ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Μοντελοποίηση Επιχειρηματικών Αποφάσεων με χρήση Βαθιάς Ενισχυτικής Μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ματθαίος Φικάρδος

Επιβλέπων : Γρηγόρης Μέντζας
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2021

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Μοντελοποίηση Επιχειρηματικών Αποφάσεων με χρήση Βαθιάς Ενισχυτικής Μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ματθαίος Φικάρδος

Επιβλέπων : Γρηγόρης Μέντζας
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 19η Ιουλίου 2021.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Γρηγόρης Μέντζας	Δημήτριος Ασκούνης	Χάρης Δούκας
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2021

(Υπογραφή)

.....

Ματθαίος Φικάρδος

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

Περίληψη

Τα τελευταία χρόνια υπάρχει μια μεγάλη έξαρση στο επιστημονικό πεδίο της Μηχανικής Μάθησης, όπου καθημερινά παρατηρούνται καινοτομίες. Η χρήση της Μηχανικής μάθησης έχει γίνει δεδομένη σε πάρα πολλούς τομείς από Ιατρική και όραση υπολογιστών μέχρι συστήματα συστάσεων και παιχνίδια. Ακόμη στον τομέα των επιχειρήσεων όλο και περισσότερες επιχειρήσεις χρησιμοποιούν την μηχανική μάθηση για την βελτιστοποίηση και διεκπεραίωση των εργασιών τους και την λήψη αποφάσεων.

Στα πλαίσια αυτής της διπλωματικής εργασίας, μελετάται η μοντελοποίηση της διαδικασίας λήψης αποφάσεων σε επιχειρησιακό περιβάλλον με την χρήση βαθιάς ενισχυτικής μάθησης. Συγκεκριμένα έγινε μοντελοποίηση της διαδικασίας λήψης απόφασης σε περιβάλλον όπου μπορούν να εφαρμοστούν διάφοροι αλγόριθμοι ενισχυτικής μάθησης, το οποίο προσαρμόζεται δυναμικά ανάλογα με το συγκεκριμένο πρόβλημα για το οποίο καλείται η ενισχυτική μάθηση να επιλύσει. Για την αξιολόγηση του μοντέλου που κατασκευάσαμε χρησιμοποιήσαμε αλγόριθμους ενισχυτικής μάθησης (Deep Q-learning) όπου εκπαιδεύσαμε έναν πράκτορα για την επίλυση του προβλήματος λήψης απόφασης. Ακόμη εκτελέστηκαν πειράματα αλλάζοντας τις παραμέτρους του μοντέλου έτσι ώστε να προσομοιώσει διαφορετικά σενάρια εφαρμογής, προκειμένου να παρατηρήσουμε την συμπεριφορά της προτεινόμενης προσέγγισης.

Λέξεις Κλειδιά: <<Τεχνητή Νοημοσύνη, Ενισχυτική Μάθηση, Βαθιά Ενισχυτική Μάθηση, Λήψη Αποφάσεων, DQN>>

Abstract

In recent years, there has been a major upsurge in the scientific filed of Machine Learning, where innovations are observed every day. The use of Machine Learning has been taken for granted in many areas from Medicine and Computer Vision to recommending systems and games. Even in the business sector, more and more enterprises are using machine learning to optimize and carry out their related work and make decisions.

In this diploma thesis is being studied the modeling of decision making in an operational environment using deep reinforcing learning. In particular, the decision-making process has been modeled in an environment where different reinforcement learning algorithms can be applied, which is dynamically adapted to the specific problem for which is called upon to resolve. To assess the proposed approach, we've used reinforcement learning algorithms (DQN) where we've trained an agent to solve the decision-making problem. Moreover, various experiments were carried out by changing the parameters of the model in order to simulate different application scenarios, to observe the behavior of the proposed approach.

