πρόοδο στον τομέα της ενισχυτικής μάθησης, με τη χρήση αλγορίθμων deep learning στην ενισχυτική μάθηση που ορίζουν το πεδίο του DRL-Deep Reinforcement Learning. Στα επόμενα υποκεφάλαια θα εξετάσουμε κάποιες από τις μεθόδους της βαθιάς ενισχυτικής μάθησης.[8]

4.2 Value-based μέθοδοι στη βαθιά ενισχυτική μάθηση

Η value-based κλάση αλγορίθμων έχει ως στόχο να κατασκευάσει μία συνάρηση τιμών, η οποία θα μας οδηγήσει στη εύρεση της ζητούμενης συνάρτησης πολιτικής. Περιγράψαμε στο προηγούμενο υποκεφάλαιο έναν από τους διάσημους αλγορίθμους, τον αλγόριθμο της μάθησης-Q (Q-Learning). Σε αυτό το υποκεφάλαιο θα ασχοληθούμε με μία παραλλαγή του, τον αλγόριθμο Deep Q-Learning (DQN), ο οποίος έχει ξεπεράσει τις ανθρώπινες δυνατότητες στο χειρισμό της παιχνιδομηχανής Atari. Στη συνέχεια θα αναλύσουμε και τη βελτίωση του αλγορίθμου DQN, τον αλγόριθμο Double DQN.

4.2.1 Αλγόριθμος DQN

Για να γίνουμε πιο κατανοητοί, θα υπενθυμίσουμε πρώτα στον αναγνώστη κάποιες έννοιες από τη θεωρία της ενισχυτικής μάθησης και στην ροή του παρόντος υποκεφαλαίου θα αναλύσουμε τον αλγόριθμο DQN. Στην παρούσα διπλωματική εργασία εκπαιδεύσαμε όλα τα προγράμματά μας πάνω στο απλό παιχνίδι Cart-pole, το οποίο πληροφορούσε τους αλγορίθμους για την κατάσταση του περιβάλλοντος χωρίς να χρειαστεί η αναπαράσταση του παιχνιδιού σε πίξελ. Όμως για να παρέχουμε στον αναγνώστη μία πιο σφαιρική εικόνα του αλγορίθμου DQN θα περιγράψουμε τον αλγόριθμο DQN στα πλαίσια του περιβάλλοντος, $\mathcal{E}$, Arcade Learning Environment, το οποίο είναι ένα object-oriented framework. Το συγκεκριμένο περιβάλλον επιτρέπει σε ερευνητές και χομπίστες να αναπτύσουν AI πράκτορες για τα Arcade 2600 games. Θεωρούμε τις διεργασίες στις οποίες ο πράκτορας αλληλεπιδρά με το περιβάλλον,

$\mathcal{E}$. Στη δική μας περίπτωση το περιβάλλον είναι ο προσομοιωτής Atari, ο οποίος παρέχει μία ακολουθία από καταστάσεις states, δράσεις actions και ανταμοιβές rewards. Σε κάθε χρονικό βήμα ο πράκτορας επιλέγει μία δράση a_t από ένα σύνολο επιτρεπόμενων δράσεων $A_t = 1, .., K$. Η συγκεκριμένη δράση, a_t, περνά στον προσομοιωτή και αλλάζει την εσωτερική του κατάσταση και το σκορ του παιχνιδιού. Στη γενική περίπτωση το περιβάλλον, $\mathcal{E}$, μπορεί να είναι στοχαστικό. Η εσωτερική κατάσταση του προσομοιωτή δεν μπορεί να παρατηρηθεί από τον πράκτορα. Ο πράκτορας παρατηρεί μία εικόνα $x_t \in R_d$ από τον προσομοιωτή. Η συγκεκριμένη εικόνα αναπαρίσταται από έναν πίνακα από τιμές των πίξελ, οι τιμές των οποίων αντιστοιχούν στην τρέχουσα εικόνα του προσομοιωτή. Επιπλέον, δέχεται σήματα ανταμοιβής r_t, τα οποία αναπαριστούν το σκορ του παιχνιδιού. Αξίζει να σημειώσουμε ότι στη γενικότητα το σκορ ενός παιχνιδιού μπορεί να εξαρτάται από προηγούμενες δράσεις και παρατηρήσεις. Μπορεί να λάβουμε σήμα ανάδρασης για μία κίνηση ακόμα και όταν περάσει πολλές χιλιάδες βήματα στο παιχνίδι.

Στην περίπτωση του αλγορίθμου DQN ο πράκτορας λαμβάνει την τρέχουσα εικόνα x_t, η οποία, όμως, δεν είναι αρκετή για να κατανοήσει ο πράκτορας την τρέχουσα κατάσταση. Για να γνωρίζει ο πράκτορας την τρέχουσα κατάσταση πρέπει να γνωρίζει ακολουθίες από δράσεις και παρατηρήσεις, προέκυψε η τρέχουσα κατάσταση, $s_t = x_1, a_1, x_2, a_2, ..., a_{t-1}, x_t$. Μαθαίνει στρατηγικές παιχνιδιού με βάση αυτές τις ακολουθίες. Έχουμε πραγματοποιήσει την υπόθεση ότι όλες οι ακολουθίες στον προσομοιωτή τερματίζουν σε πεπερασμένο αριθμό βημάτων. Η συγκεκριμένη τυποποίηση μας επιτρέπει να χρησιμοποιήσουμε κάθε ακολουθία, s_t, σαν μία κατάσταση ενός Markov Decision Process. Με αυτόν τον τρόπο μπορούμε να χρησιμοποιήσουμε τις γνωστές μεθόδους ενισυχτικής μάθησης πάνω σε MDPs, θεωρώντας ως καταστάσεις τις ακολουθίες s_t για κάθε χρονική στιγμή t.

Όπως έχουμε αναφέρει, προηγουμένως, ο στόχος του πράκτορα είναι να αλληλεπιδρά με τον προσομοιωτή με τρόπο τέτοιον ώστε να επιλεγούν δράσεις που να μεγιστοποιούν τη μελλοντική ανταμοιβή. Πραγματοποιούμε και στον αλγόριθμο DQN τη γνωστή υπόθεση, ότι, δηλαδή, οι μελλοντικές ανταμοιβές πολλαπλασιάζονται με έναν εκπτωτικό πράγοντα γ. Με αποτέλεσμα η μελλοντική συνολική ανταμοιβή να λαμβάνει τη μορφή $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} * r'_t$, όπου T είναι το χρονικό βήμα τερματισμού. Ορίζουμε τη βέλτιστη συνάρτηση κατάστασης-δράσης $Q^*(s, a)$ σα τη μέγιστη δυνατή αναμενόμενη επιστροφή, ακολουθώντας οποιαδήποτε στρατηγική. Πιο επεξηγηματικά, όταν ο πράκτορας έχει παρακολουθήσει μία ακολουθία παρατηρήσεων, βρίσκεται δηλαδή στην κατάσταση s_t και λαμβάνει τη δράση a_t τότε η συνάρτηση κατάστασης-δράσης είναι $Q^*(s, a) = \max_\pi E[R_t|s_t = s, a_t = a, \pi]$, όπου π είναι η πολιτική που μεγιστοποιεί τη μελλοντική συνολική ανταμοιβή.

Η βέλτιστη συνάρτηση πράξης αξίας ακολουθεί την εξίσωση Bellman. Θα μπορούσαμε διαισθητικά να περιγράψουμε την εξίσωση Bellman ως εξής: Αν γνωρίζαμε τη βέλτιστη τιμή $Q^*(s', a')$ για κάθε δυνατή πράξη a' στην επόμενη κατάσταση s', τότε η βέλτιστη στρατηγική είναι η επιλογή της πράξης a', η οποία μεγιστοποιεί την αναμενόμενη τιμή της έκφρασης $R + \gamma * Q^*(s, \alpha)$.

$$Q^*(s, a) = E_{s' \sim S}[R + \gamma * max_{a'} Q(s', \alpha')|s, \alpha]$$

Πολλοί αλγόριθμοι ενισχυτικής μάθησης χρησιμοποιούν την εξίσωση Bellman για να υπολογίσουν τη συνάρτηση πράξης-αξίας. Η συνάρτηση $Q^*(s, a) = E_{s' \sim S}[R + \gamma * max_{a'} Q(s', \alpha')|s, \alpha]$ ενημερώνεται επαναληπτικά και οι αλγόριθμοι συγκλίνουν στη βέλτιστη λύση $Q_i \sim Q^*$ όταν $i \Longrightarrow \inf$. Η προσέγγιση αυτή δεν επιτυγχάνει πάντα στην πράξη. Γι'αυτόν το λόγο έχουν εισαχθεί οι προσεγγιστικές συναρτήσεις τιμών για την εκτίμηση της συνάρτησης $Q(s, a; \theta) \approx Q$.

Η συγκεκριμένη τεχνική των προσεγγιστικών συναρτήσεων τιμών χρησιμοποιείται εκτεταμένα στον αλγόριθμο DQN. Αξίζει να σημειώσουμε ότι με τον όρο προσεγγιστική συνάρτηση τιμών εννοούμε ένα νευρωνικό δίκτυο με βάρη ϑ. Το ονομάζουμε Q-δίκτυο. Η εκπαίδευση του δικτύου επιτυγχάνεται με με την ελαχιστοποίηση των loss functions $L_i(\theta_i)$, οι οποίες αλλάζουν με κάθε επανάληψη i.

$$L_i(\theta_i) = E_{\rho(s,a)}[(y_i - Q(s, a; \theta))^2]$$

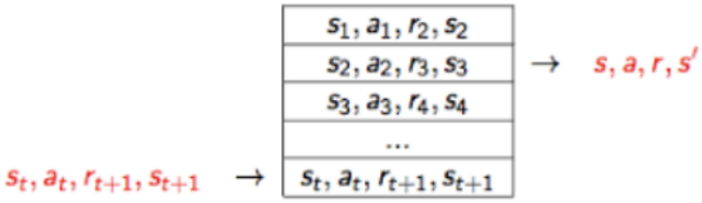
όπου $y_i = E_{s' \sim S}[R + \gamma * max_{a'} Q(s', \alpha'; \theta_{i-1})|s, \alpha]$ είναι ο στόχος της επανάληψης i και η ρ(s, α) είναι μια κατανομή πιθανότητας συναρτήσει των καταστάσεων και των πράξεων, την οποία ονομάζουμε κατανομή συμπεριφοράς. Αξίζει να σημειώσουμε ότι παράμετροι από τις προηγούμενες επαναλήψεις, παραμένουν σταθερές καθώς βελτιστοποιούμε την loss function.

Ο αλγόριθμος DQN είναι model-free αλγόριθμος. Επιχειρεί να επιλύσει το πρόβλημα χρησιμοιποιώντας μόνο δείγματα από το περιβάλλον και δίχως να κατασκευάζει κάποια εκτίμηση του περιβάλλοντος. Επίσης, αποτελεί έναν off-policy αλγόριθμο. Ακολουθεί μία άπληστη στρατηγική , $a = max_a Q(s, a; \theta)$, ενώ με πιθανότητα 1-ε επιλέγει κάποια τυχαία πράξη.[39]

Πριν παραθέσουμε τον αλγόριθμο θα περιγράψουμε αναλυτικά δύο βασικές καινοτομίες, οι οποίες τον καθιστούν τόσο πετυχημένο:

Target network

Όπως γίνεται εμφανές από τις παραπάνω εξισώσεις, καθώς πραγματοποιείται ενημέρωση των παραμέτρων ϑ οι τιμές στόχοι των Q-values, Q(s,a), ενημερώνονται για όλες τις δυνατές δράσεις και όχι μόνο για τη δράση που επιλέχθηκε. Οι τιμές στόχοι-ετικέτες εκπαίδευσης, δηλαδή, διαρκώς μετακινούνται αλλάζοντας τιμές. Η συγκεκριμένη συμπεριφορά είναι μη αποδεκτή στη βαθιά μάθηση.



Εικόνα 4.1: Experience Replay Buffer

Η συνεχής μετακίνηση του στόχου μπορεί να δημιουργήσει βρόγχους ανατροφοδότησης, με αποτέλσμα το νευρωνικό δίκτυο να μη συγκλίνει. Με τη χρήση της τεχνικής των target networks ο αλγόριθμος DQN επιλύει το πρόβλημα.

Τα target networks παρουσιάζουν την ίδια ακριβώς αρχιτεκτονική νευρωνικών δικτύων με τα δίκτυα prediction networks, με τη μόνη διαφορά ότι έχουν διαφορετικές τιμές παραμέτρων. Οι παράμετροι, θ, του prediction network ενημερώνονται σε κάθε επανάληψη του βρόχου, loop, του αλγορίθμου. Σε αυτές εφαρμόζεται σε κάθε επανάληψη ο αλγόριθμος Gradient Descent για τη ενημέρωση των παραμέτρων. Το prediction network, όμως, δε χρησιμοποιείται για τη πρόβλεψη των τιμών στόχων Y. Κάθε τακτό χρονικό διάστημα C οι τιμές των πραμετρων , θ^-, του δικτύου target network λαμβάνουν τις τιμές των παραμέτρων, ϑ, του prediction network. Η συγκεκριμένη διαδικασία βοηθάει το δίκτυο να συγκλίνει, καθώς η μεταβλητή στόχος Y παραμένει σταθερή για C επαναλήψεις.

Experience Replay

Η υψηλή συσχέτιση εικόνων από διαδοχικές καταστάσεις αποτελεί ένα σημαντικό πρόβλημα για την εκπαίδευση των νευρωνικών δικτύων. Επιζητούμε ο πράκτορας να λαμβάνει εμπειρίες ανεξάρτητες μεταξύ τους, ώστε να επιτευχθεί η καλύτερη δυνατή εκπαίδευση του δικτύου. Κάθε εμπειρία θα είναι μία πλειάδα, tuple, τεσσάρων στοιχείων. Θα περιέχει την αρχική κατάσταση s, την επιλεγμένη δράση, a, την άμεση ανταμοιβή r και την επόμενη κατάσταση s'. Οι εμπειρίες, θα είναι της μορφής $< s, a, r, s' >$. Οι συγκεκριμένες εμπειρίες θα επιλέγονται τυχαία και ομοιόμορφα από έναν buffer-χώρο αποθήκευσης. Με την ομάδα εμπειριών, mini batch, που θα επιλέγεται κάθε φορά θα εκπαιδεύεται το δίκτυο. Με την προσθήκη νέων εμπειριών στον αποθηκευτή, οι παλιές θα διαγράφονται.

Ψευδοκώδικας αλγορίθμου DQN with experience replay

Initialize replay memory D to capacity N
Initialize action-value function Q with random weights
for episode = 1,... M do
Initialise sequence $s_1 = x_1$ and preprocessed sequenced $\varphi_1 = \varphi(s_1)$
for t = 1,... T do
With probability 1-ε select aν random action based on a greedy policy a_t
With probability ε select a random action a_t
Execute action a_t in emulator and observe reward r_t and image x_{t+1}
Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\varphi_{t+1} = \varphi(s_{t+1})$
Store transition $(\varphi_t, a_t, r_t, \varphi_{t+1})$ in D
Sample random minibatch of transitions $(\varphi_j, a_j, r_j, \varphi_{j+1})$ from D
Set

$$y_j = \begin{cases} r_j & for \quad terminal \quad \varphi_{tj+1} \\ r_j + \gamma * \max_{a_0} Q(\varphi_{j+1}, a_0; \theta^-) & for \quad non \quad terminal \quad \varphi_{tj+1} \end{cases}$$

Perform a gradient descent step on $(y_j - Q(\varphi_j, a_j; \theta))^2$ according to equation
Every C steps perform $\theta^- = \theta$
end for
end for

Πλεονεκτήματα αλγορίθμου DQN με experience replay:

- Ο αλγόριθμος DQN αποτελεί βελτίωση του αλγορίθμου fitted Q-learning (NFQ), ο οποίος αποτελούσε ό,τι πιο κοντινό υπήρχε στον αλγόριθμο DQN. Καταφέρνει να ελέγχει επιτυχημένα τον πράκτορα δεχόμενο εισόδους κατευθείαν από δεδομένα αισθητήρων μεγάλων διαστάσεων, όπως δεδομένα όρασης και φωνής, η διαχείρηση των οποίων αποτελεί μεγάλη πρόκληση για τη βαθιά ενισχυτική μάθηση. Σε σχέση με τον αλγόριθμο fitted Q-learning (NFQ) το υπολογιστικό κόστος για κάθε batch update είναι πολύ χαμηλότερο. Επίσης, στον αλγόριθμος DQN το νευρωνικό δίκτυο δέχεται κατευθείαν υψηλής διάστασης δεδομένα, όπως παράγονται και όχι ύστερα από προεπεξεργασία, περίπτωση fitted Q-learning (NFQ). Με αυτόν τον τρόπο το δίκτυο του αλγορίθμου DQN μπορεί να "αντιληφθεί" περισσότερα χαρακτηριστικά.

- Η βαθιά ενιχυτική μάθηση για να επιτύχει απαιτεί μεγάλο αριθμό από hand-labelled δεδομένα. Η ενισχυτική μάθηση από την άλλη πλευρά

επιχειρεί την εκμάθηση του πράκτορα, μέσω ενός βαθμωτού, θορυβώδους και συχνά αραιού σήματος ανταμοιβής. Ο αλγόριθμος DQN συνδυάζει επιτυχημένα τις δύο αυτές κατηγορίες μάθησης. Παρόλο που ο στόχος δεν είναι σταθερός, μπορεί ο αλγόριθμος και εκπαιδεύει το νευρωνικό δίκτυο κατάλληλα.

- Ο αλγόριθμος DQN καταφέρνει να λύσει το πρόβλημα της υψηλής συσχέτισης εικόνων από διαδοχικές καταστάσεις. Εισάγει την έννοια του experience replay γι' αυτόν το λόγο.

- Το νευρωνικό δίκτυο στόχος, target network, $Q(s, a; \theta_k^-)$ αντικαθίσταται από το δίκτυο $Q(s, a; \theta_k^-)$ στην εξίσωση της συνάρτησης απωλειών. Όπου οι μεταβλητλές θ_k^- ενημερώνονται κάθε $C \in N$ επαναλήψεις, μέσω της αντικατάστασης $\theta_k^- = \theta$. Η συγκεκριμένη διαδικασία προφυλάσει βοηθάει το δίκτυο να συγκλίνει στο ολικό βέλτιστο και όχι να εγκλωβιστεί σε τοπικά βέλτιστα, καθώς η μεταβλητή στόχος Y_k^Q παραμένει σταθερή για C επαναλήψεις.

- Όπως έχουμε αναφέρει ο αλγόριθμος DQN αποθηκεύει τις πληροφορίες από τα τελευταία N βήματα του πράκτορα, όπου η πληροφορία συγκεντρώνεται ακολουθώντας ε-greedy πολιτική. Οι ενημερώσεις πραγματοποιούνται από την τυχαία επιλογή, random choice, ενός συνόλου από πλειάδες $< s, a, r, s' >$, το οποίο ονομάζεται και mini-batch. Η πρακτική αυτή επιτρέπει στον αλγόριθμο να πραγματοποιεί ενημερώσεις βασιζόμενος σε ένα μέγαλο εύρος δειγμάτων κατάστασης-δράσης. Με αυτόν τον τρόπο μία mini-batch ενημέρωση προκαλεί λιγότερη variance σε σχέση με αυτήν που δημιουργεί η απλή ενημέρωση από μόνο ένα tuple ,$< s, a, r, s' >$. Επίπλέον, η τεχνική επιτρέπει να πραγματοποιούμε ποσοτικά μεγαλύτερες ενημερώσεις στις μεταβλητές του νευρωνικού δικτύου, ενώ ταυτόχρονα παραλληλοποιεί αποτελεσματικά τον αλγόριθμο.

4.2.2 Αλγόριθμος Double DQN

Είναι γνωστό ότι ο συντελεστής max στη βασική εξίσωση του αλγορίθμου της Q-μάθησης χρησιμοποιεί τις ίδιες τιμές τόσο για να επιλέξει πράξη όσο και για να εκτιμήσει την πράξη αυτή. Η συγκεκριμένη λειτουργία του αλγορίθμου έχει ως αποτέλεσμα να επιλέγονται υπερεκτιμημένες πράξεις σε περιπτώσεις

ανακριβειών ή θορύβου με αποτέλεσμα να λαμβάνονται και υπερεκτιμημένες εκτιμήσεις τιμών. Γι' αυτόν το λόγο ο αλγόριθος DQN εμφανίζει μεροληψία προς τα πάνω στις τιμές του. Η συγκεκριμένη συμπεριφορά είναι πιθανό να εμφανιστεί, ειδικά στην αρχή της εκπαίδευσης, που δεν έχει ακόμα αρκετή πληροφορία ο πράκτορας. Η μέθοδος Double DQN, με διπλό εκτιμητή, χρησιμοποιεί δύο διαφορετικά νευρωνικά δίκτυα για να υπολογίσει την τιμή της μεταβλητής στόχου y_i. Τη μελλοντική πράξη a' την υπολογίζει με τη χρήση του prediction network που χρησιμοποιεί τις μεταβλητές θ_i, ενώ τη με τη χρήση του target network υπολογίζει την τιμή της συνάρτησης $Q(s', a'; \theta_{i-1})$. Πιο συγκεκριμένα η εξίσωση στην περίπτωση του Double DQN λαμβάνει την παρακάτω μορφή:

$$y_i = [R + \gamma * Q(s', argmax_{a' \in A} Q(s', a, \theta_i); \theta_{i-1})]$$

Η συγκεκριμένη πρακτική οδηγεί σε χαμηλότερη υπερεκτίμηση των τιμών Q-learning, ενώ βελτιώνει και την ευστάθεια του αλγορίθμου με αποτέλεσμα να βελτιώνεται η απόδοση του αλγορίθμου. Σε σχέση με τον κλασσικό DQN αλγόριθμο, το νευρωνικό δίκτυο στόχος με βάρη τα θ_{i-1} χρησιμοποιείται μόνο για την εκτίμηση της τιμής Q-value, χρησιμοποιώντας την πράξη a που έλαβε από το prediction network. Τόσο το prediction network όσο και το target network εμφανίζουν ίδια αρχιτεκτονική. [11]

4.2.3 Αλγόριθμος Dueling DQN

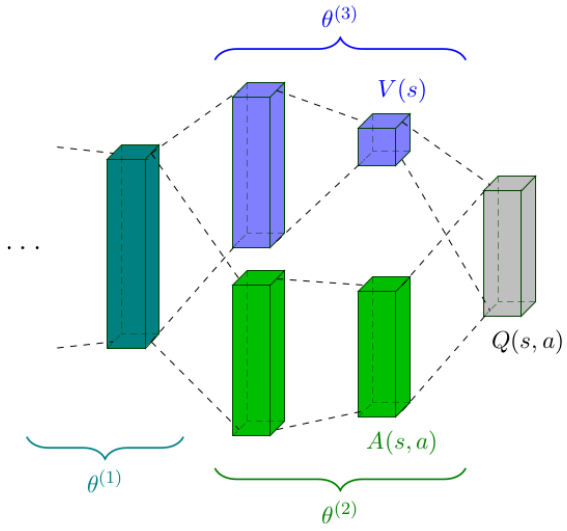
Μία, ακόμα, βελτίωση του αλγορίθμου DQN αποτελεί ο αλγόριθμος Dueling DQN. Στον αλγόριθμο Dueling DQN χρησιμοποιούνται ξανά οι συναρτήσεις της ενισχυτικής μάθησης $Q(s, a)$ και $V(s)$. Ορίζεται, όμως, και μία νέα συνάρτηση η $A_\pi(s, a)$, η οποία ονομάζεται συνάρτηση πλεονεκτήματος και οδηγεί το σύστημα σε βελτιωμένες επιδόσεις.

$$A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s)$$

Η συγκεκριμένη συνάρτηση, εκφράζει πόσο χειρότερη είναι η κάθε δράση a, από την καλύτερη δράση a^* στην κατάσταση s, δεδομένης μίας πολιτικής π. Η συνάρτηση Q-value του αλγορίθμου Dueling DQN δίνεται από τη σχέση:

$$Q(s, a; \theta_{(1)}, \theta_{(2)}, \theta_{(3)}) = V(s; \theta_{(1)}, \theta_{(3)}) + (A(s, a; \theta_{(1)}, \theta_{(2)}) - max_{a' \in A} A(s, a'; \theta_{(1)}, \theta_{(2)}))$$

Είναι εμφανές ότι για $a^* = argmax_{a' \in A} Q(s, a; \theta_{(1)}, \theta_{(2)}, \theta_{(3)})$ λαμβάνουμε ότι $Q(s, a^*; \theta_{(1)}, \theta_{(2)}, \theta_{(3)}) = V(s; \theta_{(1)}, \theta_{(3)})$. Επιπλέον, παρατηρούμε ότι η ροή $V(s; \theta_{(1)}, \theta_{(3)})$ στην αρχιτεκτονική Dueling DQN παρέχει μία εκτίμηση της



Εικόνα 4.2: Dueling Dqn Network

συνάρτησης τιμής, ενώ η άλλη ροή παρέχει μία εκτίμηση ης συνάρτησης πλεονεκτήματος.

Στην πράξη προτιμάται η παρακάτω προσέγγιση της αρχετικτονικής Dueling DQN:

$$Q(s,a;\theta_{(1)},\theta_{(2)},\theta_{(3)}) = V(s;\theta_{(1)},\theta_{(3)}) + (A(s,a;\theta_{(1)},\theta_{(2)} - (1/|A|) * \sum_{a' \in A} A(s,a';\theta_{(1)},\theta_{(2)}))$$

Βέβαια, με τη συγκεκριμένη προσέγγιση της αρχετεκτονικής Dueling DQN χάνεται η έννοια των όρων $V(s)$ και $A_\pi(s,a)$. Όμως, η συγκεκριμένη εναλλακτική βελτιώνει την ευστάθεια του αλγορίθμου. [11]

4.3 Policy gradient μέθοδοι στη βαθιά ενισχυτική μάθηση

Σε αυτήν την υποενότητα θα επικεντρωθούμε σε μία άλλη οικογένεια αλγορίθμων της βαθιάς ενισχυτικής μάθησης, τις policy gradient μεθόδους. Οι συγκεκριμένες μέθοδοι βελτιστοποιούν έναν στόχο (objective), συνήθως ως στόχος εννοείται η συσωρευτική ανταπόδοση cumulative reward. Ο στόχος βελτιστοποιείται μέσω της εύρεσης μίας καλής πολιτικής, (π.χ.neural network

parameterized policy), λόγω των διαφόρων στοχαστικών μεθόδων gradient ascent. Σημειώνουμε ότι οι μέθοδοι policy gradient ανήκουν σε μία ευρύτερη κλάση μεθόδων, τις policy-based μεθόδους. Σε αυτό το υποκεφάλαιο θα αναφερθούμε στο θεώρημα stochastic gradient, το οποίο παρέχει gradients στις παραμέτρους της εκάστοτε πολιτικής με σκοπό τη βελτιστοποίση της απόδοση του αλγορίθμου.

Stochastic Policy Gradient

Στις μεθόδους κλίσης πολιτικής, η πολιτική μπορεί να παραμετροποιηθεί, αρκεί η συνάρτηση $\pi(a|s,\theta)$ να είναι διαφορίσιμη σε σχέση με τις παραμέτρους της, δηλαδή, εφόσον υπάρχει ο όρος $\nabla_\theta \pi(a|s,\theta)$ και είναι πεπερασμένος. Συνήθως απαιτούμε η πολιτικής να μην είναι πλήρως ντετερμινιστική, να είναι στοχαστική, ώστε να επιτευχθεί η καλυτερή δυνατή εξερεύνηση από τον πράκτορα. Αξίζει να σημειώσουμε ότι οι μέθοδοι με παραμετροποιημένες πολιτικές εμφανίζουν ένα κύριο συγκριτικό πλεονέκτημα σε σχέση με τις ε-άπληστες μεθόδους. Με τη διαρκή πραμετροποίηση πολιτικής οι πιθανότητες δράσης αλλάζουν ομαλά ως συνάρτηση των παραμέτρων. Αντίθετα, στις ε-άπληστες μεθόδους οι πιθανότητες δράσης πολύ συχνά αλλάζουν καθοριστικά από μετά από την επιλογή του πράκτορα μη πιθανών ε-κινήσεων. Θεώρείται ότι οι μέθοδοι με παραμετροποιημένες πολιτικές εμφανίζουν ισχυρότερη σύγκλιση σε σχέση με τις ε-άπληστες μεθόδους.[38]

Θεώρημα Policy Gradient

Θα ξεκινήσουμε την ανάλυση του θεωρήματος Policy Gradient περιγράφοντας πρώτα μία μαθηματική βελτιστοποίηση: Υπάρχει ένα γνωστό "τρικ" στους τομείς του DL και του RL. Η μερική παράγωγος μία συνάρτησης $f(x)$ ισούται με το γινόμενο της συνάρτησης $f(x)$ επί τη μερική παράγωγο της συνάρτησης $log(f(x))$. Πιο συγκεκριμένα παραθέτουμε παρακάτω τη συγκεκριμένη σχέση:

$$f(x) * \nabla_\theta log(f(x)) = f(x) * \nabla_\theta f(x)/f(x) = \nabla_\theta f(x)$$

Αν αντικαταστήσουμε τη συνάρτηση $f(x)$ με τη συνάρτηση πολιτικής π, τότε παίρνουμε την παρακάτω σχέση:

$$\pi(x) * \nabla_\theta log(\pi(x)) = \nabla_\theta \pi(x)$$

Για το συνεχή χώρο παίρνουμε ότι:

$$E_{x\sim p(x)}[f(x)] = \int (x) * p(x) dx$$

Σε αυτό το σημείο θα επιχειρήσουμε να μοντελοποιήσουμε τον μαθηματικό μας στόχο.

$$\theta^* = argmax_\theta E_{\tau \sim p_\theta(\tau)}[\sum_t (s_t, a_t)]$$

Με βάση το παραπάνω τρικ μπορούμε να ξαναγράψουμε τη συνάρτηση συνολικής ανταμοιβής ως εξής:

$$J(\theta) = E_{\tau \sim \pi_\theta(\tau)}[r(\tau)] = \int \pi_\theta(\tau) * r(\tau) d\tau$$

Είμαστε σε θέση να εκφράσουμε τώρα το θεώρημα policy gradient:

$$\nabla_\theta E[r(\tau)] = \nabla_\theta \int \pi_\theta(\tau) * r(\tau) d\tau = \int \nabla_\theta \pi_\theta(\tau) * r(\tau) d\tau$$

$$= \int \pi_\theta(\tau) * \nabla_\theta log(\pi_\theta(\tau)) * r(\tau) d\tau = E[r(\tau) * \nabla_\theta log(\pi_\theta(\tau))]$$

Δηλαδή ισχύει ότι:

$$\nabla_\theta E[r(\tau)] = E[r(\tau) * \nabla_\theta log(\pi_\theta(\tau))]$$

Με απλά λόγια το θεώρημα policy gradient λέει το εξής:

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

Με βάση το θεώρημα policy gradient παρατηρούμε ότι πλέον η μερική παράγωγος ως προς ϑ της συνάρτησης πολιτικής μπορεί να εκφραστεί και σα συνάρτηση της αναμενόμενης τιμής. Μπορούμε, δηλαδή να πραγματοποιήσουμε δειγματοληψία για να προσεγγίσουμε την αναμενόμενη τιμή.

Σε αυτό το σημείο θα εξηγήσουμε αναλυτικότερα τον όρο $\pi_\theta[(\tau)]$:

$$\pi_\theta[(\tau)] = P(s_0) * \prod_1^T \pi_\theta(a_t|s_t) p(s_{t+1}, R_{t+1}|s_t, a_t)$$

Ας περιγράψουμε σε αυτό το σημείο την παραπάνω σχέση. Εξηγούμε κάθε όρο ξεχωριστά. Με τον όρο $P(s_0)$ εννοούμε την κατανομή πιθανότητας εκκίνησης του πράκτορα σε κάποια κατάσταση s_0. Η επιλογή κάθε πράξης του πράκτορα είναι ανεξάρτητη από τις προηγούμενες καταστάσεις που βρέθηκε και εξαρτάται μόνο από τη κατάσταση που βρίσκεται την τρέχουσα στιγμή. Ο ισχυρισμός αυτός ισχύει γιατί είναι ιδιότητα των μαρκοβιανών αλυσίδων. Σε κάθε βήμα, πραγματοποιεί από ο πράκτορας μία δράση χρησιμοποιώντας την πολιτική π_θ και οδηγείται σε μια καινούργια κατάσταση. Οι όροι αυτοί πολλαπλασιάζονται για το χρονικό διάστημα T βημάτων, το οποίο αποτελεί και

το μήκος της συνολικής τροχιάς. Λογαριθμίζοντας λαμβάνουμε την παρακάτω σχέση:

$$log(\pi_\theta(\tau)) = log(P(s_0)) + \sum_1^T \pi_\theta(a_t|s_t) + \sum_1^T p(s_{t+1}, R_{t+1}|s_t, a_t) \Rightarrow$$

Όμως ο πρώτος και ο τελευταίος όρος της σχέσης δεν εξαρτώνται από τη μεταβλητή ϑ, οπότε παραγωγίζοντας ως προς ϑ διαγράφονται οι συγκεκριμένοι όροι. Λαμβάνουμε λοιπόν:

$$\nabla_\theta log(\pi_\theta(\tau))] = \nabla_\theta \sum_1^T \pi_\theta(a_t|s_t) \Rightarrow$$

$$E_{\pi_\theta(r(\tau))} = E_{\pi_\theta}(r(\tau) * \sum_1^T \pi_\theta(a_t|s_t))$$

Το παραπάνω αποτέλεσμα είναι εξαιρετικά χρήσιμο. Η παραπάνω σχέση μας λέει ότι δε χρειάζεται να γνωρίζουμε την κατανομή $P(s_0)$, ούτε τις κατανομες p, οι οποίες είναι δύσκολο να μοντελοποιηθούν. Μπορούμε να υπολογίσουμε την τιμή αναμενόμενης τιμής της συνολικής ανταμοιβής (επιστροφής) μόνο με την παραπάνω απλή σχέση και υπολογιστικά σύντομη. Αξίζει να σημειώσουμε ότι όλοι οι αλγόριθμοι, οι οποίοι χρησιμοποιούν το συγκεκριμένο αποτέλεσμα ονομάζονται Model-free αλγόριθμοι.

Με αυτόν τον τρόπο παίρνουμε την παρακάτω σχέση:

$$\nabla_\theta J = E_{\tau\ \pi_\theta(\tau)}[\nabla_\theta \pi_\theta(\tau) r(\tau)]$$

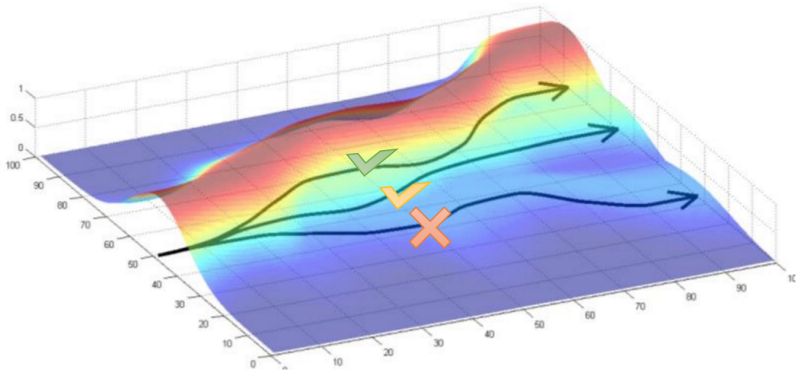
Προσεγγιστικά παίρνουμε ότι:

$$\nabla_\theta J = (1/N) * \sum_1^N \sum_1^T (\nabla_\theta log \pi_\theta(a_{i,t}|s_{i,t})) * \sum_1^T (r(s_{i,t}, a_{i,t}))$$

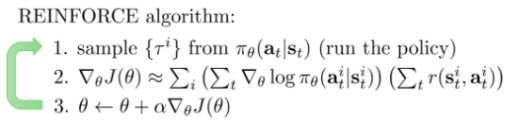
$$\theta \longleftarrow \theta + a * \nabla_\theta J$$

Χρησιμοποιούμε το συγκεκριμένο gradient για να ενημερωθεί το ϑ.

Πώς, όμως, μπορούμε να εξηγήσουμε την παραπάνω σχέση διαισθητικά; Με τον όρο $\nabla_\theta log \pi_\theta(a_{i,t}|s_{i,t})$ εννοούμε την έννοια maximum log likelihood που χρησιμοποιείται στη βαθιά μάθηση. Στη βαθιά μάθηση ο συγκεκριμένος όρος μετράει ποιά είναι η πιθανότητα να παρατηρηθούν τα δεδομένα. Στη δική μας περίπτωση μετράμε το πόσο πιθανό είναι να ακολουθηθεί από τον πράκτορα μία τροχιά κάτω από την τρέχουσα πολιτική. Πολλαπλασιάζοντας το συγκεκριμένο



Εικόνα 4.3: maximum log likelihood[14]



Εικόνα 4.4: Βασική ιδέα πίσω από τον αλγόριθμο REINFORCE σε ψευδοκώδικα

όρο με τις ανταμοιβές $r(s_{i,t}, a_{i,t})$ θέλουμε να αυξήσουμε τη πιθανοφάνεια μίας τροχιάς αν μας αποδίδει υψηλή συνολική ανταμοιβή. Από την ανάποδη θέλουμε να μειώσουμε τη πιθανοφάνεια μίας τροχιάς αν επιφέρει χαμηλές συνολικές ανταποδόσεις.[14]

Βλέπουμε στην εικόνα 4.3 με τις πιθανοφάνειες ότι πρέπει να επιλεγεί τροχιά που θα αποδίδει υψηλές ανταμοιβές.