Keywords: <<Artificial intelligent, Reinforcement learning, Deep Reinforcement learning, Decision-Making, DQN>>

Ευχαριστίες

Καταρχάς θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου τον κ. Γρηγόρη Μέντζα, για την ευκαιρία που μου έδωσε να ασχοληθώ με το συγκεκριμένο θέμα και στο γεγονός ότι με βοήθησε σε όλη την διάρκεια της διπλωματικής. Επίσης θα ήθελα να ευχαριστήσω τους διδακτορικούς ερευνητές Κατερίνα Λεπενιώτη και Αλέξανδρο Βουσδέκη για την συνεχή καθοδήγηση και άμεση στήριξη τους.

Τέλος θα ήθελα να ευχαριστήσω όλους του συμφοιτητές και φίλους μου που με στήριξαν και βοήθησαν καθ' όλη την διάρκεια των φοιτητικών μου χρόνων.

Ματθαίος Φικάρδος

Αθήνα, 19η Ιουλίου 2021

Πίνακας περιεχομένων

Κατάλογος Πινάκων

Κατάλογος Εικόνων

1 Εισαγωγή

1.1 Χρήση ενισχυτικής μάθησης για λήψη αποφάσεων

Τα τελευταία χρόνια γίνεται όλο και περισσότερη χρήση τεχνητής νοημοσύνης και μηχανικής μάθησης σε πολλά επιστημονικά πεδία. Αποτέλεσμα αυτής της χρήσης ήταν η ανάπτυξη μοντέλων και αλγορίθμων τα οποία μπορούν να χρησιμοποιηθούν σε προβλήματα του πραγματικού κόσμου. Σκοπός τους είναι η ψηφιοποίηση και ο αυτοματισμός πολλών διεργασιών που μέχρι πρόσφατα χρειαζόντουσαν επίβλεψη από τον άνθρωπο. Στα πλαίσια του σκοπού αυτού έχουν αναπτυχθεί μοντέλα και αλγόριθμοι για την λήψη αποφάσεων σε προβλήματα και εφαρμογές του πραγματικού κόσμου [1]. Παραδοσιακά χρειαζόταν η επίβλεψη από άνθρωπο για να παρθούν οι αποφάσεις αυτές κάτι που ήταν δύσκολο όταν η λήψη της απόφασης αυτής βασιζόταν σε ανάλυση πολλών δεδομένων και παραμέτρων. Έτσι με την βοήθεια συγκεκριμένων αλγορίθμων, οι οποίοι έχουν πολύ μεγαλύτερη υπολογιστική δύναμη από έναν άνθρωπο μπορούμε να ξεπεράσουμε την πιο πάνω δυσκολία [2]–[4].

1.2 Αντικείμενο διπλωματικής

Στην παρούσα διπλωματική μελετάται η μοντελοποίηση του προβλήματος λήψης επιχειρηματικών αποφάσεων έτσι ώστε να μπορεί να επιλυθεί με βαθιά ενισχυτική μάθηση. Σημαντικό χαρακτηριστικό της μοντελοποίησης μας είναι η ικανότητα του μοντέλου μας να προσαρμόζεται στο συγκεκριμένο πρόβλημα το οποίο καλούμαστε να επιλύσουμε, κάνοντας έτσι δυνατή την χρήση του μοντέλου μας για πολλά και διαφορετικά προβλήματα. Η ικανότητα αυτή βασίζεται στον τρόπο με τον οποίο κατασκευάζουμε το περιβάλλον όπου θα εφαρμοστεί η βαθιά ενισχυτική μάθηση. Το περιβάλλον αυτό κατασκευάζεται δυναμικά έτσι ώστε να προσαρμοστεί στις ανάγκες του συγκριμένου προβλήματος.