4.3.1 Αλγόριθμος REINFORCE

Θα ξεκινήσουμε με τη βασική ιδέα πίσω από τον αλγόριθμο REINFORCE. Ο αλγόριθμος REINFORCE χρησιμοποιεί τα Monte Carlo rollouts για να υπολογίσει τις ανταμοιβές. Προσομοιώνει ο αλγόριθμος ολόκληρο το επεισόδιο για να υπολογίσει τις ανταμοιβές. Βλέπουμε στον αντίστοιχο ψευδοκώδικα, βλέπε εικόνα 4.4, την ιδέα πίσω από τον αλγόριθμο REINFORCE.

Βελτιώσεις πάνω στους αλγορίθμους Policy Gradient

Οι αλγόριθμοι Policy Gradient παρουσιάζουν εγγενή προβλήματα υψηλού variance και δυσκολία σύγκλισης. Οι αλγόριθμοι Monte Carlo βάζουν τον πράκτορα να πραγματοποιήσει ολόκληρη την τροχιά παρακοουθώντας τις ανταμοιβές καθ' όλη τη διάρκεια της τροχιάς. Παρόλα αυτά η στοχαστική φύση αυτών των αλγορίθμων μπορεί να έχει ως αποτέλεσμα ο πράκτορας, ακολουθών-

τας την ίδια πάντα πολιτική, να μπορεί να επιλέγει διαφορετικές δράσεις όταν βρίσκεται στην ίδια κατάσταση κάθε φορά. Αλλά μία μόνο επιλογή διαφορετικής κίνησης μπορεί να αλλάξει τελείως την τροχιά κάθε φορά. Με αυτόν τον τρόπο οι αλγόριθμοι Monte Carlo δεν εμφανίζουν bias, αλλά εμφανίζουν υψηλό variance. Το πρόβλημα του υψηλού variance αποτελεί σημαντικό θέμα για τις βελτιστοποιήσεις στη βαθιά μάθηση γενικά. Το πρόβλημα του υψηλού variance προκαλεί πολλαπλές descent directions κατά την εκμάθηση. Μία ανταμοιβή που έχουν πάρει με δειγματοληψία μπορεί να θέλει να αυξήσει τη λογαριθμική πιθανοφάνεια, ενώ μία άλλη ανταμοιβή μπορεί να θέλει να τη μειώσει. Αυτή, όμως, η "συμπεριφορά" καταστρέφει τη σύγκλιση του αλγορίθμου. Για να μειωθεί το variance που προκαλείται από την επιλογή διαφορετικής δράσης σε κάθε ίδια κατάσταση, επιχειρούμε να μειωθεί το variance μέσω των ανταμοιβών που δειγματοληπτούνται.

$$\sum_{1}^{T}(r(a_{i,t}, s_{i,t}))$$

Μία λύση για να αντιμετωπιστεί το πρόβλημα του variance είναι να αυξηθεί το batch size κατά την εκπαίδευση. Παρόλα αυτά η μεγάλη αύξηση του batch size έχει σαν αποτελέσμα να γίνει η δειγματοληψία αναποτελεσματική. Γι' αυτό έπρεπε να αναζητηθούν άλλες λύσεις για να αντιμετωπιστεί το ζήτημα του variance. [14]

4.3.2 Αλγόριθμος REINFORCE με πρότυπη τιμή

Λύση στο πρόβλημα του variance έρεχεται να λύση ο αλγόριθμος REINFORCE με πρότυπη τιμή. Στον αλγόριθμο REINFORCE η παράγωγος της συνάρτησης J έχει την παρακάτω μορφή:

$$\nabla_\theta J = (1/N) * \sum_{1}^{N}(\sum_{1}^{T}(\nabla_\theta log\pi_\theta(a_{i,t}|s_{i,t})) * \sum_{1}^{T}(r(s_{i,t}, a_{i,t}))$$

Ο όρος $\sum_1^T(r(s_{i,t}, a_{i,t}))$ αποτελεί τη γνωστή συνάρτηση $Q(s, a)$.

Μπορούμε πάντα να αφαιρέσουμε έναν όρο από τη συνάρτηση $Q(s, a)$ για να επιλύσουμε το πρόβλημα της βελτιστοποίησης. Ο όρος αυτός πρέπει μην εξαρτάται από τη μεταβλητή ϑ, με την οποία και παραγωγίζεται η συνάρτηση J. Γι' αυτό το λόγο αφαιρούμε τη συνάρτηση $V(s)$ από τη συνάρτηση $Q(s, a)$. Με αυτόν τον τρόπο λαμβάνουμε τη νέα εξίσωση:

$$\nabla_\theta J = (1/N) * \sum_{1}^{N}\sum_{1}^{T}(\nabla_\theta log\pi_\theta(a_{i,t}|s_{i,t})) * (Q(s_{i,t}, a_{i,t}) - V(s_{i,t})))$$

Σε αυτό το σημείο ορίζουμε για άλλη μία φορά στην παρούσα διπλωματική τη συνάρτηση πλεονεκτήματος A.

$$A_t = Q(s_t, a_t) - V(s_t))$$

Η ζητούμενη σχέση παίρνει τη μορφή:

$$\nabla_\theta J = (1/N) * \sum_{1}^{N} \sum_{1}^{T} \nabla_\theta log \pi_\theta(a_{i,t}|s_{i,t})) * A_{i,t}$$

Στη βαθιά μηχανική μάθηση θέλουμε οι είσοδοι να είναι εστιασμένες στο μηδέν, zero-centered. Από τη φύση της η ενισχυτική μάθηση ενδιαφέρεται για το αν μία δράση επιφέρει καλύτερη ανταμοιβή από τη μέση περίπτωση. Θα καταστούμε πιο κατανοητοί με ένα αντιπροσωπευτικό παράδειγμα. Ας δούμε αναλυτικά:

Περίπτωση 1: Η τροχιά A δέχεται συνολική ανταμοιβή 10, ενώ η τροχιά B δέχεται συνολική ανταμοιβή -10.

Περίπτωση 2:Η τροχιά A δέχεται συνολική ανταμοιβή 10, ενώ η τροχιά B δέχεται συνολική ανταμοιβή 1.

Με βάση τον αλγόριθμο στην πρώτη περίπτωση θα αυξηθεί η πιθανότητα να επιλέξει ο πράκτορας την τροχιά A, ενώ θα μειωθεί η πιθανότητα να επιλέξει ο πράκτορας την τροχιά B. Στη δεύτερη περίπτωση θα αυξηθεί η πιθανότητα να επιλέξει ο πράκτορας τόσο την τροχιά A όσο και την τροχιά B. Στην πραγματικότητα, όμως, θα θέλαμε στην περίπτωση 2 ο πράκτορας να επιλέξει την τροχιά A και όχι τη B, παρά το γεγονός ότι η τροχιά B επιφέρει θετική συνολική ανταμοιβή. Για κινηθεί ο αλγόριθμος προς αυτήν την κατεύθυνση εισάγεται η έννοια της πρότυπης τιμής. Στον αντίστοιχο ψευδοκώδικα της αντίστοιχης εικόνας 4.5 παραθέτουμε έναν basic ψευδοκώδικα του αλγορίθμου REINFORCE με πρότυπη τιμή.

Αιτιότητα

Οι επιλογές του παρόντος δεν θα πρέπει να επηρεάζουν τις επιλογές του παρελθόντος. Οι δράσεις που επιλέγει ο πράκτορας στην κάθε τρέχουσα στιγμή θα πρέπει να επηρεάζουν μόνο το μέλλον. Η αντικειμενική συνάρτηση, objective function, λαμβάνει την παρακάτω μορφή στον αλγόριθμο vanilla reinforce algorithm.

$$\nabla_\theta log \pi_\theta(a_{i,t}|s_{i,t})$$

$$\nabla_\theta J = (1/N) * \sum_{1}^{N} \sum_{1}^{T} (\nabla_\theta log \pi_\theta(a_{i,t}|s_{i,t})) * \sum_{t}^{T} (r(s_{i,t}, a_{i,t}))$$

Έκπτωτικός παράγοντας στη συνάρτηση συνολικής ανταμοιβής

Algorithm 1 "Vanilla" policy gradient algorithm

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

Εικόνα 4.5: Ψευδοκώδικας vanilla REINFORCE algorithm

Και σε αυτόν τον αλγόριθμο ενισχυτικής μάθησης οι ανταμοιβές πολλαπλασιάζονται με έναν εκπτωτικό παράγοντα. Ο εκπτωτικός παράγοντας επιχειρεί και αυτός με τη σειρά του να επιλύσει το πρόβλημα του variance. Ας δούμε για άλλη μία φορά, σε αυτήν τη διπλωματική, με ποιό τρόπο υπολογίζεται η συνάρτηση $Q(s, a)$ με τη χρήση του εκπτωτικού παράγοντα

$$Q(s, a) = (r_0 + \gamma * r_1 + \gamma^2 * r_1 + ...|s = s_0, a = a_0)$$

Με αυτόν τον τρόπο η νέα αντικειμενική συνάρτηση λαμβάνει τη μορφή

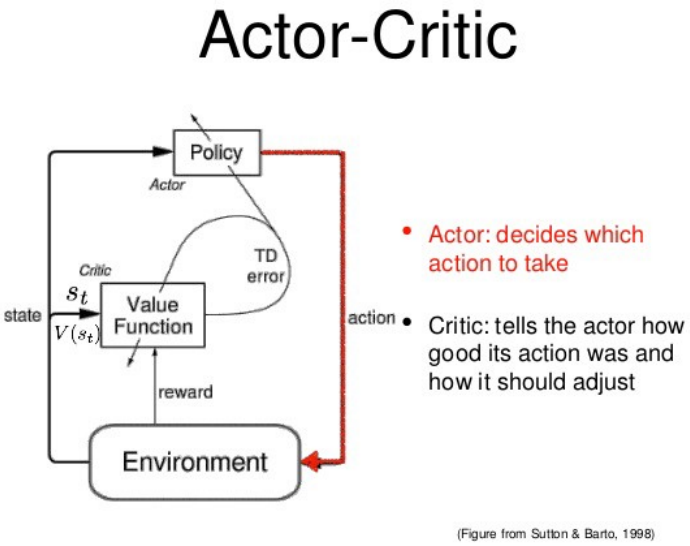
$$\nabla_\theta J = (1/N) * \sum_{1}^{N} \sum_{1}^{T} \nabla_\theta log \pi_\theta(a_{i,t}|s_{i,t}) * \sum_{t'=t}^{T} \gamma_{t-t'} * r(s_{i,t'}, a_{i,t'})$$

[14]

4.4 Μέθοδοι Δράστη - Κριτή

Αναλύσαμε στις προηγούμενες υποενότητες δύο από τις πιο βασικές κατηγορίες ενισχυτικής μάθησης. Τις μεθόδους Value-based και τις μεθόδους Policy based:

- Value-based: Οι Value-based μέθοδοι επιχειρούν να βρουν τη βέλτιστη συνάρτηση τιμών η οποία αντιστοιχίζει τις δράσεις με μία συνάρτηση τιμών. Όσο υψηλότερη είναι η τιμή τόσο καλύτερη θα είναι η δράση. Ο πιο δημοφιλής Value-based είναι ο αλγόριθμος της μάθησης-Q και οι παραλλαγές του αλγορίθμου DQN, Double DQN και Dueling DQN.



Εικόνα 4.6: Actor Critic Model

- Policy based: Οι Policy based αλγόριθμοι, όπως οι αλγόριθμοι Policy Gradient και REINFORCE επιχειρούν να βρουν τη βέλτιστη πολιτική απευθείας χωρίς να χρησιμοποιούν Q-Values.

Το επόμενο λογικό βήμα για την επιστημονική κοινότητα ήταν να επιχειρήσει να συνδυάσει τις δύο παραπάνω κατηγορίες αλγορίθμων, τις Value-based μεθόδους και τις Policy based. Με αυτόν τον τρόπο γενήθηκαν οι Actor-Critic αλγόριθμοι. Οι Actor-Critic αλγόριθμοι επιχειρούν να αναδείξουν και να συνδυάσουν τα πλεονεκτήματα των αλγορίθμων Value-based και Policy based, ενώ ταυτόχρονα επιχειρούν να εξαλείψουν τα αρνητικά των δύο αυτών κατηγοριών αλγορίθμων. Η κύρια ιδέα πίσω από τα συγκεκριμένα μοντέλα είναι ότι χωρίζεται ο κυρίως αγλόριθμος σε δύο μέρη. Στο πρώτο μέρος υπολογίζεται μία δράση βασιζόμενη στην τρέχουσα κατάσταση του πράκτορα, ενώ στη δεύτερη υπολογίζονται οι Q-Values που παράγει η δράση αυτή.[17]

Ας δούμε πιο αναλυτικά στο σημείο αυτό πώς υλοποιείται ένας Actor Critic αλγόριθμος.

Μπορούμε να γράψουμε την αντικειμενική συνάρτηση ως εξής:

$$\nabla_\theta J = E_{s_0,a_0,...s_t,a_t}[\sum_{t=0}^{T-1} \nabla_\theta log\pi_\theta(a_t|s_t)Q_w(a_t,s_t)]$$

Είναι γνωστό ότι η συνάρτηση Q μπορεί να παραμετροποιηθεί με τη χρήση

Algorithm 1 Q Actor Critic

Initialize parameters s, θ, w and learning rates α_θ, α_w; sample $a \sim \pi_\theta(a|s)$.
for $t = 1 \dots T$: **do**
  Sample reward $r_t \sim R(s, a)$ and next state $s' \sim P(s'|s, a)$
  Then sample the next action $a' \sim \pi_\theta(a'|s')$
  Update the policy parameters: $\theta \leftarrow \theta + \alpha_\theta Q_w(s, a) \nabla_\theta \log \pi_\theta(a|s)$; Compute the correction (TD error) for action-value at time t:
    $\delta_t = r_t + \gamma Q_w(s', a') - Q_w(s, a)$
  and use it to update the parameters of Q function:
    $w \leftarrow w + \alpha_w \delta_t \nabla_w Q_w(s, a)$
  Move to a $\leftarrow a'$ and s $\leftarrow s'$
end for

Εικόνα 4.7: Ψευδοκώδικας Q Actor Critic

κατάλληλου νευρωνικού δικτύου.

Τα παραπάνω μας οδηγούν στις Actor Critic μεθόδους.

1.**Κριτής**: Ο κριτής εκτιμάει τη συνάρτηση τιμής. Αυτή μπορεί να είναι η συνάρτηση τιμής $V(s)$ είτε η συνάρτηση τιμής δράσης $Q(s, a)$.

2.**Δράστης**: Ο δράστης ενημερώνει τις παραμέτρους της κατανομής πολιτικής κατά την κατεύθυνση που θα του προτείνει ο κριτής.

Τόσο η συνάρτηση δράστη όσο και η συνάρτηση κριτή παραμετροποιούνται με τη χρήση νευρωνικών δικτύων. Στον αλγόριθμο Actor-Critic που παραθέτουμε η συνάρτηση κριτή παραμετροποιεί την Q τιμή και γι' αυτό το λόγο ο συγκεκριμένος Actor-Critic αλγόριθμος που παρουσιάζουμε ονομάζεται Q Actor-Critic. Ο ψευδοκώδικας του αλγορίθμου φαίνεται στην εικόνα 4.7.

Για την υλοποίηση του αγλορίθμου A2C χρησιμοποιείται η συνάρτηση τιμής $V(s)$ σα συνάρτηση πρότυπης τιμής. Πραγματοποιείται η αφαίρεση $Q(s, a) - V(s)$. Διασθητικά με τη χρήση πρότυπης τιμής εξετάζουμε κατά πόσο η επιλογή μίας δράσης είναι καλύτερη σε σχέση με τη μέση δράση. Παίρνουμε με αυτόν τον τρόπο τη συνάρτηση πλεονεκτήματος $A(s, a)$.

$$A(s, a) = Q_w(s, a) - V_u(s)$$

Αυτό σημαίνει ότι πρέπει να κατασκευάσουμε δύο νευρωνικά δίκτυα ένα για τη συνάρτηση τιμής $V(s)$ και ένα για τη συνάρτηση δράσης τιμής $Q(s, a)$; Όχι η συγκεκριμένη επιλογή θα ήταν τελείως ασύμφορη. Ας θυμηθούμε με ποιον τρόπο συνδέεται συνάρτηση $V(s)$ με τη συνάρτηση $Q(s, a)$.

$$Q(s_t, a_t) = E[r_{t+1} + \gamma * V(s_{t+1})]$$

Μπορούμε να ξαναγράψουμε με αυτόν τον τρόπο τη συνάρτηση πλεονεκτήμας ως εξής:

$$A(s, a) = r_{t+1} + \gamma * V_u(s_{t+1}) - V_u(s_t)$$

Θα χρειαστούμε δηλαδή μόνο ένα νευρωνικό δίκτυο για τη συνάρτηση $V(s)$. Μπορούμε να ξαναγράψουμε τώρα την αντικειμενική συνάρτηση ως εξής:

$$\nabla_\theta J = \sum_{t=0}^{T-1} \nabla_\theta log\pi_\theta(a_t|s_t)(r_{t+1} + \gamma * V_u(s_{t+1}) - V_u(s_t)) \Rightarrow$$

$$\nabla_\theta J = \sum_{t=0}^{T-1} \nabla_\theta log\pi_\theta(a_t|s_t)A(s_t, a_t)$$

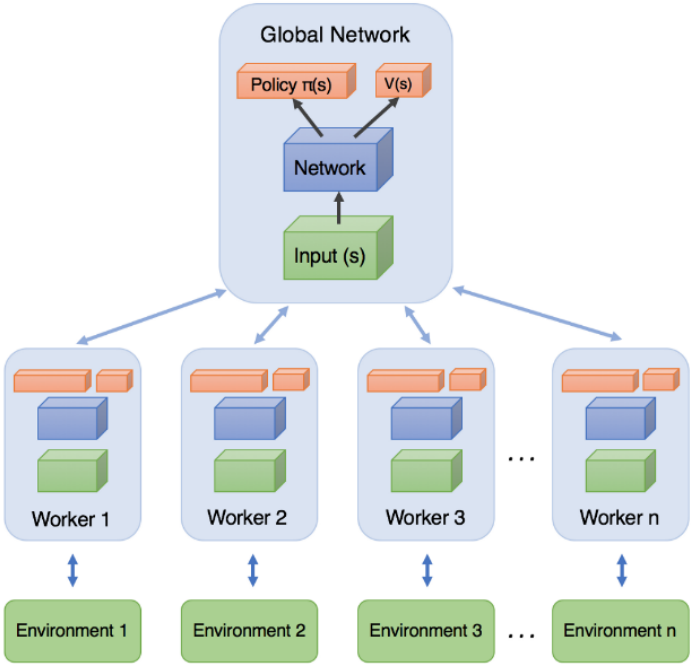
Αυτή η συνάρτηση χρησιμοποιείται στον αλγόριθμο A2C.[36]

Ο αλγόριθμος Asynchronous Advantage Actor-Critic (A3C) επινοήθηκε από την Deepmind στην έρευνα [19]. Στον αλγόριθμο αυτόν η κύρια ιδέα είναι ότι πολλαπλοί παράλληλοι πράκτορες "εργάζονται" σε διαφορετικά ανεξάρτητα περιβάλλοντα και ενημερώνουν όλοι μαζί ασύγχρονα τις global μεταβλητές του κυρίως περιβάλλοντος. Γι' αυτό και ονομάζεται ασύγχρονος ο συγκεκριμένος αλγόριθμος. Ένα από τα κύρια πλεονεκτήματα της ασυγχρονης λειτουργίας είναι ότι πραγματοποιείται καλύτερη και αποδοτικότερη εξερεύνηση του περιβάλλοντος εκμάθησης. Στην εικόνα με τίτλο 4.8 παραθέτουμε μία εικονικοποίηση της λειτουργίας του συγκεκριμένου αλγορίθμου.

Ο αλγόριθμος A2C είναι ακριβώς όπως ο αλγόριθμος A3C, απλά δεν περιλαμβάνει τις ασύγχρονες λειτουργίες, δηλαδή τους πολλαπλούς πράκτορες που λειτουργούν παράλληλα. Με απλά λόγια μπορούμε να πουμε ότι ο αλγόριθμος A2C αποτελεί μία single-worker εναλλακτική του αλγορίθμου A3C. Σε πολλές περιπτώσεις έχει βρεθεί ότι ο αλγόριθμος A2C έχει συγκρίσιμη απόδοση με τον αλγόριθμο A3C. [36]

4.5 Model-based μέθοδοι στη βαθιά ενισχυτική μάθηση

Στην παρούσα διπλωματική εργασία δεν πρόκειται να ασχοληθούμε με τις model-based μεθόδους ενισχυτικής μάθησης. Επειδή, όμως, αποτελούν μία σημαντική κατηγορία αλγορίθμων ενισχυτικής μάθησης θα αναφερθούμε συνοπτικά σε αυτές. Πιο συγκεκριμένα, στις model-based μεθόδους ο αλγόριθμος δημιουργεί ένα μοντέλο του πραγματικού περιβάλλοντος με βάση την εμπειρία που λαμβάνει από το πραγματικό περιβάλλον. Το νέο προσομοιωμένο περιβάλλον ονομάζεται μοντέλο. Με βάση το νέο προσομοιωμένο περιβάλλον, το οποίο αποτελεί μία μοντελοποίηση του πραγματικού περιβάλλοντος, ο αλγόριθμος σχεδιάζει τη συνάρτηση τιμής και την πολιτική που πρόκειται να ακολουθήσει ο πράκτορας.



Εικόνα 4.8: Asynchronous Advantage Actor-Critic (A3C)

Ας δούμε λιγο πιο αναλυτικά. Ο πράκτορας πρώτα λαμβάνει εμπειρία από το πραγματικό περιβάλλον. Στη συνέχεια, με βάση αυτήν την εμπειρία δημιουργείται ένα μοντέλο το οποίο θα προβλέπει τη συμπεριφορά του πραγματικού περιβάλλοντος. Στο επόμενο βήμα, ο αλγόριθμος χρησιμοποιεί το μοντέλο για να σχεδιάσει τη συμπεριφορά του πράκτορα. Αυτό μας επιτρέπει να σχεδιάσουμε τη συνάρτηση τιμής και την πολιτική που θα ακολουθήσει ο πράκτορας μόνο με βάση τη μοντελοποίηση του περιβάλλοντος που έχει πραγματοποιηθεί. Σε τελευταίο βήμα, ελέγχουμε την απόδοση του πράκτορα στο πραγματικό περιβάλλον.

Οι model-based μέθοδοι εμφανίζουν σίγουρα πολλά πλεονεκτήματα. Πιο βασικό είναι το γεγονός ότι η μοντελοποίηση του περιβάλλοντος μπορεί να μας δίνει καλύτερη εικόνα για το περιβάλλον σε σχέση με το πραγματικό περιβάλλον. Παρόλα αυτά εμφανίζουν και δύο βασικά μειονεκτήματα. Ας δούμε παρακάτω ποια είναι αυτά:

Μειονεκτήματα model-based μεθόδων

1. Η μοντελοποίηση του περιβάλλοντος αποτελεί μία προσέγγιση, η οποία πολύ συχνά εμπεριέχει ανακρίβειες.
2. Ο πράκτορας μπορεί να μάθει να κινείται τόσο καλά όσο του επιτρέπει η γνώση που δέχεται από το μοντελοποιημένο περιβάλλον.[10]

Κεφάλαιο 5

Ασύγχρονοι αλγόριθμοι ενισχυτικής μάθησης

5.1 Εισαγωγή

Στην παρούσα διπλωματική εργασία έχουμε στηριχθεί στην βιβλιογραφία [19]. Υλοποιούμε τους αλγορίθμους DQN, Double DQN, Dueling DQN, τον ασύγχρονο αλγόριθμο ενός βήματος μάθησης-Q (asynchronous one step Q-learning), τον ασύγχρονο αλγόριθμο ενός βήματος Sarsa (asynchronous one-step Sarsa), τον ασύγχρονο αλγόριθμο n-βημάτων μάθησης-Q (asynchronous n-step Q-learning) και τον ασύγχρονο αλγόριθμο δράστη-κριτή με συνάρτηση πλεονεκτήματος (asynchronous advantage actor-critic (A3C)) πάνω στο παιχνίδι Cart-pole για να εξετάσουμε κατά πόσο βελτιστοποιούν οι ασύγχρονοι αλγόριθμοι την απόδοση σε σχέση με τους κλασσικούς αλγορίθμους ενισχυτικής μάθησης. Μας ενδιαφέρει να κατανοήσουμε σε τι βαθμό βελτιώνουν την απόδοση του πράκτορα οι νέοι ασύγχρονοι αλγόριθμοι, καθώς και κατά πόσο χρόνο μας εξοικονομούν σε σχέση με του κλασσικούς μη κατανεμημένους αλγόριθμους.

Στα επόμενα υποκεφάλαια θα παρουσιάσουμε κάποια σημαντικά θεωρητικά τμήματα της έρευνας [19]. Θα συσχετίσουμε τη συγκεκριμένη θεωρία με το θεωρητικό υπόβαθρο που παρέχουμε στα προηγούμενα κεφάλαια. Παρουσιάζουμε, επίσης, τι έχουν παράξει και άλλοι ερευνητές πάνω στο ίδιο. Τέλος, στην παρούσα ενότητα θα περιγράψουμε και τι είδους πειράματα εκτελέσαμε για τη διεκπαιρέωση της παρούσας διπλωματικής εργασίας, καθώς και τις διαδικασίες που ακολουθήσαμε. Τα αποτελέσματα των πειραμάτων μας θα αναλυθούν στο επόμενο κεφάλαιο. Σε αυτήν την εργασία πραγματοποιούμε μία πολύ διαφορετική προσέγγιση της βαθιάς ενισχυτικής μάθησης. Αντί να χρησιμοποιείται η τεχνική replay experience, στους συγκεκριμένους αλγόριθμους

πολλοί πράκτορες λειτουργούν ασύγχρονα, σε πολλαπλά διαφορετικά στιγμιότυπα του περιβάλλοντος. Αυτού του είδους ο παραλληλισμός αποσυσχετίζει τα δεδομένα που επισκέπτονται οι πράκτορες, με αποτέλεσμα να έχουμε μία πιο στατική κατάσταση. Σε κάθε χρονικό βήμα, κάθε παράλληλος πράκτορας θα επισκέπτεται μία κατάσταση που δε θα σχετίζεται με τις καταστάσεις που θα επισκέπτονται οι άλλοι πράκτορες την ίδια χρονική στιγμή. Αυτή η απλή ιδέα μας παρέχει μία πληθώρα on-policy αλγορίθμων ενισχυτικής μάθησης. Τέτοιοι αλγόριθμοι είναι ο Sarsa, οι μέθοδοι n-step και οι μέθοδοι actor-critic. Επίσης, η συγκεκριμένη ιδέα εφαρμόζεται και σε off-policy αλγορίθμους, όπως ο αλγόριθμος της μάθησης-Q.

Η παραλληλοποίηση της ενισχυτικής μάθησης που παρουσιάζουμε προσφέρει πρακτικά οφέλη. Οι μέθοδοι άσυγχρονης ενισχυτικής μάθησης που παρουσιάζουμε δε χρησιμοποιούν hardware όπως οι GPU ή μαζικά κατανεμημένες εφαρμογές. Οι αλγόριθμοι ασύγχρονης ενισχυτικής μάθησης που θα χρησιμοποιήσουμε εφαρμόζονται στους κλασσικούς πολυπύρηνους επεξεργαστές CPU. Όταν εφαρμόστηκαν οι συγκεκριμένοι CPU-based αλγόριθμοι σε παιχνίδια της πλατφόρμας Atari 2006, σε πολλά παιχνίδια πέτυχαν καλύτερα αποτελέσματα από τους GPU-based αλγορίθμους. Την καλύτερη "συμπεριφορά" έμφάνισε ο αλγόριθμος A3C, ο οποίος κατάφερε να μάθει σε πολύ καλό βαθμό να χειρίζεται συνεχείς εργασίες μηχανής. Επίσης κατάφερε να μάθει να χειρίζεται 3D μάζες μόνο από τα οπτικά δεδομένα εισόδου. Θεωρείται ότι ο αλγόριθμος A3C μπορεί να μάθει επιτυχημένα να συμπεριφέρεται στο 2D και στο 3D χώρο, αλλά και στο συνεχή και στο διακριτό χώρο καταστάσεων. Επίσης μπορεί να εκπαιδεύσει διάφορα είδη πρακτόρων. Γι' αυτούς τους λόγους είναι ιδιαίτερα δημοφιλής και πετυχημένος, ενώ ταυτόχρονα θεωρείται και πολύ γενικός.

5.2 Σχετικές έρευνες

Η υλοποίηση του αλγορίθμου Gorila[22] αποτελεί μία από τις πιο σχετικές έρευνες στο τομέα της ασύχρονης βαθιάς ενισχυτικής μάθησης. Στο συγκεκριμένο αλγόριθμο κάθε διεργασία διατηρεί τον δικό της πράκτορα, ο οποίος δρα σε ένα αντίγραφο του πραγμτικού περιβάλλοντος που μας ενδιαφέρει. Κάθε πράκτορας διαθέτει ξεχωριστή ιδιωτική μνήμη, με μορφή replay memory και υπολογίζει την ποσότητα gradient descent και το DQN loss με βάση τις παραμέτρους της πολιτικής. Τα gradients στέλνονται ασύγχρονα σε ένα κεντρικό global server (διακομιστή), ο οποίος διατηρεί το global μοντέλο και παραμέτρους. Χρησιμοποιώντας 100 διαφορετικές διεργασίες και 30 παραμέτρους ο αλγόριθμος Gorila έχει καταφέρει να ξεπεράσει σε απόδοση τον κλασσικό αλγόριθμο DQN σε 49 παιχνίδια της πλατφόρμας Atari 2006. Σε πολλά παιχνίδια ο αλγόριθμος Gorila πέτυχε τα σκορ του DQN ακόμα και 20 φορές ταχύτερα. [19]

Σε άλλες προηγούμενες εργασίες εφαρμόστηκε ο αλγόριθμος Map Reduce πάνω στην παραλληλοποίηση batch μεθόδων ενισχυτικής μάθησης με χρήση προσεγγιστικών συναρτήσεων. Ο παραλληλισμός αυτός, όμως, χρησιμοποιήθηκε για την παραλληλοποίηση των πράξεων πινάκων και όχι για την παραλληλοποίηση της συγκέντρωσης δεδομένων ή για την εστάθεια της μάθησης. Σύμφωνα με τη βιβλιογραφία [13] παρουσιάζεται μία παραλλαγή του αλγορίθμου ενισχυτικής μάθησης Sarsa που χρησιμοποιεί πολλούς διαφορετικούς πράκτορες εκμάθησης. Κάθε πράκτορας μαθαίνει ατομικά και στέλνει τις ενημερώσεις για βάρη που έχουν αλλάξει σημαντικά στους άλλους πράκτορες με peer to peer επικοινωνία. [19]

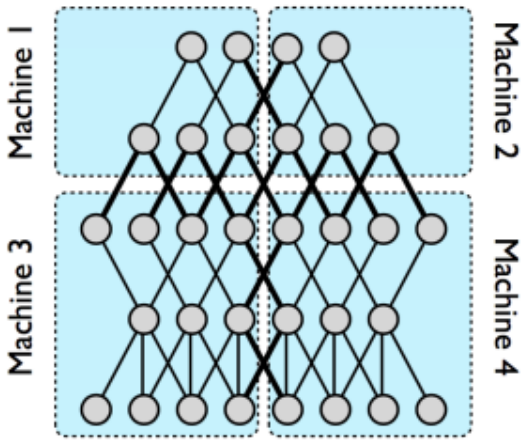
Ακόμα παλαιότερα, σύμφωνα με τη βιβλιογραφία [30] το έτος 1994 μελετήθηκαν οι ιδιότητες σύγκλισης του αλγορίθμου μάθησης-Q με βάση τους κανόνες ασύγχρονης βελτιστοποίησης. Τα αποτελέσματα της έρευνας έδειξαν ότι οι ασύγχρονοι αλγόριθμοι μάθησης-Q συγκλίνουν πάντα, όταν ικανοποιούνται κάποιες υποθέσεις και όταν εξασφαλιστεί ότι οι μη ενημερωμένες πληροφορίες κάποια στιγμή θα αποβληθούν. Ακόμα παλαιότερα, με βάση τη βιβλιογραφία [9] φαίνεται ότι ασχολήθηκε η επιστημονική κοινότητα με το συγκεκριμένο πρόβλημα από το έτος 1982. Τέλος, αξίζει να αναφέρουμε ότι και εξελικτικοί αλγόριθμοι έχει επιχειρηθεί να παραλληλοποιηθούν κατά το παρελθόν. Βέβαια, από τη φύση τους οι συγκεκριμένοι αλγόριθμοι είναι πιο εύκολο να παραλληλοποιηθούν. Τέτοιου είδους παραλληλοποιήσεις έχει επιχειρηθεί να εισαχθούν στην ενισχυτική μάθηση. [19]

5.3 Παραλληλοποίηση αλγορίθμων

Για να επιτευχθεί ταχύτερος υπολογισμός πολλών προβλημάτων μηχανικής μάθησης, αλλά και γενικότερα των προβλημάτων όλων των κατηγοριών πολλές φορές επιχειρείται η παραλληλοποίηση των αλγορίθμων για τη βελτίωση του χρόνου υπολογισμού, αλλά και την καλύτερη αξιοποίηση των παρεχόμενων υπολογιστικών πόρων. Δεν υπάρχει κάποια κοινώς αποδεκτή μεθοδολογία παραλληλοποίησης αλγορίθμων που να έχει επικρατήσει στην διεθνή επιστημονική κοινότητα. Όμως, υπάρχουν τρεις γενικότερες μέθοδοι παραλληλοποίησης αλγορίθμων και αλγορίθμων μηχανικής μάθησης ειδικότερα. Ας δούμε ποιές είναι αυτές:

1.Data-centric: Μοιράζονται τα δεδομένα σε μονάδες επεξεργασίας και στη συνέχεια οι υπολογισμοί. Κάθε μονάδα επεξεργασίας αναλαμβάνει να εκπαιδεύσει ένα υποσύνολο των παραμέτρων του μοντέλου με όλα τα δεδομένα εισόδου.

2.Task-centric: Μοιράζονται οι υπολογισμοί σε μονάδες επεξεργασίες και στη συνέχεια τα δεδομένα. Κάθε μονάδα επεξεργασίας διατηρεί ένα αντίγραφο



Εικόνα 5.1: data centric προσέγγιση[23]

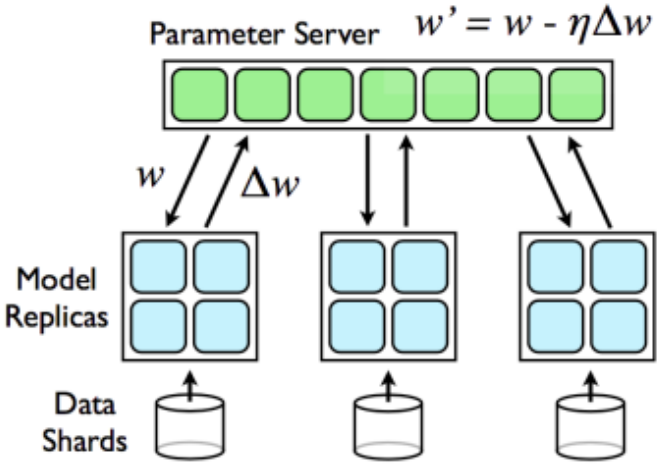
του μοντέλου και εκπαιδεύει τις παραμέτρους με ένα υποσύνολο των δεδομένων εισόδου.

3.Function-centric: Μοιράζονται διαφορετικές λειτουργίες/στάδια σε διαφορετικές μονάδες επεξεργασίας. Μία μονάδα επεξεργασίας αναλαμβάνει την προ-επεξεργασία (pre-processing) των δεδομένων και μια άλλη μονάδα επεξεργασίας αναλαμβάνει την εκπαίδευση του μοντέλου με τα δεδομένα που προκύπτουν από την προ-επεξεργασία.[23]

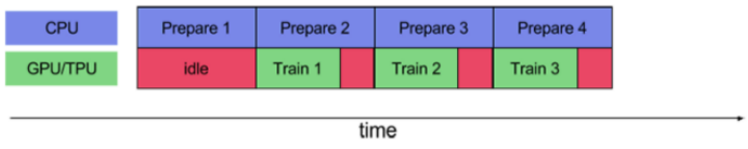
Στη δική μας εργασία έχουμε εστιάσει στους ασύγχρονους αλγορίθμους, οι οποίοι δεν ανήκουν αμιγώς σε κάποια από τις παραπάνω κατηγορίες. Όμως, αν θέλαμε να τους κατηγοριοποιήσουμε θα λέγαμε ότι ανήκουν στο ενδιάμεσο μεταξύ των κατηγοριών data-centric και task-centric. Εμφανίζουν οι αλγόριθμοι κοινά και με τις δύο κατηγορίες, αλλά και διαφορές. Προφανώς, θα μπορούσαν να γίνουν πολλές τροποποίησεις στους ασύγχρονους αλγορίθμους που χρησιμοποιούμε με στόχο τη βελτιστοποίησή τους, οι οποίες θα μπορούσαν να έχουν είτε data-centric είτε task-centric μορφή είτε και τις δύο. Θα είχε εξαιρετικό ενδιαφέρον η συγκεκριμένη προσεγγιση, καθώς θα εξετάζαμε σε τι θα οφελούσαν και κατά πόσο οι συγκεκριμένες μέθοδοι παραλληλοποιήσης στη μηχανική μάθηση και ακόμα πιο συγκεκριμένα στον ιδιαίτερα αναπτυσσόμενο πεδίο της βαθιάς μηχανικής μάθησης.