Στόχος της συγκεκριμένης εργασίας είναι η μελέτη της απόδοσης του μοντέλου που κατασκευάζουμε για την επίλυση του προβλήματος λήψης απόφασης εφαρμόζοντας αλγορίθμους βαθιάς ενισχυτικής μάθησης. Η επίλυση του προβλήματος βασίζεται στην εκπαίδευση ενός ευφυή πράκτορα (agent), ο οποίος προσπαθεί να βρει την βέλτιστη λήψη απόφασης για την επίλυση του προβλήματος. Η αξιολόγηση της απόδοσης του μοντέλου έγινε εκτελώντας πειράματα χρησιμοποιώντας διάφορους αλγορίθμους ενισχυτικής μάθησης και εφαρμόζοντας τους σε διάφορα σενάρια εφαρμογής προβλημάτων λήψης απόφασης.

1.3 Οργάνωση κειμένου

Η διπλωματική εργασία αποτελείτε από έξι κεφάλαια. Στο Κεφάλαιο 1 (Εισαγωγή) γίνεται αναφοράς για την χρήση ενισχυτικής μάθησης στα προβλήματα λήψης αποφάσεων. Στο Κεφάλαιο 2 (Βιβλιογραφική Επισκόπηση) περιγράφεται το επιστημονικό πεδίο της ενισχυτικής μάθησης και της βαθιάς ενισχυτικής μάθησης αναλύοντας το θεωρητικό τους υπόβαθρο . Στο Κεφάλαιο 3 (Προτεινόμενη Προσέγγιση) επεξηγούμε τον τρόπο με τον οποίο μοντελοποιούμε το πρόβλημα λήψης απόφασης, παραθέτοντας και μερικά παραδείγματα εφαρμογής. Ακολούθως στο Κεφάλαιο 4 (Υλοποίηση) περιγράφουμε τον τρόπο με τον οποίο υλοποιήσαμε το μοντέλο μας εξηγώντας τα εργαλεία που χρησιμοποιήσαμε. Στο Κεφάλαιο 5 (Αξιολόγηση και Συμπεράσματα) παρατίθενται τα πειράματα που εκτελέσαμε και τα αποτελέσματα τους για την αξιολόγηση του μοντέλου μας. Τέλος στο Κεφάλαιο 6 (Συμπεράσματα και Μελλοντική Εργασία) ανακεφαλαιώνουμε με τα τελικά μας συμπεράσματα από το μοντέλο που υλοποιήσαμε και παρουσιάζουμε τον τρόπο με τον οποίο θα μπορούσε να κατευθυνθεί η μελλοντική έρευνα μέσω της συγκεκριμένης διπλωματικής.

2 Βιβλιογραφική Επισκόπηση

2.1 Ενισχυτική Μάθηση

2.1.1 Εισαγωγή

Η Ενισχυτική Μάθηση είναι μια από τις τρείς μεθόδους του επιστημονικού πεδίου Μηχανική Μάθηση (Machine Learning). Οι άλλες δύο μέθοδοι είναι η επιβλεπόμενη (Unsupervised Learning) και μη επιβλεπόμενη μάθηση (Supervised Learning). Ενώ και οι τρείς αυτές μέθοδοι έχουν αρκετά κοινά η ενισχυτική μάθηση διαφέρει ως προς την λειτουργεία της και προσπαθεί να προσεγγίσει τον τρόπο που ο ίδιος ο άνθρωπος μαθαίνει.