5.4 Ασύγχρονοι αλγόριθμοι ενισχυτικής μάθησης

Στην παρούσα διπλωματική εργασία ασχοληθήκαμε με αλγορίθμους ασύγχρονης ενισχυτικής μάθησης, σύμφωνα πάντα με την βιβλιογραφία [19]. Όλοι οι αλ-



Εικόνα 5.2: task centric προσέγγιση[23]



Εικόνα 5.3: function centric προσέγγιση [23]

γόριμοι που θα περιγράψουμε σε αυτό το κεφάλαιο είναι πολυνηματικοί (multi-threaded). Οι ασύγχρονοι αλγόριθμοι που θα περιγράψουμε παρακάτω είναι τροποποιήσεις των αλγορίθμων ασύγχρονος αλγόριθμος ενός βήματος μάθησης-Q, ασύγχρονος αλγόριθμος ενός βήματος Sarsa, ο ασύγχρονος αλγόριθμος n-βημάτων μάθησης-Q και A3C. Οι συγκεκριμένοι αλγόριθμοι ενισχυτικής μάθησης αναπτύχθηκαν με στόχο να εκπαιδευθούν νευρωνικά δίκτυα βαθιάς μηχανικής μάθησης, τα οποία θα είναι επιυχημένα ενώ ταυτόχρονα δε θα απαιτούν πολλούς υπολογιστικούς πόρους. Αξίζει να σημειώσουμε ότι οι ασύγχρονοι αλγόριθμοι ενισχυτικής μάθησης που χρησιμοποιούμε στην παρούσα διπλωματική είναι τελείως διαφορετικοί μεταξύ τους. Από την μία πλευρά ο αλγόριθμος A3C αποτελεί μία καθαρά on-policy μέθοδο ενισχυτικής μάθησης. Από την άλλη πλευρά ο αλγόριθμος μάθησης-Q θεωρείται ως μία καθαρά off-policy οικογένεια μεθόδων ενισχυτικής μάθησης. Παρόλες τις διαφορές όλοι οι ασύγχρονοι αλγόριθμοι που χρησιμοποιούμε εφμανίζουν δύο κοινά χαρακτηριστικά.

Πρώτον, στους ασύγχρονους αλγορίθμους που εστιάζουμε χρησιμοποιούνται παράλληλοι πράκτορες που λειτουργούν ασύγχρονα όπως ακριβώς με το framework Gorila. Με τη μόνη διαφορά ότι στο framework Gorila κάθε πράκτορας "τρέχει"σε διαφορετικό διακομιστή, ενώ στη δική μας περίπτωση κάθε πράκτορας (νήμα) "τρέχει" σε διαφορετικό πυρήνα του ίδιου επεξεργαστή. Με τη συγκεκριμένη καινοτομία μειώνεται το κόστος επικοινωνίας μεταξύ των νημάτων, καθώς δεν απαιτείται πλέον η επικοινωνία μεταξύ διαφορετικών διακομιστών.

Δεύτερον, από τη στιγμή που πολλοί πράκτορες λειτουργούν παράλληλα ακολουθώντας τροποποιημένη πολιτική ο καθένας είναι δεδομένο ότι με τους ασύγχρονους αλγόριθμους που περιγράφουμε θα υπάρχει καλύτερη και ταχύτερη εξερεύνηση του περιβάλλοντος δράσης. Το γεγονός ότι κάθε παράλληλος πράκτορας ακολουθεί ελαφρώς τροποποιημένη πολιτική, μας δίνει νέες προοπτικές. Οι παράλληλοι πράκτορες λαμβάνουν ασυσχέτιστες μεταξύ τους πληροφορίες από το περιβάλλον. Γι' αυτόν το λόγο οι ενημερώσεις στις global παραμέτρους πραγματοποιούνται χωρίς να υπάρχει το πρόβλημα της συσχέτισης των δεδομένων εισόδου που παρατηρείται στους κλασσικούς μη παράλληλους αλγορίθμους ενισχυτικής μάθησης. Στους ασύγχρονους αλγόριθμους δε χρησιμοποιείται η τεχνική replay memory που περιγράψαμε αναλυτικά στο προηγούμενο κεφάλαιο. Πέρα πάντως από την αυξημένη ευστάθεια που παρουσιάζουν οι ασύγχρονοι αλγόριθμοι ενισχυτικής μάθησης εμφανίζουν και άλλα πλεονεκτήματα. Οι συγκεκριμένοι αλγόριθμοι εκπαιδεύουν ταχύτερα τις global παραμέτρους του συστήματος σε σχέση με τους σειριακούς αλγόριθμους, για όλους τους λόγους που εξηγήσαμε στις προηγούμενες παραγράφους. Επιπρόσθετα από τη στιγμή που δε στηριζόμαστε στην τεχνική replay memory μπορούμε πλέον να χρησιμοποιήσουμε on-policy αλγορίθμους όπως ο A3C και ο Sarsa.

Παραθέτουμε αναλυτικά τους αλγόριθμους που χρησιμοποιήσαμε στην παρούσα

διπλωματική εργασία, βασισμένοι πάντα στην βιβλιογραφία [19]:

Ασύγχρονος αλγόριθμος ενός βήματος μάθησης-Q:

Κατ' αρχάς παρέχουμε ψευδοκώδικα του αλγορίθμου στην εικόνα 5.4. Με βάση τον ψευδοκώδικα κάθε νήμα αλληλεπιδρά με το δικό του προσωπικό αντίγραφο περιβάλλοντος και σε κάθε βήμα υπολογίζεται το gradient descent της συνάρτησης Q-loss. Επίσης, σε κάθε νήμα συσωρεύονται τα $d\theta$ του gradient descent της συνάρτησης Q-loss για προκαθορισμένο αριθμό επαναλήψεων και μετά πραγματοποιούνται οι ενημερώσεις των μεταβλητών ϑ και θ^- (βλέπε ψευδοκώδικα εικόνας 5.4). Δηλαδή, πραγματοποιείται μία αντίστοιχη προσέγγιση με το να χρησιμοποιούνταν minibatches. Η μέθοδος αυτή μειώνει την πιθανότητα πράκτορες διαφορετικών νημάτων να επικαλύπτουν τις ενημερώσεις που πραγματοποιούν στις παραμέτρους του global μοντέλου. Με το να συσσωρεύονται οι αλλαγές στις ενημερώσεις για τις μεταβλητές που επικεντρωνόμαστε, έχει ως αποτέλεσμα να λαμβάνουμε τη βέλτιστη ισορροπία από το δίπτυχο πλήθος μαθηματικών υπολογισμών σε σχέση με τον αριθμό των δεδομένων που χρησιμοποιούνται.

Σε τελική ανάλυση παρατηρείται ότι με το να μπορεί κάθε νήμα να ακολουθεί τη δική του ξεχωριστή πολιτική πραγματοποιείται καλύτερη εξερεύνηση του περιβάλλοντος. Υπάρχουν πολλοί τρόποι να διαφοροποιηθούν οι αλγόριθμοι που ακολουθεί το κάθε νήμα. Οι συγγραφείς της βιβλιογραφίας [19] έχουν επιλέξει κάθε νήμα να ακολουθεί ε-greedy πολιτική, με το κάθε νήμα να επιλέγει την επιμέρους ε-greedy πολιτική ακολουθώντας διαφορετική κατανομή για την επιλογή των ε κινήσεων. Με αυτόν τον τρόπο υπάρχει εγγύηση ότι κάθε νήμα θα ακολουθήσει αρκετά διαφορετικά μονοπάτια.

Ασύγχρονος αλγόριθμος ενός βήματος Sarsa:

Ο ασύγχρονος αλγόριθμος ενός βήματος Sarsa είναι ίδιος ακριβώς με τον ασύγχρονο αλγόριθμο ενός βήματος μάθησης-Q με τη μόνη διαφορά ότι ο ασύγχρονος αλγόριθμος ενός βήματος Sarsa δε χρησιμοποιεί την ίδια συνάρτηση Q. Πιο συγκεκριμένα δε χρησιμοποιεί τη συνάρτηση $Q(s, a)$, αλλά τη συνάρτηση $Q'(s, a) = r + \gamma * Q(s', a'; \theta^-)$, όπου με a' συμβολίζεται η δράση που ακολουθεί ο πράκτορας στην κατάσταση s'. Και σε αυτήν την περίπτωση χρησιμοποιείται target network, ενώ και σε αυτήν την περίπτωση οι ενημερώσεις πραγματοποιούνται μετά από συσσώρευση, όπως ακριβώς και στον ασύγχρονο αλγόριθμο ενός βήματος μάθησης-Q.

Ασύγχρονος αλγόριθμος n-βημάτων μάθησης-Q:

Κατ'αρχάς παρέχουμε τον ψευδοκώδικα του συγκεκριμένου αλγορίθμου στην αντίστοιχη εικόνα με τίτλο "Ψευδοκώδικας ασύγχρονου αλγορίθμου n-βημάτων μάθησης-Q", δηλαδή την εικόνα 5.5. Ο συγκεκριμένος αλγόριθμος είναι κάπως ασυνήθιστος. Επιχειρεί μία προς τα εμπρός προσέγγιση σε αντίθεση με άλλους αλγορίθμους που πραγματοποιούν προσέγγιση προς τα πίσω. Με στόχο να πραγματοποιηθεί η επόμενη ενημέρωση ο αλγόριθμος επιλέγει

Algorithm 1 Asynchronous one-step Q-learning - pseudocode for each actor-learner thread.

// Assume global shared θ, θ^-, and counter $T = 0$.
Initialize thread step counter $t \leftarrow 0$
Initialize target network weights $\theta^- \leftarrow \theta$
Initialize network gradients $d\theta \leftarrow 0$
Get initial state s
repeat
  Take action a with ϵ-greedy policy based on $Q(s, a; \theta)$
  Receive new state s' and reward r
  $$y = \begin{cases} r & \text{for terminal } s' \\ r + \gamma \max_{a'} Q(s', a'; \theta^-) & \text{for non-terminal } s' \end{cases}$$
  Accumulate gradients wrt θ: $d\theta \leftarrow d\theta + \frac{\partial (y - Q(s,a;\theta))^2}{\partial \theta}$
  $s = s'$
  $T \leftarrow T + 1$ and $t \leftarrow t + 1$
  if $T \mod I_{target} == 0$ **then**
    Update the target network $\theta^- \leftarrow \theta$
  end if
  if $t \mod I_{AsyncUpdate} == 0$ or s is terminal **then**
    Perform asynchronous update of θ using $d\theta$.
    Clear gradients $d\theta \leftarrow 0$.
  end if
until $T > T_{max}$

Εικόνα 5.4: Ψευδοκώδικας ασύγχρονου αλγορίθμου ενός βήματος μάθησης-Q[19]

Algorithm S1 Asynchronous n-step Q-learning - pseudocode for each actor-learner thread.

// Assume global shared parameter vector θ.
// Assume global shared target parameter vector θ^-.
// Assume global shared counter $T = 0$.
Initialize thread step counter $t \leftarrow 1$
Initialize target network parameters $\theta^- \leftarrow \theta$
Initialize thread-specific parameters $\theta' = \theta$
Initialize network gradients $d\theta \leftarrow 0$
repeat
  Clear gradients $d\theta \leftarrow 0$
  Synchronize thread-specific parameters $\theta' = \theta$
  $t_{start} = t$
  Get state s_t
  repeat
    Take action a_t according to the ϵ-greedy policy based on $Q(s_t, a; \theta')$
    Receive reward r_t and new state s_{t+1}
    $t \leftarrow t + 1$
    $T \leftarrow T + 1$
  until terminal s_t **or** $t - t_{start} == t_{max}$
  $R = \begin{cases} 0 & \text{for terminal } s_t \\ \max_a Q(s_t, a; \theta^-) & \text{for non-terminal } s_t \end{cases}$
  for $i \in \{t-1, \ldots, t_{start}\}$ **do**
    $R \leftarrow r_i + \gamma R$
    Accumulate gradients wrt θ': $d\theta \leftarrow d\theta + \frac{\partial (R - Q(s_i, a_i; \theta'))^2}{\partial \theta'}$
  end for
  Perform asynchronous update of θ using $d\theta$.
  if $T \mod I_{target} == 0$ **then**
    $\theta^- \leftarrow \theta$
  end if
until $T > T_{max}$

Εικόνα 5.5: Ψευδοκώδικας ασύγχρονου αλγορίθμου n-βημάτων μάθησης-Q[19]

τις επόμενες n δράσεις για μέχρι και t_{max} βήματα ή μέχρι να βρεθεί η τελική κατάσταση. Ο πράκτορας λαμβάνει μέχρι t_{max} ανταποδόσεις (rewards) από το περιβάλλον. Στη συνέχεια, ο αλγόριθμος υπολογίζει τα gradients για τα επόμενα n-step Q-learning updates, για κάθε ένα από τα καινούργια ζευγάρια κατάστασης-δράσης που βρήκε στο μονοπάτι ο πράκτορας. Κάθε n επόμενα βήματα ενημέρωσης χρησιμοποιεί τη μέγιστη n-step επιστροφή που μπορεί. Πιο συγκεκριμένα, χρησιμοποιεί one-step ενημέρωση για την τελευταία κατάσταση, two-step ενημέρωση για την προτελευταία και n-step ενημέρωση για τη νιοστή (t_{max}) από το τέλος κατάσταση. Οι συσσωρευμένες ενημερώσεις πραγματοποιούνται σε ένα μόνο βήμα gradient.

Ασύγχρονος αλγόριθμος δράστη-κριτή με συνάρτηση πλεονεκτήματος (Asynchronous advantage actor-critic (A3C)):

Στο συγκεκριμένο αλγόριθμο, αλλά και στη συγκεκριμένη οικογένεια αλγορίθμων αναφερθήκαμε στο προγηγούμενο κεφάλαιο αναλυτικά. Σε αυτό το σημείο θα περιγράψουμε πιο αναλυτικά το συγκεκριμένο αλγόριθμο. Σύμφωνα με τη βιβλιογραφία [19] παρουσιάζεται μία παραλλαγή του ασύγχρονου αλγορίθμου A3C. Ο εναλλακτικός αλγόριθμος της έρευνας [19] διατηρεί μία συνάρτηση πολιτικής $\pi(a_t|s_t;\theta)$ και μία συνάρτηση τιμής πολιτικής $V(s_t;\theta)$. Όπως ο εναλλακτικός ασύγχρονος αλγόριθμος n βημάτων μάθησης-Q παρατηρεί τα n επόμενα βήματα, με τον ίδιο τρόπο και ο εναλλακτικός A3C της ίδιας έρευνας παρατηρεί τα n επόμενα βήματα για να ενημερώσει τόσο τη συνάρτηση πολιτικής όσο και τη συνάρτηση τιμής του αλγορίθμου. Η συνάρτηση τιμής και

η συνάρτηση πολιτικής του εναλλακτικού A3C ενημερώνονται κάθε t_{max} βήματα ή νωρίτερα από t_{max} βήματα αν ο πράκτορας φτάσει στην τελική του κατάσταση. Η παράγωγος της συνάρτησης απωλειών δίνεται από τη σχέση $\nabla_{\theta'} log\pi_{\theta'}(a_t|s_t)A(s_t, a_t; \theta, \theta_v)$, όπου A είναι η συνάρτηση πλεονεκτήματος που έχουμε περιγράψει αναλυτικά σε προηγούμενα κεφάλαια. Στο συγκεκριμένο, όμως, αλγόριθμο η συνάρτηση A λαμβάνει τη μορφή $A = \sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^\kappa V(s_{t+k}; \theta_v) - V(s_t; \theta_v)$. Η μεταβλητή κ μπορεί να μεταβάλλεται από κατάσταση σε κατάσταση αλλά δεν μπορεί να ξεπερνάει σε τιμή τη μεταβλητή t_{max}. Ο ψευδοκώδικας του συγκεκριμένου εναλλακτικού A3C αλγορίθμου δίνεται από την εικόνα με τίτλο "Ψευδοκώδικας αλγορίθμου A3C", δηλαδή την εικόνα 5.6.

Όπως ακριβώς και με τις ασύγχρονες value-based μεθόδους, έτσι στις μεθόδους δράστη-κριτή οι παράλληλοι πράκτορες και οι συσσωρευμένες αλλαγές που πραγματοποιούν στις global μεταβλητές του κεντρικού περιβάλλοντος δίνουν σταθερότητα στο σύστημα. Οι συγγραφείς της έρευνας [19] έχουν πραγματοποιήσει και μία ακόμα τροποποίηση. Η συνάρτηση πολιτικής π έχει τις μεταβλητές ϑ και η συνάρτηση τιμής έχει τιμές θ_v. Οι μεταβλητές κανονικά ϑ και θ_v είναι διαφορετικές. Όμως, στο συγκεκριμένο αλγόριθμο κάποιες από τις μεταβλητές ϑ και θ_v είναι κοινές. Οι συγγραφείς της έρευνας [19] υλοποίησαν ένα συνελικτικό νευρωνικό δίκτυο για τη συνάρτηση πολιτικής π με συνάρτηση softmax για έξοδο, ενώ για τη συνάρτηση τιμής $V(s_t; \theta_v)$ έχουν χρησιμοποιήσει μία μόνο linear έξοδο. Η καινοτομία είναι ότι σε όλα τα μη τελικά επίπεδα του νευρωνικού δικτύου περιέχονται κοινές μεατβλητές και στα δύο νευρωνικά δίκτυα. Δηλαδή, οι μεταβλητές ϑ και θ_v ταυτίζονται σε όλα τα επίπεδα του νευρωνικού δικτύου εκτός από το τελικό επίπεδο. Είναι προφανές ότι με τη συγκεκριμένη καινοτομία εξοικονομούνται πολλοί υπολογιστικοί πόροι κατά την εκπαίδευση.

Επιπρόσθετα στην έρευνα [19] έχει προστεθεί μία ακόμα βελτιστοποίηση. Πιο συγκεκριμένα, οι ερευνητές της πρόσθεσαν την εντροπία πολιτικής στην αντικειμενική συνάρτηση, ώστε να εξαλειφθεί το φαινόμενο της πρόωρης σύγκλισης σε μη ολικά βέλτιστα σημεία. Πρώτοι τη συγκεκριμένη καινοτομία εισήγαγαν οι επιστήμονες που συνέγραψαν την έρευνα [34]. Θεώρησαν ότι η συγκεκριμένη καινοτομία είναι ιδιαίτερα αποτελεσματική σε εργασίες με ιεραρχική συμπεριφορά. Ας δούμε σε αυτό το σημείο τη βελτιστοποιημένη εξίσωση που μας ενδιαφέρει. Το gradient της αντικειμενικής συνάρτησης λαμβάνει την ακόλουθη μορφή:

$$\nabla_{\theta'} log\pi_{\theta'}(a_t|s_t)(R_t - V(s_t; \theta_v)) + \beta * \nabla_{\theta'} H(\pi(s_t); \theta')$$

όπου η συνάρτηση H είναι η εντροπία. Η υπερπαράμετρος β ρυθμίζει σε τι βαθμό η τιμή της εντροπίας επηρεάζει την τελική τιμής του gradient της αντικειμενικής συνάρτησης.