2.1.2 Τι είναι η Ενισχυτική Μάθηση

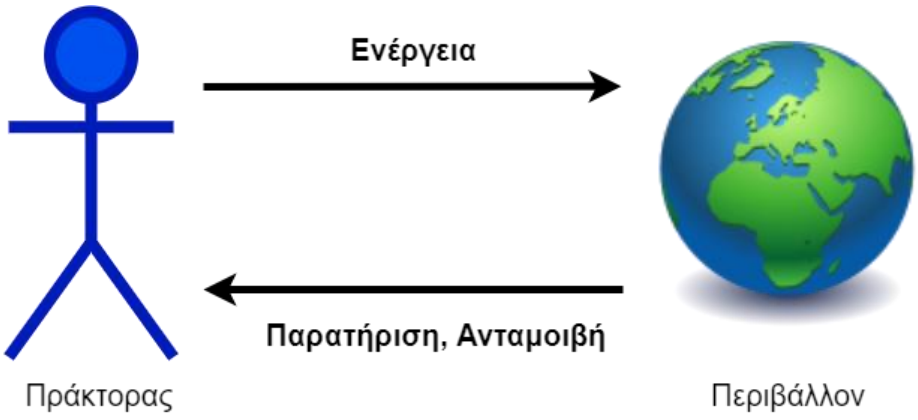
Όπως αναφέραμε η διαφορά της Ενισχυτικής μάθησης με τις υπόλοιπες μεθόδους της Μηχανικής μάθησης είναι ότι προσεγγίζει ή καλύτερα αναπαριστά τον τρόπο με τον οποίο ο άνθρωπος μαθαίνει. Ο άνθρωπος και κάθε έμβιος οργανισμός από την αρχή της ζωής του ξεκίνα να ενεργεί χωρίς να έχει κάποιο δάσκαλο ή προϋπάρχουσα γνώση. Ωστόσο οι ενέργειες οι οποίες εκτελεί έχουν κάποιο αντίκτυπο στο περιβάλλον του οργανισμού όπως επίσης και στον ίδιο, υπάρχει δηλαδή μια επικοινωνία μεταξύ αυτού και του περιβάλλοντος του. Σίγα σιγά μέσο αυτής της επικοινωνίας ο οργανισμός ξεκινά να μαθαίνει πως να αλληλοεπιδρά με το περιβάλλον του και να παράγει πληθώρα πληροφοριών αιτίου – αιτιατού. Βάση αυτών τον πληροφοριών κτίζεται η γνώση του οργανισμού για το περιβάλλον του και αναπτύσσονται στρατηγικές για την επίτευξη διαφόρων στόχων [5]. Όλη αυτή η διαδικασία που αναφέραμε είναι και ο κύριος τρόπος λειτουργείας της ενισχυτικής μάθησης, δηλαδή εκμάθηση μέσο της αλληλεπίδρασης.

Εξ ορισμού η ενισχυτική μάθηση βασίζεται στην αλληλεπίδραση 2 οντοτήτων τις οποίες αποκαλούμε Περιβάλλον (Environment) και Πράκτορα (Agent). Η αλληλεπίδραση αυτώ των οντοτήτων περιγράφεται με τρείς έννοιες:

1. Δράσεις/Ενέργειες (Actions)
2. Καταστάσεις/Παρατηρήσεις (States/Observations)
3. Ανταμοιβές (Rewards)

2.1.3 Μοντελοποίηση Προβλήματος Λήψης Αποφάσεων για επίλυση με Ενισχυτική Μάθηση

Η εκμάθηση του πράκτορα βασίζεται στην συλλογή πληροφορίας από τις πιο πάνω έννοιες μέσο της αλληλεπίδρασης του με το περιβάλλον. Ο πράκτορας δηλαδή εκτελεί μια ενέργεια την οποία στέλνει στο περιβάλλον. Με την σειρά του το περιβάλλον απαντά στον πράκτορα με την παρατήρηση που θα έχει και την ανταμοιβή που θα πάρει μετά από την ενέργεια του. Αυτή η διαδικασία επικοινωνίας μεταξύ των οντοτήτων εκτελείτε πολλές φορές με αποτέλεσμα να γίνεται συλλογή πληροφορίας και εκμάθησή του πράκτορα . Η εκμάθηση του πράκτορα γίνεται με χρήση μεθόδων δυναμικού προγραμματισμού και στατιστικών μοντέλων που ως στόχο έχουν την αύξηση του μακροπρόθεσμου αθροίσματος των ανταμοιβών του πράκτορα [6]. Ο στόχος αυτός μπορεί να επιτευχθεί με χρήση πολλών διαφορετικών αλγορίθμων που θα αναλύσουμε στην συνέχεια του κεφαλαίου.

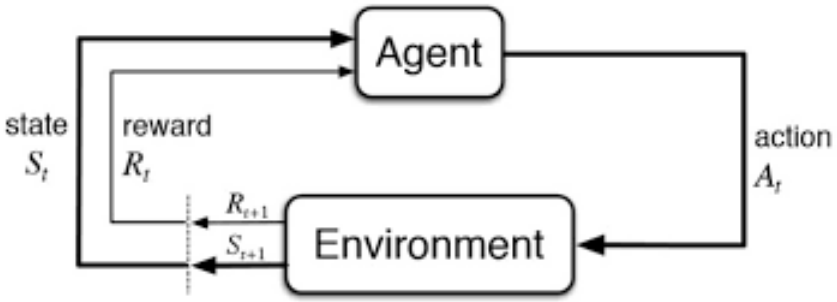


Εικόνα 2-1: Αλληλεπίδραση Πράκτορα-Περιβάλλον

Για την μοντελοποίηση κάποιου προβλήματος με ενισχυτική μάθηση χρειάζεται πρώτα να ορίσουμε τις έννοιες που αναφέραμε πιο πάνω. Οι έννοιες αυτές ορίζονται για κάθε βήμα t που αλληλοεπιδρά ο πράκτορας με το περιβάλλον ως εξής [5]:

- Ως κατάσταση ορίζουμε την αναπαράσταση της θέσης που έχει στο περιβάλλον ο πράκτορας και την συμβολίζουμε με S_t. Επίσης μπορεί να οριστεί και σαν O_t όπου είναι η παρατήρηση που έχει ο πράκτορας για το περιβάλλον.
- Ως ενέργεια ορίζουμε μία από τις επιλογές που έχει ο πράκτορας μας για να αλληλοεπιδράσει με το περιβάλλον του και την συμβολίζουμε με A_t. Η εκτέλεση της ενέργειας έχει ως αποτέλεσμα την αλλαγή της κατάστασης του πράκτορα και την λήψη ανταμοιβής.
- Ως ανταμοιβή ορίζουμε ένα βαθμωτό σήμα ανάδρασης το οποίο υποδεικνύει την επίδραση που είχε η ενέργεια του πράκτορα σε αυτόν, και συμβολίζεται με R_t. Η ανταμοιβή αυτή μπορεί να είναι ένα άμεσο κέρδος για τον πράκτορα ή μια περιγραφή του πόσο καλή είναι η κατάσταση που βρέθηκε ο πράκτορας. Επίσης ο σκοπός του πράκτορα είναι να αυξήσει αύτη την ανταμοιβή.

Η αλληλεπίδραση του πράκτορα με το περιβάλλον βασίζεται στην ανταλλαγή πληροφορίων βάση των πιο πάνω εννοιών που αναφέραμε και το πώς υλοποιείται φαίνεται στην πιο κάτω Εικόνα 2-2.

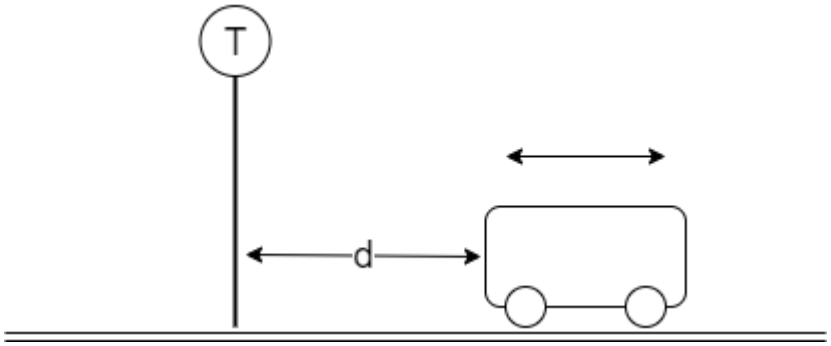


Εικόνα 2-2: Λειτουργεία ενισχυτικής μάθησης [1]