Algorithm S2 Asynchronous advantage actor-critic - pseudocode for each actor-learner thread.

// Assume global shared parameter vectors θ and θ_v and global shared counter $T = 0$
// Assume thread-specific parameter vectors θ' and θ'_v
Initialize thread step counter $t \leftarrow 1$
repeat
  Reset gradients: $d\theta \leftarrow 0$ and $d\theta_v \leftarrow 0$.
  Synchronize thread-specific parameters $\theta' = \theta$ and $\theta'_v = \theta_v$
  $t_{start} = t$
  Get state s_t
  repeat
    Perform a_t according to policy $\pi(a_t|s_t;\theta')$
    Receive reward r_t and new state s_{t+1}
    $t \leftarrow t + 1$
    $T \leftarrow T + 1$
  until terminal s_t **or** $t - t_{start} == t_{max}$
  $R = \begin{cases} 0 & \text{for terminal } s_t \\ V(s_t, \theta'_v) & \text{for non-terminal } s_t \text{// Bootstrap from last state} \end{cases}$
  for $i \in \{t-1, \ldots, t_{start}\}$ **do**
    $R \leftarrow r_i + \gamma R$
    Accumulate gradients wrt θ': $d\theta \leftarrow d\theta + \nabla_{\theta'} \log \pi(a_i|s_i;\theta')(R - V(s_i;\theta'_v))$
    Accumulate gradients wrt θ'_v: $d\theta_v \leftarrow d\theta_v + \partial\,(R - V(s_i;\theta'_v))^2/\partial\theta'_v$
  end for
  Perform asynchronous update of θ using $d\theta$ and of θ_v using $d\theta_v$.
until $T > T_{max}$

Εικόνα 5.6: Ψευδοκώδικας αλγορίθμου A3C[19]

5.5 Περιγραφή πειραμάτων

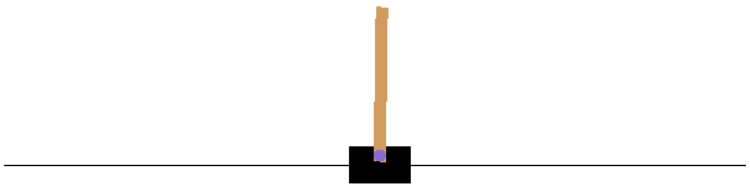
Στη δική μας διπλωματική εργασία έχουμε χρησιμοποιήσει τους αλγορίθμους DQN, Double DQN, Dueling DQN και τις ασύγχρονες παραλλαγές των αλγορίθμων ασύγχρονος αλγόριθμος ενός βήματος μάθησης-Q, ασύγχρονος αλγόριθμος ενός βήματος Sarsa, ασύγχρονος αλγόριθμος n-βημάτων μάθησης-Q και A3C , με βάση το [19], ώστε να συγκρίνουμε την απόδοσή τους πάνω στο σχετικά απλό παιχνίδι Cart-pole. "Τρέξαμε" τους ασύγχρονους multithreaded αλγορίθμους και εμείς σε CPU, ενώ τους αλγορίθμους DQN, Double DQN και Dueling DQN σε GPU. Εκτελέσαμε τους ασύγχρονους αλγορίθμους, μεταβάλλοντας χωριστά τον αριθμό των νημάτων και τον αριθμό των υπερπαρμέτρων τους ώστε να παρατηρήσουμε και εμείς πειραματικά τη "συμπεριφορά" τους πάνω παιχνίδι Cart-pole .

Κεφάλαιο 6

Πειράματα

6.1 Εισαγωγή

Στο παρόν κεφάλαιο ολοκληρώνουμε την ανάλυση των ασύγχρονων αλγορίθμων ενισχυτικής μάθησης. Υλοποιήσαμε τους 4 αλγορίθμους ασύγχρονης ενισχυτικής μάθησης που περιγράψαμε στην προηγούμενη ενότητα. Αναλυτικότερα, υλοποιήσαμε τους ασύγχρονους αλγορίθμους ενισχυτικής μάθησης ασύγχρονος αλγόριθμος ενός βήματος μάθησης-Q, ασύγχρονος αλγόριθμος ενός βήματος Sarsa, ασύγχρονος αλγόριθμος n-βημάτων μάθησης-Q και A3C. Τους συγκεκριμένους αλγορίθμους τους περιγράψαμε αναλυτικά στην προηγούμενη ενότητα. Για να συγκρίνουμε την απόδοση των ασύγχρονων αλγορίθμων επιλέξαμε να υλοποιήσουμε τους αλγορίθους βαθιάς ενισχυτικής μάθησης DQN, Double DQN και Dueling DQN, οι οποίοι εμφανίζουν state-of-the-art επιδόσεις σε πολλές εφαρμογές της ενισχυτικής μάθησης. Έχουμε αναφερθεί εκτεταμένα στους συγκεκριμένους αλγορίθμους στο αντίστοιχο κεφάλαιο. Όλα τα παραπάνω προγράμματα τα εφαρμόσαμε στο σχετικά απλό παιχνίδι Cart-pole. Σημειώνουμε ότι τους ασύγχρονους αλγορίθμους ενισχυτικής μάθησης τους εκτελέσαμε σε CPU, ενώ τους αλγορίθμους της κατηγορίας DQN τους εκτελέσαμε σε μία GPU Nvidia 1080. Επίσης, πρέπει να αναφέρουμε ότι για την υλοποίηση όλων των προγραμμάτων της παρούσας διπλωματικής εργασίας χρησιμοποιήσαμε το σχετικά νέο framework pytorch, το οποίο είναι κατάλληλο για την κατασκευή έργων βαθιάς μηχανικής μάθησης. Τέλος, οφείλουμε να αναφέρουμε ότι για την εκτέλεση του παιχνιδιού Cart-pole χρησιμοποιήσαμε τη βιβλιοθήκη gym της γλώσσας python.



Εικόνα 6.1: Στιγμιότυπο από το παιχνίδι Cart-pole[25]

6.2 Παιχνίδι Cart-pole

Το παιχνίδι Cart-pole είναι γνωστό και ως το παιχνίδι ανάποδο εκρεμμές. Σε αυτό το παιχνίδι ο πράκτορας προσπαθεί να ισορροπήσει το εκκρεμές για όσο περισσότερο χρονικό διάστημα μπορεί. Υποτίθεται ότι στην άκρη του στύλου, υπάρχει ένα αντικείμενο που τον καθιστά ασταθή και πολύ πιθανό να πέσει. Ο στόχος αυτής της εργασίας είναι να μετακινεί ο πράκτορας την κάρτα αριστερά και δεξιά, έτσι ώστε το κοντάρι να μπορεί να σταθεί (εντός μιας συγκεκριμένης γωνίας) όσο το δυνατόν περισσότερο. [24]

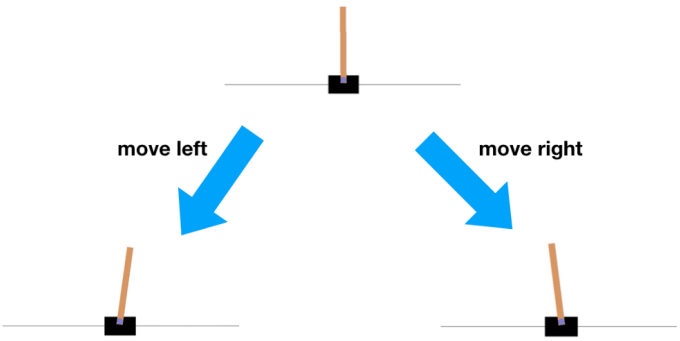
Κατά την εκπαίδευσή του ο πράκτορας με τους αλγορίθμους ενισχυτικής μάθησης δεν ξέρει πώς θα κινείται. Μετά, όμως, από σχετικά σύντομο χρονικό διάστημα και πάντα σταδιακά ο πράκτορας μαθαίνει να λύνει το συγκεκριμένο πρόβλημα κινούμενος πότε δεξιά και πότε αριστερά ανάλογα με το τι χρειάζεται να κάνει. [24]

Ανταμοιβές παιχνιδιού Cart-pole

Ας προσπαθήσουμε να καταλάβουμε αυτό το παιχνίδι λίγο περισσότερο. Και πάλι, ο στόχος είναι να παραμείνει "ζωντανός" ο πράκτορας όσο το δυνατόν περισσότερο. Όσο περισσότερο κρατάει ο πράκτορας το κοντάρι όρθιο, τόσο περισσότερο σκορ θα λαμβάνει. Για κάθε χρονικό βήμα που ο πράκτορας παραμένει "ζωντανός" στο παιχνίδι λαμβάνει ανταμοιβή +1, αλλιώς λαμβάνει ανταμοιβή 0 και το παιχνίδι τελειώνει σε εκείνο το χρονικό βήμα. Η βαθμολογία αυτή, που ονομάζεται επίσης επιβράβευση, είναι η ανταμοιβή που δίνεται στον πράκτορα για να γνωρίζει αν η δράση του είναι καλή ή όχι. Με βάση αυτή τη λογική, ο πράκτορας θα προσπαθήσει να βελτιστοποιήσει και να επιλέξει τη σωστή ενέργεια. Αξίζει να σημειώσουμε ότι, το παιχνίδι τελειώνει όταν το κοντάρι υπερβεί τη γωνία των 12 μοιρών ή το καλάθι βγαίνει από την οθόνη.[24]

Καταστάσεις παιχνιδιού Cart-pole

Η τρέχουσα κατάσταση του στύλου που βρίσκεται πάνω στην κάρτα του

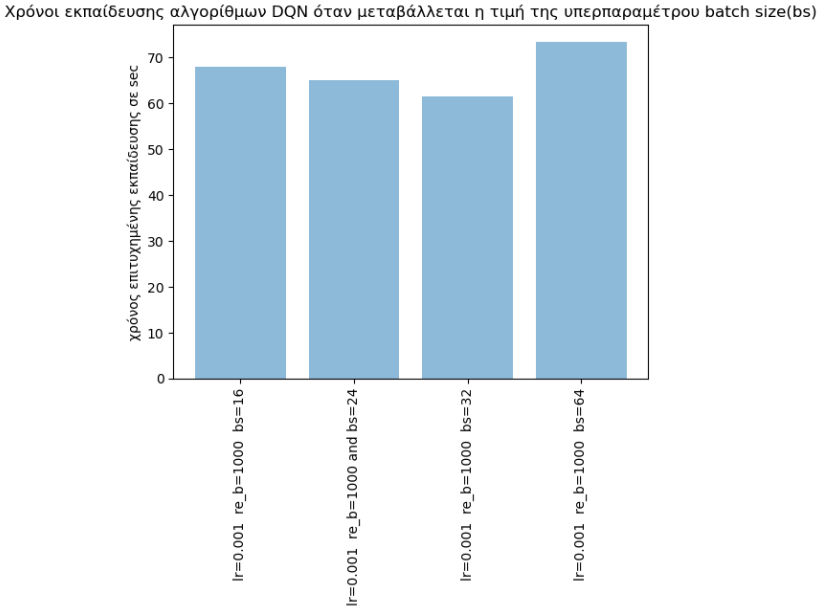


Εικόνα 6.2: Στιγμιότυπο από την κίνηση του κονταριού στο παιχνίδι Cart-pole[24]

παιχνιδιού (αν ανατρέπει προς τα αριστερά ή προς τα δεξιά) είναι γνωστή ως κατάσταση. Μια κατάσταση μπορεί να είναι το τρέχον πλαίσιο (σε pixel) που δέχεται ο πράκτορας. Μία κατάσταση μπορεί, επίσης, να είναι κάποια άλλη πληροφορία που μπορεί να αντιπροσωπεύει τη σχέση του στύλου με το καρότσι, για παράδειγμα, την ταχύτητα και τη θέση του καροτσιού, τη γωνία του πόλου και την ταχύτητα του πόλου στην άκρη. Στην περίπτωσή μας ο πράκτορας δε λαμβάνει σε πίξελς την πληροφορία για την κατάσταση του συστήματος. Λαμβάνει, όμως, ένα διάνυσμα 4 στοιχείων που του παρέχει όλη την απαραίτητη πληροφορία για να κατανοήσει την κατάσταση του συστήματος.

Με βάση τη δράση που επιλέγει ο πράκτορας πρόκειται να οδηγηθεί την επόμενη χρονική στιγμή σε μία νέα κατάσταση. Ας υποθέσουμε ότι το κοντάρι ξεκινά ευθεία, αν πάμε αριστερά την κάρτα, το κοντάρι ξεκινάει να πηγαίνει προς τα δεξιά, που είναι μια νέα κατάσταση. Επομένως, κατά τη διάρκεια κάθε χρονικού βήματος, οποιαδήποτε ενέργεια κάνουμε θα οδηγεί πάντα σε διαφορετική κατάσταση, βλέπε εικόνα 6.2.[24]

Εμείς στην παρούσα διπλωματική, όπως έχουμε αναφέρει, επιχειρούμε να εκπαιδεύσουμε τους πράκτορες ενισχυτικής χρησιμοποιώντας τους τέσσερις ασύγχρονους αλγορίθμους της έρευνας [19] και τους ιδιαίτερα δημοφιλής αλγορίθμους βαθιάς ενισχυτικής μάθησης DQN, Double DQN και Dueling DQN. Χρησιμοποιήσαμε το συγκεκριμένο σχετικά απλό παιχνίδι Cart-pole γιατί είναι ταχύτερη η εκπαίδευση των πρακτόρων μας σε σχέση με άλλα παιχνίδια της ίδιας κατηγορίας. Με αυτόν τον τρόπο, όμως, δεν εκμεταλλευόμαστε πλήρως τις "εντυπωσιακές" δυνατότητες των αλγορίθμων τύπου DQN. Οι συγκεκριμένοι αλγόριθμοι είναι κατασκευασμένοι ώστε να δέχονται σαν είσοδο εικόνες (πίνακες από πίξελς), χωρίς να πραγματοποιείται καμία απολύτως προεργασία. Στη δική μας περίπτωση, την περίπτωση του απλού προβλήματος Cart-pole τα νευρωνικά δίκτυα τύπου DQN δέχονται σαν είσοδο κατευθείαν ένα διάνυσμα



Εικόνα 6.3: Χρόνοι εκπαίδευσης αλγορίθμων DQN όταν μεταβάλλεται η τιμή της υπερπαραμέτρου batch size(bs)

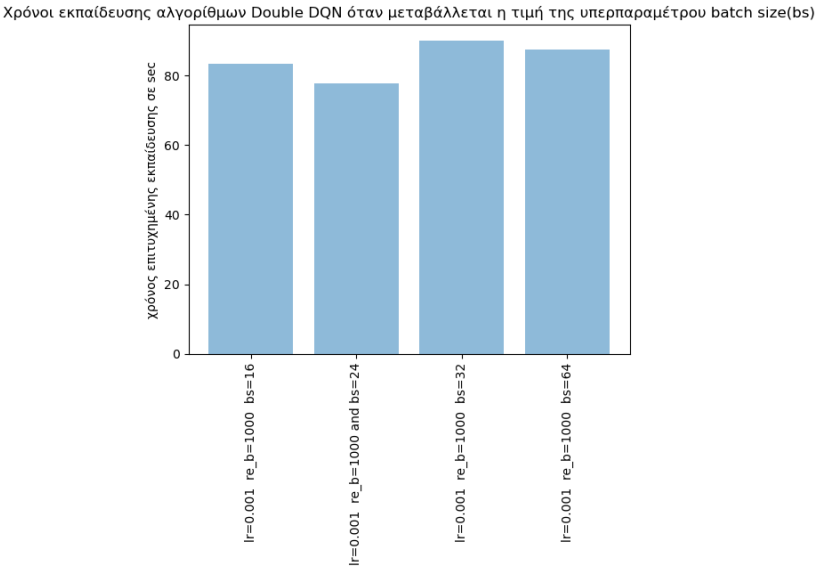
τεσσάρων στοιχείων, το οποίο περιγράφει πλήρως την κατάσταση του περιβάλλοντος και όχι εικόνα από πίξελς. Γι' αυτόν κυρίως το λόγο δεν εξαντλούμε τις δυνατότητες των αλγορίθμων τύπου DQN στην παρούσα διπλωματική εργασία.

6.3 Ανάλυση αποτελεσμάτων

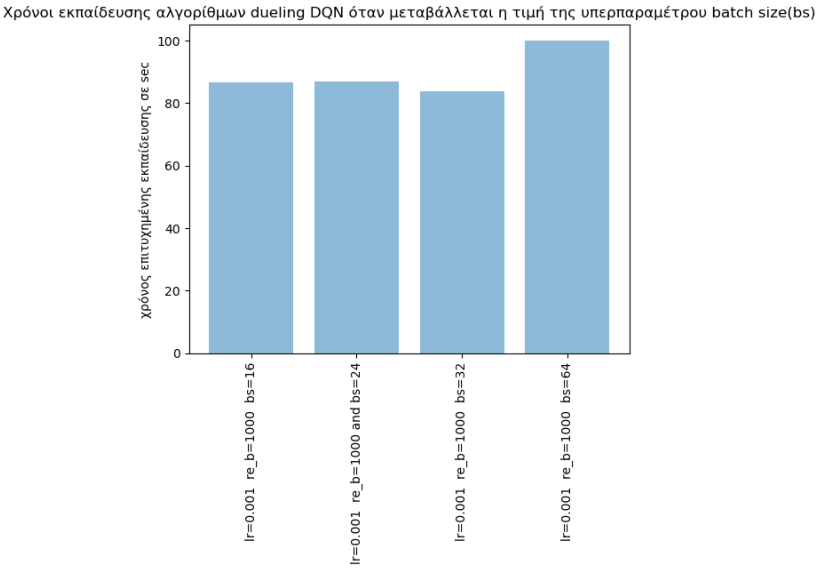
Σε πρώτη φάση, εκτελέσαμε τους αλγορίθμους DQN, Double DQN και Dueling DQN, τροποποιώντας τον κώδικα από το αποθετήριο github [15]. Όταν οι αλγόριθμοι έφταναν να έχουν σκορ 300 η εκπαίδευσή τους σταματούσε. Εκτελέσαμε τους συγκεκριμένους αλγορίθμους βαθιάς ενισχυτικής μάθησης για διάφορους συνδυασμούς των υπερπαραμέτρων. Πιο συγκεκριμένα εκτελέσαμε τους συγκεκριμένους αλγόριθμους για τις τιμές της υπερπαραμέτρου batch size 16, 24, 32, 64. Στη συνέχεια εκτελέσαμε τους αλγορίθμους για τιμές learning rate 0.001 και 0.005. Επιπρσθέτως, χρησιμοποιήσαμε replay buffer, του οποίου δώσαμε ξεχωριστά τις τιμές 250, 500, 1000.

Πρώτα θα δούμε τους DQN, Double DQN και Dueling DQN πώς αποδίδουν όταν η τιμή της υπερπαρμέτρου learning rate είναι 0.001 και η τιμή υπερπαραμέτρου replay buffer είναι 1000. Τα αποτελέσματα των μετρήσεων παρουσιάζονται στις εικόνες 6.3, 6.4 και 6.5.

Με βάση τα πειραματικά αποτελέσματα παρατηρούμε ότι και στις τρεις περιπ-



Εικόνα 6.4: Χρόνοι εκπαίδευσης αλγορίθμων Double DQN όταν μεταβάλλεται η τιμή της υπερπαραμέτρου batch size(bs)

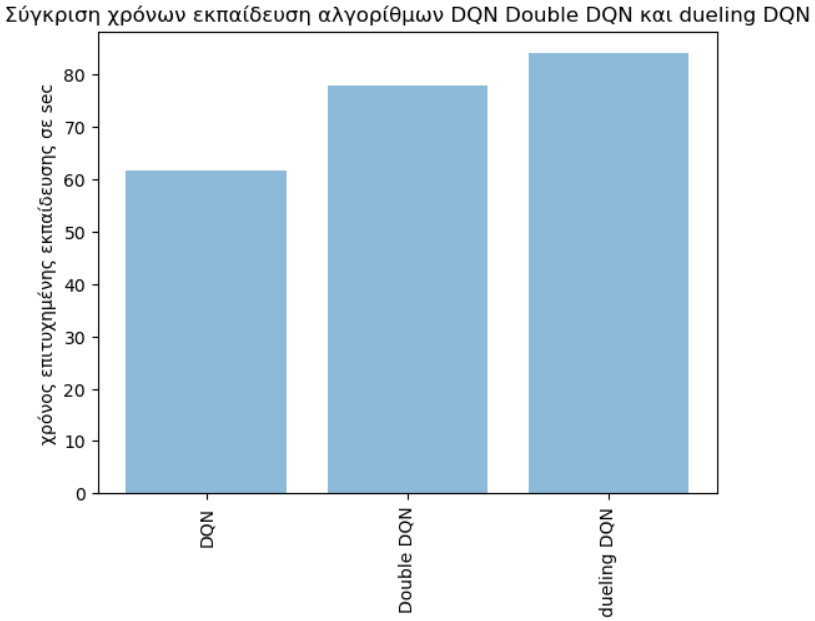


Εικόνα 6.5: Χρόνοι εκπαίδευσης αλγορίθμων Dueling DQN όταν μεταβάλλεται η τιμή της υπερπαραμέτρου batch size(bs)

τώσεις των DQN, Double DQN και Dueling DQN οι αλγόριθμοι εκπαιδεύονται ταχύτερα όταν η υπερπαράμετρος batch size παίρνει τιμές μεταξύ 24 και 32. Τόσο για μεγαλύτερες τιμές της υπερπαραμέτρου batch size όσο και για μικρότερες από αυτό το διάστημα οι αλγόριθμοι φαίνεται να εκπαιδεύονται πιο αργά. Στη συνέχεια, διατηρώντας τη βέλτιστη τιμή της υπερπαραμέτρου batch size, αλλάξαμε την τιμή της υπερπαραμέτρου learning rate από 0.001 σε 0.005. Με αυτήν την αλλαγή και οι τρεις αλγόριθμοι DQN, Double DQN και Dueling DQN εκπαιδεύθηκαν σε κατα πολύ αργότερους χρόνους σε σχέση με προηγούμενη περίπτωση. Όταν δηλαδή η τιμή της υπερπαραμέτρου learning rate ήταν 0.001. Γι' αυτόν το λόγο διατηρήσαμε την τιμή του learning rate σε 0.001. Επιπρόσθετα, σε όλα τα πειράματα χρησιμοποιήσαμε replay buffer, του οποίου δώσαμε ξεχωριστά τις τιμές 250, 500, 1000. Σε όλες τις περιπτώσεις και οι τρεις αλγόριθμοι απέδιδαν ταχύτερα όταν ο replay buffer επαιρνε την τιμή 1000. Γι' αυτόν τον λόγο όταν μεταβάλαμε την τιμή του batch size, το μέγεθος του replay buffer ήταν πάντα 1000. Τέλος, συγκρίναμε την επίδοση των τριών αλγορίθμων DQN, Double DQN και Dueling DQN, βλέπε εικόνα 6.6. Με βάση τα πειραματικά αποτελέσματα ταχύτερα εκπαιδεύτηκε ο αλγόριθμος DQN και όχι οι άλλες δύο βελτιστοποιήσεις του, οι αλγόριθμοι Double DQN και Dueling DQN. Τη συγκεκριμένη συμπεριφορά δεν την αναμέναμε. Βέβαια το παιχνίδι Cart-pole επειδή θεωρείται εξαιρετικά απλό δεν μπορεί να χρησιμοποιηθεί για την εξαγωγή ασφαλών συμπερασμάτων πάνω σε αλγορίθμους μηχανικής μάθησης.

Σε δεύτερη φάση, υλοποιήσαμε τους αλγορίθμους της έρευνας [19] για διάφορες τιμές των υπερπαραμέτρων και εκτελώντας τους χωριστά για αριθμό νημάτων 1, 2, 4, 8, 16, ώστε να παρατηρήσουμε πώς κλιμακώνει η απόδοσή τους με την αύξηση του αριθμού των νημάτων. Στη συνέχεια, μεταβάλαμε την τιμή της παραμέτρου learning rate, ώστε να εξετάσουμε με ποιο τρόπο αλλάζει ο χρόνος εκπαίδευσης του προγράμματος A3C. Πιο αναλυτικά, υλοποιήσαμε και "τρέξαμε" τις υλοποιήσεις των ασύγχρονων αλγορίθμων ασύγχρονος αλγόριθμος ενός βήματος μάθησης-Q, ασύγχρονος αλγόριθμος ενός βήματος Sarsa, ασύγχρονος αλγόριθμος n-βημάτων μάθησης-Q και αλγόριθμος A3C.

Αρχικά, εκτελέσαμε το πρόγραμμα A3C, δηλαδή τον πιο υποσχόμενο αλγόριθμο της έρευνας [19]. Η επίδοση του αλγορίθμου δεν κλιμάκωσε όπως θα θέλαμε. Δηλαδή, ο αλγόριθμος A3C δεν εκπαιδεύονταν υπεργραμμικά ταχύτερα σε σχέση με την αύξηση των νημάτων εκτέλεσης του προγράμματος. Πίστεύουμε ότι για τη συγκεκριμένη συμπεριφορά ευθύνεται το γεγονός ότι το παιχνίδι Cart-pole δεν αποτελεί καλό μέτρο σύγκρισης αλγορίθμων, γιατί θεωρείται ιδιαίτερα απλό. Γι' αυτόν το λόγο οι δομικές καθυστερήσεις των νημάτων σίγουρα επιβραδύνουν σημαντικά τους χρόνους εκτέλεσης των πολυνηματικών εκτελέσεων. Όταν μιλάμε για σημαντική επιβράδυνση, αναφερόμαστε στην ποσοστιαία επιβράδυνση των πολυνηματικών εκτελέσεων σε σχέση

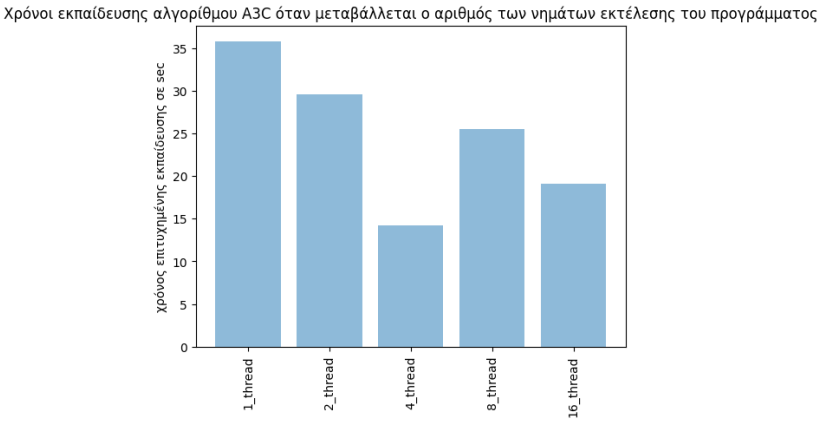


Εικόνα 6.6: Σύγκριση χρόνων εκπαίδευση αλγορίθμων DQN, Double DQN και dueling DQN

με τους χρόνους εκτέλεσεις που θα εμφάνιζαν δυνητικά οι πολυνηματικές εκτελέσεις αν με κάποιο τρόπο εξαλείφονταν οι νηματικές καθυστερήσεις. Επίσης, ενδεχομένως και ο τρόπος υλοποίησης του προγράμματος A3C πιθανώς να ευθύνεται για την μέτρια κλιμάκωση της απόδοσης, βλέπε πίνακας 6.1. Πάντως σε κάθε περίπτωση το πρόγραμμα A3C εμφανίζει συντριπτικά ταχύτερη εκπαίδευση σε σχέση με τα state-of-the-art προγράμματα DQN, Double DQN και Dueling DQN που υλοποιήσαμε προηγουμένως.

Αριθμός νημάτων	Χρόνος εκπαίδευσης προγράμματος A3C
1	35.8s
2	29.6s
4	14.2s
8	25.5s
16	19.1s

Όπως, ακριβώς και στην έρευνα [19] έτσι και εμείς μεταβάλαμε την τιμή της υπερπαραμέτρου learning rate για να δούμε με ποιόν τρόπο μεταβάλλεται ο χρόνος εκπαίδευσης του αλγορίθμου A3C. Τα αποτελέσματα της έρευνάς μας τα παρουσιάζουμε στην εικόνα 6.8. Βλέπουμε ότι ο αλγόριθμος αργεί να εκπαιδευτεί μόνο για κάποιες από τις τιμές της παραμέτρου learning rate.



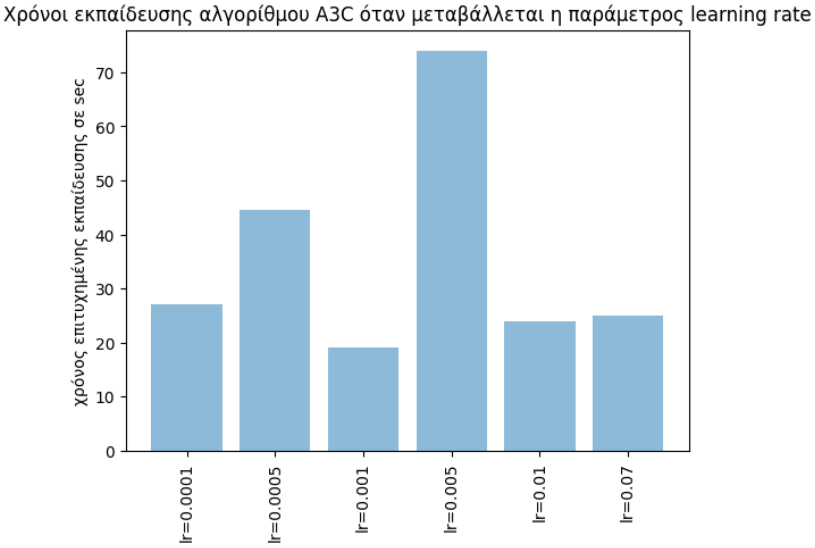
Εικόνα 6.7: Χρόνος εκπαίδευσης αλγορίθμου A3C σε σχέση με την αύξηση των νημάτων

Θα περιμέναμε να μην εκπαιδεύεται καλά μόνο για τις πολύ χαμηλές τιμές της παραμέτρου learning rate, γεγονός το ποίο δε συνέβη.

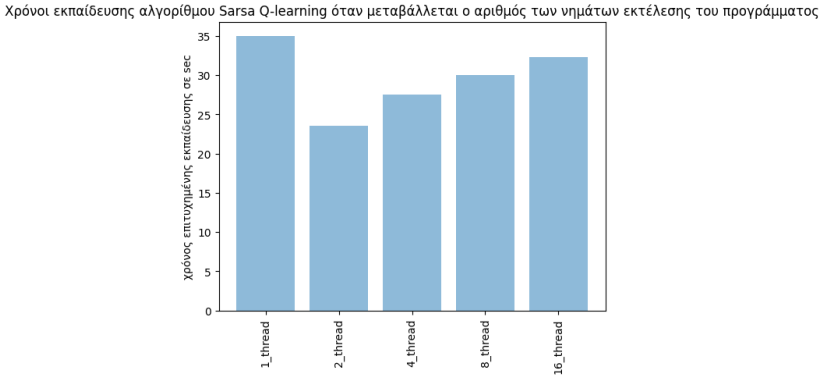
Στη συνέχεια υλοποιήσαμε και εκπαιδεύσαμε τους αλγορίθμους με ονομασίες ασύγχρονος αλγόριθμος ενός βήματος μάθησης-Q, ασύγχρονος αλγόριθμος ενός βήματος Sarsa και ασύγχρονος αλγόριθμος n-βημάτων μάθησης-Q. Παρατηρούμε ότι οι αλγόριθμοι ασύγχρονος αλγόριθμος ενός βήματος μάθησης-Q, ασύγχρονος αλγόριθμος ενός βήματος Sarsa και ασύγχρονος αλγόριθμος n-βημάτων μάθησης-Q δεν κλιμακώνουν ακριβώς όπως θα θέλαμε. Αυτή η συμπεριφορά θεωρούμε ότι παρατηρείται για τους ίδιους λόγους με εκείνους που εξηγήσαμε στην περίπτωση του αλγορίθμου A3C. Στις Εικόνες 6.9 ,6.10 και 6.11 βλέπουμε τους χρόνους εκπαίδευσης των αλγορίθμων ασύγχρονος αλγόριθμος ενός βήματος μάθησης-Q, ασύγχρονος αλγόριθμος ενός βήματος Sarsa και ασύγχρονος αλγόριθμος n-βημάτων μάθησης-Q όταν μεταβάλλεται πάντα, ο αριθμός των νημάτων εκτέλεσης. Παρόλα αυτά θεωρούμε ότι η συμπεριφορά των αλγορίθμων προσεγγίζει την επιθυμητή συμπεριφορά με τις όποιες διαφορές εμφανίζουν στην απόδοσή τους σε σχέση με την έρευνα [19].

6.4 Σύγκριση αλγορίθμων

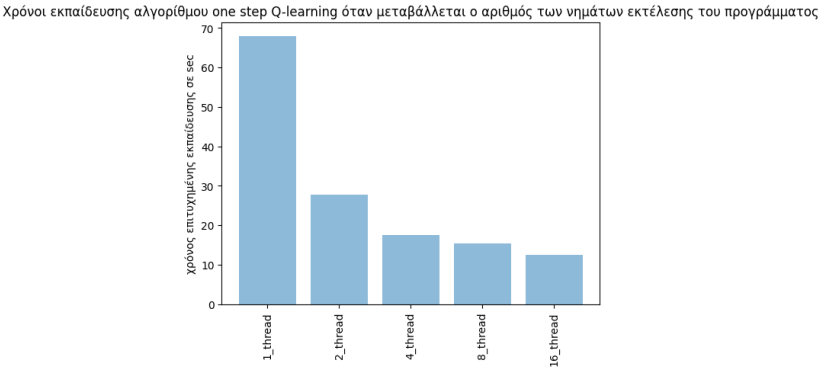
Παρατηρούμε και σχηματικά, βλέπε Εικόνα 6.12 ότι ο αλγόριθμος A3C εκπαιδεύεται ταχύτερα σε σχέση με οποιονδήποτε άλλον αλγόριθμο βαθιάς ενισχυτικής μάθησης τόσο ασύγχρονο όσο και τύπου DQN, τουλάχιστον στις ταχύτερες εκτελέσεις του. Επίσης, βλέπουμε ότι οι αλγόριθμοι ασύγχρονος αλγόριθμος ενός βήματος μάθησης-Q, ασύγχρονος αλγόριθμος ενός βήματος Sarsa και ασύγχρονος αλγόριθμος n-βημάτων μάθησης-Q εμφανίζουν καλύτερη



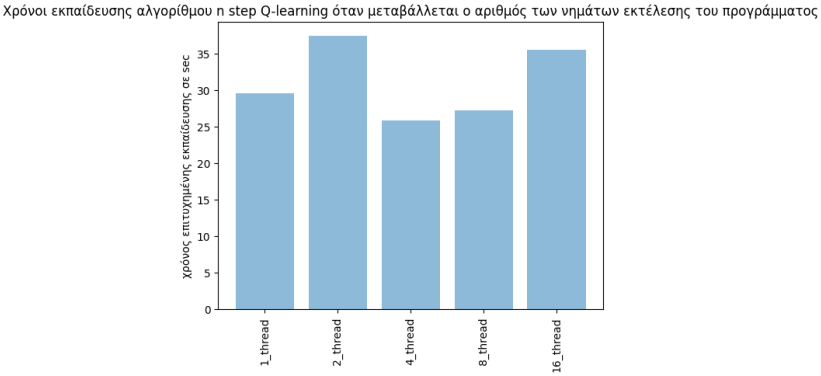
Εικόνα 6.8: Χρόνος εκπαίδευσης αλγορίθμου A3C σε σχέση με την μεταβολή της παραμέτρου learning rate



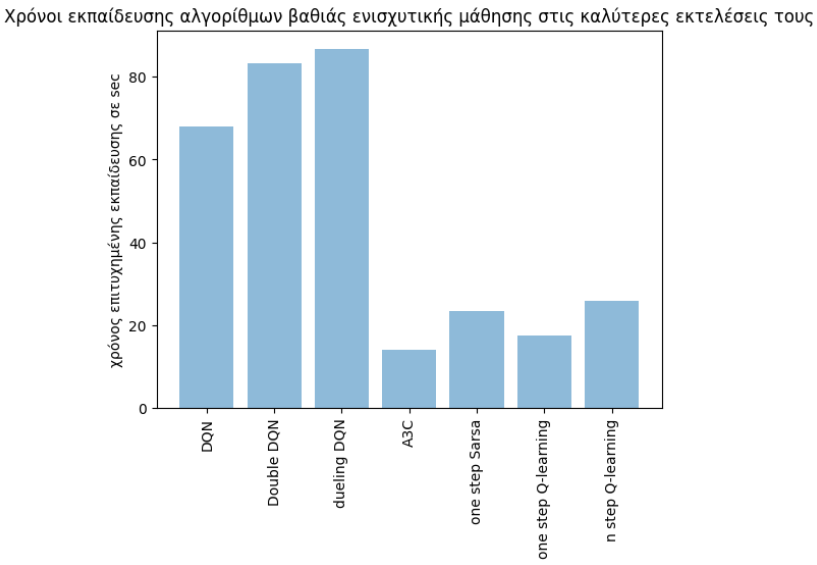
Εικόνα 6.9: Χρόνος εκπαίδευσης ασύγχρονου αλγορίθμου ενός βήματος Sarsa σε σχέση με την αύξηση των νημάτων



Εικόνα 6.10: Χρόνος εκπαίδευσης ασύγχρονου αλγορίθμου ενός βήματος μάθησης-Q σε σχέση με την αύξηση των νημάτων



Εικόνα 6.11: Χρόνος εκπαίδευσης ασύγχρονου αλγορίθμου n βημάτων μάθησης-Q σε σχέση με την αύξηση των νημάτων



Εικόνα 6.12: Καλύτεροι χρόνοι εκπαίδευσης ανά αλγόριθμο

απόδοση στις καλύτερες εκτελέσεις τους σε σχέση με τους αλγορίθμους της κατηγορίας DQN. Εξηγήσαμε, όμως, στην αρχή του υποκεφαλαίου γιατί οι αλγόριθμοι τύπου DQN δεν αποδίδουν βέλτιστα στο συγκεκριμένο απλό πρόβλημα. Θα χρειαζόμασταν ένα πιο σύνθετο παιχνίδι για να κατανοήσουμε και να αναδείξουμε τις εντυπωσιακές δυνατότητες και των συγκεκριμένων αλγορίθμων τύπου DQN. Οι συγκεκριμένοι αλγόριθμοι είναι κατασκευασμένοι ώστε να δέχονται σαν είσοδο εικόνες (πίνακες από πίξελς), χωρίς να πραγματοποιείται καμία απολύτως προεργασία. Στη δική μας περίπτωση, την περίπτωση του απλού προβλήματος Cart-pole τα νευρωνικά δίκτυα τύπου DQN δέχονται σαν είσοδο κατευθείαν ένα διάνυσμα τεσσάρων στοιχείων, το οποίο περιγράφει πλήρως την κατάσταση του περιβάλλοντος και όχι εικόνα από πίξελς. Γι' αυτόν κυρίως το λόγο δεν εξαντλούμε τις δυνατότητες των αλγορίθμων τύπου DQN στην παρούσα διπλωματική εργασία. Στην εικόνα 6.12 παρουσιάζουμε τους καλύτερους χρόνους εκτέλεσης για τους 4 αλγορίθμους ασύγχρονης ενισχυτικής μάθησης και για τους 3 αλγορίθμους τύπου DQN.