Ο τρόπος με τον οποίο λειτουργεί η ενισχυτική μάθηση βασίζεται στις ανταμοιβές που παίρνει ο πράκτορας ο οποίος προσπαθεί να τις αυξήσει. Ο πράκτορας μέσο της αλληλεπίδρασης που έχει με το περιβάλλον ψάχνει να βρει και να μάθει ποιες ενέργειες, του επιφέρουν την μεγαλύτερη ανταμοιβή σε συγκεκριμένες καταστάσεις στις οποίες βρίσκεται. Βάση των ανταμοιβών που παίρνει μαθαίνει πώς να αλληλοεπιδρά με το περιβάλλον και αναπτύσσει στρατηγικές έτσι ώστε να αυξήσει τη συνολική ανταμοιβή που παίρνει. Αυτή η αλληλεπίδραση του πράκτορα με το περιβάλλον σε διακριτά χρονικά βήματα έχει ως αποτέλεσμα την δημιουργία μιας ακολουθίας από πληροφορίες ή τροχιάς (trajectory) [5] με την εξής μορφή:

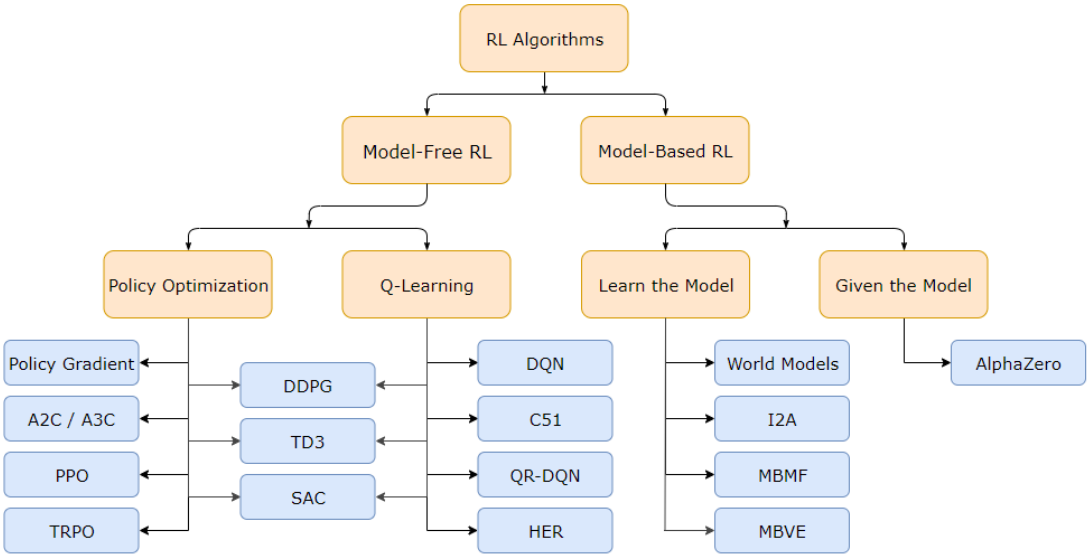
$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, \ldots \quad (1)$$

Για παράδειγμα έστω ο πράκτορας μας είναι υπεύθυνος για ένα καροτσάκι το οποίο πρέπει να φτάσει σε μια συγκεκριμένη τελική θέση (T) όπως φαίνεται στην Εικόνα 2-3 : Αναπαράσταση παραδείγματος με το καρότσι που πρέπει να κινηθεί στην επιθυμητή κατάσταση T. Η κατάσταση του πράκτορα θα ήταν η απόσταση (d) που έχει από την τελική θέση το καροτσάκι ενώ οι κινήσεις που θα μπορούσε να κάνει θα ήταν να κινηθεί μπροστά και πίσω ή να μην κινηθεί. Η ανταμοιβή θα μπορούσε να ήταν αντίθετη σε μέτρο από την απόσταση που έχει το καροτσάκι από την τελική θέση, έτσι ώστε να γίνεται ίση με 0 και να μεγιστοποιείται όταν βρισκόμαστε σε εκείνη τη θέση. Με αυτόν τον τρόπο θα μπορούσαμε να μοντελοποιήσουμε αυτό το απλό παράδειγμα και να είμαστε σε θέση να το επιλύσουμε.