6.5 Αξιολόγηση αλγορίθμων

Κατ'αρχάς, επειδή το παιχνίδι Cart-pole είναι σχετικά απλό δεν παρατηρήσαμε σημαντικές διαφορές στον χρόνο εκπαίδευσης και γενικότερα στην απόδοση των προγραμμάτων DQN, Double DQN και Dueling DQN. Επίσης, παρόλο που το παιχνίδι Cart-pole ήταν σχετικά απλό και σε αυτήν την περίπτωση βλέπουμε

ότι ο αλγόριθμος A3C εκπαιδεύεται συντριπτικά ταχύτερα σε σχέση με τους αλγορίθμους της κατηγορίας DQN για κάθε αριθμό νημάτων. Παρατηρούμε, ακόμα, ότι και οι αλγόριθμοι ασύγχρονος αλγόριθμος ενός βήματος μάθησης-Q, ασύγχρονος αλγόριθμος ενός βήματος Sarsa και ασύγχρονος αλγόριθμος n-βημάτων μάθησης-Q εμφανίζουν καλύτερη απόδοση στις καλύτερες εκτελέσεις τους σε σχέση με τους αλγορίθμους της κατηγορίας DQN. Όμως, πρέπει να τονίσουμε ότι οι αλγόριθμοι τύπου DQN είναι σχεδιασμένοι ώστε να επιλύουν πιο πολύπλοκα προβλήματα. Λαμβάνουν κατευθείαν την εικόνα των παιχνιδιών σε πίξελς και χωρίς να χρειάζεται καμία απολύτως προεπεξεργασία εικόνας και εκπαιδεύονται. Στη δική μας περίπτωση αυτό δε συμβαίνει γιατί μπορούν να λάβουν τα δεδομένα χωρίς να έχουν εικόνα αλλά μόνο κάποιες πληροφορίες για την κατάσταση του παιχνιδιού. Γι' αυτόν το λόγο δε βλέπουμε όλες τις δυνατότητες των αλγορίθμων DQN στο συγκεκριμένο πρόβλημα.

Αφού πραγματοποιήσαμε την αξιολόγηση των αλγορίθμων, ας παραθέσουμε συνοπτικά τα πλεονεκτήματα και τα μειονεκτήματα των ασύγχρονων αλγορίθμων ενισχυτικής μάθησης.

Πλεονεκτήματα ασύγχρονων αλγορίθμων ενισχυτικής μάθησης

- Οι συγκεκριμένοι αλγόριθμοι πραγματοποιούν ταχύτερη, καλύτερη και αποδοτικότερη εξερεύνηση του περιβάλλοντος μάθησης σε σχέση με τους αλγορίθμους ενισχυτικής μάθησης που μαθαίνουν από έναν και μόνο πράκτορα.

- Εκτελούνται σε CPU και όχι σε GPU.

- Οι ασύγχρονοι αλγόριθμοι ενισχυτικής μάθησης εκμεταλλεύονται σε μέγιστο βαθμό του πόρους του συστήματος.

- Ξεπερνάνε σε απόδοση, κατά βάση, state-of-the-art αλγορίθμους όπως ο DQN.

- Σύμφωνα με τη βιβλιογραφία [19] φαίνεται να κλιμακώνει η απόδοσή τους υπεργραμμικά με την αύξηση του αριθμού των νημάτων.

- Εμφανίζουν μεγάλα περιθώρια βελτίωσης.

- Είναι σχετικά απλοί στην υλοποίηση.

- Συγκλίνουν με μεγαλύτερη σιγουριά στο ολικό βέλτιστο.

- Εισάγουν τον παραλληλισμό των εργασιών σαν πιθανή λύση στη μηχανική μάθηση.

- Παρέχουν σύγχρονη και ανταγωνιστική μορφή σε θεμελιώδης έννοιες της ενισχυτικής μάθησης, όπως ο αλγόριθμος μάθησης-Q και ο αλγόριθμος Sarsa.

Μειονεκτήματα ασύγχρονων αλγορίθμων ενισχυτικής μάθησης

- Δεν έχουν τελειοποιηθεί ακόμα οι ασύγχρονοι αλγόριθμοι ενισχυτικής μάθησης. Υπάρχει περιθώριο για πολλές βελτιστοποιήσεις στο συγκεκριμένο τομέα ακόμα.

- Καταναλώνουν πολλούς πόρους του συστήματος.

- Δεν είναι τόσο διαδεδομένοι στη διεθνή επιστημονική κοινότητα, όσο άλλοι state-of-the-art αλγόριθμοι, με αποτέλεσμα να μην υπάρχει όσο υποστηρικτικό υλικό χρειάζεται για τους προγραμματιστές στο διαδίκτυο. Πιο συγκεκριμένα δεν υπάρχουν πολλά tutorials για την υλοποίηση των συγκεκριμένων αλγορίθμων.

Κεφάλαιο 7

Συμπεράσματα

7.1 Ανάλυση συμπερασμάτων

Στην παρούσα διπλωματική εργασία περιγράψαμε και υλοποιήσαμε τους ασύγχρονους αλγορίθμους ενισχυτικής μάθησης τον ασύγχρονο αλγόριθμο ενός βήματος μάθησης-Q, τον ασύγχρονο αλγόριθμο ενός βήματος Sarsa, τον ασύγχρονο αλγόριθμο n-βημάτων μάθησης-Q και τον A3C. Ως μέτρο σύγκρισης χρησιμοποιήσαμε τους state-of-the-art αλγορίθμους DQN, double DQN και dueling DQN. Και οι επτά αλγόριθμοι εκπαιδεύθηκαν πάνω στο σχετικά dummy παιχνίδι Cart-pole. Με βάση τα αποτελέσματα είδαμε ότι ο αλγόριθμος A3C εκπαιδεύνονταν επιτυχημένα συντριπτικά ταχύτερα σε σχέση με τους αλγορίθμους DQN, Double DQN και Dueling DQN, όπως και στην έρευνα [19]. Επίσης είδαμε ότι ο ασύγχρονος αλγόριθμος n-βημάτων μάθησης-Q εκπαιδεύονταν ταχύτερα σε αρκετές από τις εκτελέσεις του σε σχέση με τους state-of-the-art αλγορίθμους της κατηγορίας DQN, όχι όμως όσο "γρήγορα" εκπαιδεύονταν ο αλγόριθμος A3C. Τέλος, οι αλγόριθμοι ασύγχρονος αλγόριθμος ενός βήματος μάθησης-Q και ασύγχρονος αλγόριθμος ενός βήματος Sarsa είχαν καλύτερους χρόνους εκπαίδευσης σε σχέση με τους αλγορίθμους της κατηγορίας DQN. Μάλιστα σε πολλές περιπτώσεις ξεπεράσανε την απόδοση του ασύγχρονου αλγορίθμου n-βημάτων μάθησης-Q. Όμως, η εκπαίδευση των συγκεκριμένων αλγορίθμων ήταν πιο αργή σε σχέση με εκείνη του ασύγχρονου αλγορίθμου A3C. Σε κάθε περίπτωση φαίνεται ότι ο αλγόριθμος A3C μπορεί να δώσει state-of-the-art επιδόσεις σε πολλές εφαρμογές της βαθιάς ενισχυτικής μάθησης.

Συμπερασματικά, θεωρούμε ότι και οι 4 αλγόριθμοι ασύγχρονης ενισχυτικής μάθησης που παρουσιάσαμε είναι ιδιαίτερα πετυχημένοι από τη στιγμή που συναγωνίζονται, αν δεν ξεπερνούν σε επιδόσεις, τους ιδιαίτερα δημοφιλείς και πετυχημένους αλγορίθμους DQN, Double DQN και Dueling DQN. Ιδιαίτερα

σημαντικό είναι το γεγονός ότι οι ασύγχρονοι αλγόριθμοι που παρουσιάσαμε είναι διαφορετικής κατηγορίας και όμως εμφανίζουν πολύ ελπιδοφόρα αποτελέσματα. Αν λάβουμε υπόψη και το γεγονός ότι κανένας από τους 4 ασύγχρονους αλγορίθμους δε χρησιμοποιεί την τεχνική replay memory, η οποία πιστεύουμε θα βοηθούσε καθοριστικά στην επιτυχημένη σύγκλιση των ασύγχρονων αλγορίθμων, καταλήγουμε στο συμπέρασμα ότι στο μέλλον η απόδοσή τους μπορεί να είναι ακόμα πιο απολτελεσματική. Παρά το γεγονός ότι οι συγκεκριμένοι ασύγχρονοι αλγόριθμοι είναι πετυχημένοι, θεωρούμε ότι η κατάλληλη χρήση της τεχνικής replay memory θα μπορούσε να τους κάνει ακόμα καλύτερους και πιο αξιόπιστους. Βέβαια αξίζει να σημειώσουμε ότι η χρήση παράλληλων πρακτόρων φαίνεται να καθιστά τους αλγορίθμους πιο ευσταθείς από μόνη της, με αποτέλεσμα να λύνει εν μέρει το πρόβλημα της ευστάθειας, το οποίο επιλύει και η τεχνική replay memory.

7.2 Προτάσεις για μελλοντική έρευνα

Σίγουρα, βελτιώσεις στις μεθόδους παραλληλοποίησης των ασύγχρονων αλγορίθμων θα μπορούσαν να δώσουν ακόμα καλύτερα αποτελέσματα. Μία άλλη κατεύθυνση έρευνας θα ήταν και οι υβριδικοί αλγόριθμοι ασύγχρονης ενισχυτικής μάθησης, όπου οι παράλληλοι πράκτορες θα μπορούσαν ανά ομάδες να ακολουθήσουν ξεχωριστή πολιτική μάθησης. Με αυτόν τον τρόπο, με κατάλληλη διαχείριση θα μπορούσαν οι υβριδικοί αλγόριθμοι να εκμεταλλευτούν τα πλεονεκτήματα διαφορετικών αλγορίθμων και να συγκλίνουν ταχύτερα στο στόχο. Πάντως με τον ένα ή με τον άλλον τρόπο βελτιώσεις στις μεθόδους παραλληλοποίησης των ασύγχρονων αλγορίθμων θα μπορούσαν να βελιτώσουν καθοριστικά την επίδοση των ασύγχρονων αλγορίθμων. Η άποψη του συγγραφέα της παρούσας διπλωματικής είναι ότι η κατάλληλη παραλληλοποίηση των αλγορίθμων μηχανικής μάθησης θα μπορούσε να αποτελέσει μία βασική κατεύθυνση βελτιστοποίησης τέτοιου είδους αλγορίθμων στο μέλλον.

Πέρα, όμως, από την παραλληλοποίηση των αλγορίθμων θεωρούμε ότι αν προστεθεί η τεχνική replay memory στους 4 ασύγχρονους αλγορίθμους που δείξαμε στην παρούσα διπλωματική μπορούν να επιτευχθούν ακόμη καλύτερα αποτελέσματα. Βελτίωση στην απόδοση των ασύγχρονων αλγορίθμων που παρουσιάσαμε αναμένουμε όταν ενσωματωθούν σε αυτούς τελευταίες καινοτομίες της βαθιάς μάθησης και της ενισχυτικής μάθησης[19]. Ακόμα, και σε αυτή την μορφή φαίνονται ιδιαίτερα υποσχόμενοι οι ασύγχρονοι αλγόριθμοι που παρουσιάσαμε με ποιο πετυχημένο τον αλγόριθμο A3C. Θεωρούμε ότι από μόνοι τους οι συγκεκριμένοι αλγόριθμοι μόνο με κάποιες βελτιστοποιήσεις μπορούν να επιφέρουν state-of-the-art αποτελέσματα σε όποιο τομέα θεωρηθεί ότι ταιριάζει να εφαρμοστούν.

Βιβλιογραφία

[1] Artificial neural network. https://en.wikipedia.org/wiki/Artificial_neural_network.

[2] Artificial$_n euron. https://en.wikipedia.org/wiki/Artificial_neuron.$

[3] Machine learning. https://en.wikipedia.org/wiki/Machine_learning.

[4] Object-based reinforcement learning, author=Luca Anzalone, journal=https://towardsdatascience.com/object-oriented-reinforcement-learning-95c284427ea, year=2020.

[5] Supervised learning. https://en.wikipedia.org/wiki/Supervised_learning.

[6] underfitting-and-overfitting-in-machine-learning. https://datascience.foundation/sciencewhitepaper/underfitting-and-overfitting-in-machine-learning.

[7] Unsupervised machine learning. https://en.wikipedia.org/wiki/Unsupervised_learning.

[8] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[9] Dimitri Bertsekas. Distributed dynamic programming. *IEEE transactions on Automatic Control*, 27(3):610–616, 1982.

[10] Durham University Chris G. Willcocks. Reinforcement learning, lecture 9: Model-based methods. https://cwkx.github.io/data/teaching/dl-and-rl/rl-lecture9.pdf.

[11] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, Joelle Pineau, et al. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354, 2018.

[12] Andrei Frumusanu and Ryan Smith. https://towardsdatascience.com/introduction-to-reinforcement-learning-markov-decision-process-44c533ebf8da. https://datascience.foundation/sciencewhitepaper/underfitting-and-overfitting-in-machine-learning, Jul 18, 2019.

[13] Matthew Grounds and Daniel Kudenko. Parallel reinforcement learning with linear function approximation. In *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*, pages 60–74. Springer, 2005.

[14] Jonathan Hui. https://jonathan-hui.medium.com/rl-policy-gradients-explained-9b13b688b146. https://jonathan-hui.medium.com/rl-policy-gradients-explained-9b13b688b146, Sep 12, 2018.

[15] Cheol Kang. Github repository. https://github.com/g6ling/Reinforcement-Learning-Pytorch-Cartpole.git.

[16] Sanyam Kapoor. plicy-gradients-in-a-nutshell. https://towardsdatascience.com/policy-gradients-in-a-nutshell-8b72f9743c5d, Jun 3, 2018.

[17] Sergios Karagiannakos. https://towardsdatascience.com/the-idea-behind-actor-critics-and-how-a2c-and-a3c-improve-them-6dd7dfd0acb8. https://towardsdatascience.com/the-idea-behind-actor-critics-and-how-a2c-and-a3c-improve-them-6dd7dfd0acb8, Nov 17, 2018.

[18] Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

[19] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, New York, USA, 20–22 Jun 2016. PMLR.

[20] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[21] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[22] Arun Nair, Praveen Srinivasan, Sam Blackwell, Cagdas Alcicek, Rory Fearon, Alessandro De Maria, Vedavyas Panneershelvam, Mustafa Suleyman, Charles Beattie, Stig Petersen, et al. Massively parallel methods for deep reinforcement learning. *arXiv preprint arXiv:1507.04296*, 2015.

[23] Cslab Ntua. Παράλληλοι υπολογισμοί σε αρχιτεκτονικές κοινής μνήμης, διαφάνειες μαθήματος εισαγωγή στην παράλληλη με επεξεργασία με εμφαση εφαρμογές μηχανικής μάθησης. http://www.cslab.ntua.gr/courses/parml/files/spring2020/Design$_{O}penMP_{p}resentation-Spring2020.pdf$, 2020.

[24] Vitou Phy. Reinforcement learning concept on cart-pole with dqn. https://towardsdatascience.com/reinforcement-learning-concept-on-cart-pole-with-dqn-799105ca670, 6-10-2019.

[25] Suraj Regmi. Cartpole problem using tf-agents — build your first reinforcement learning application. https://towardsdatascience.com/cartpole-problem-using-tf-agents-build-your-first-reinforcement-learning-application-3e6006adeba7, 2020.

[26] Mohit Sewak. Deep q network (dqn), double dqn, and dueling dqn. pages 95–108, 2019.

[27] Richard S. Sutton and Andrew G. Barto. Reinforcement learning, an introduction, second edition. In *Reinforcement Learning, An Introduction*, 2014-2015.

[28] Tijmen Tieleman, Geoffrey Hinton, et al. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[29] Emo Todorov. Mujoco: Modeling, simulation and visualization of multi-joint dynamics with contact. *Seattle WA: Roboti Publishing*, 2016.

[30] John N Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine learning*, 16(3):185–202, 1994.

[31] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[32] Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

[33] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[34] Ronald J Williams and Jing Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

[35] Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. *Advances in Neural Information Processing Systems*, 34:25476–25488, 2021.

[36] Chris Yoon. https://towardsdatascience.com/understanding-actor-critic-methods-931b97b6df3f. https://towardsdatascience.com/understanding-actor-critic-methods-931b97b6df3f, Feb 6, 2019.

[37] Χριστόφορος Γαβαλάς. Πλοήγηση αυτόνομου οχήματος στον χώρο, με χρήση αλγορίθμων βαθιάς ενισχυτικής μάθησης. Master's thesis, ΗΜΜΥ ΕΜΠ, 2019.

[38] Χρυσόστομος Κανιούρας. Εφαρμογή Αλγορίθμων Ενισχυτικής Μάθησης και Μεταφορά Μάθησης στο sonic the hedgehog. Master's thesis, ΗΜΜΥ ΕΜΠ, 2020.

[39] Τσιατσιάνης Γεώργιος Χρήστος. Μοντελοποίηση Επεξεργαστών arm. Master's thesis, ΗΜΜΥ ΕΜΠ, 2019.