Εικόνα 2-3 : Αναπαράσταση παραδείγματος με το καρότσι που πρέπει να κινηθεί στην επιθυμητή κατάσταση T

Βάσει της πιο πάνω ακολουθίας που παράγεται από την αλληλεπίδραση του πράκτορα γίνεται η υλοποίηση των αλγορίθμων που χρησιμοποιούνται για την εκμάθησή του πράκτορα. Υπάρχουν αρκετές κατηγορίες αλγορίθμων και η χρήση τους εξαρτάται από το συγκεκριμένο πρόβλημα το οποίο προσπαθούν να επιλύσουν. Ο κύριος διαχωρισμός των αλγορίθμων γίνεται μεταξύ δυο κατηγοριών τους Χωρίς Μοντέλο (Model-Free) και Βάσει Μοντέλου (Model-Based) αλγορίθμους. Στην πρώτη κατηγορία οι αλγόριθμοι δεν χρειάζονται πλήρη αναπαράσταση του περιβάλλοντος παρά μόνο την τροχιά που παράγεται από την αλληλεπίδραση του πράκτορα με το περιβάλλον. Στην δεύτερη κατηγορία οι αλγόριθμοι χρειάζονται πρόσβαση στο ίδιο το περιβάλλον για να λειτουργήσουν. Μπορούμε να διακρίνουμε μερικούς από αυτούς του αλγορίθμους στην παρακάτω εικόνα [7].



Εικόνα 2-4: Αλγόριθμοι ενισχυτικής μάθησης [3]

Περαιτέρω κατηγοριοποίηση των αλγορίθμων μπορεί να γίνει και με βάση τη λειτουργία τους. Βάσει της πολιτικής που ακολουθούν μπορούν να χωριστούν σε αυτούς βάσει της πολιτικής (policy based) και βάσει των τιμών (value based), ενώ για τον τρόπο που χρησιμοποιούν τις πολιτικές σε εκτός πολιτικής (off-policy) και με πολιτική (on-policy).

2.2 Μαρκοβιανές διαδικασίες αποφάσεων

Ένα πρόβλημα ενισχυτικής μάθησης μπορεί να περιγραφεί με την βοήθεια της Μαρκοβιανής Διαδικασίας Αποφάσεων (Markov Decision Process - MDP). Οι MDPs είναι επέκταση της Αλυσίδας Markov, με την διαφορά ότι έχει προστεθεί η δυνατότητα λήψης αποφάσεων και η απόδοση ανταμοιβών. Η αλυσίδα Markov χαρακτηρίζεται σαν ένα στοχαστικό μοντέλο για να περιγραφεί μια σειρά από γεγονότα των οποίων η πιθανότητα να συμβούν εξαρτάται μόνο από το άμεσο παρελθόν [8]. Μία MDP ορίζεται ως μια διαδικασία στοχαστικού ελέγχου διακριτού χρόνου και χρησιμοποιείται σαν ένας τυποποιημένος τρόπος για διαδοχική λήψη αποφάσεων. Κύριο χαρακτηριστικό των MDPs είναι ότι δεν λαμβάνουν υπόψη μόνο άμεσες ανταμοιβές αλλά και μεταγενέστερες καταστάσεις και μελλοντικές ανταμοιβές [9]. Έτσι οι MDPs είναι ένας πολύ καλός και σωστός τρόπος για να αναπαραστήσουμε ένα πρόβλημα ενισχυτικής μάθησης, καθώς οι ενέργειες που εκτελεί ο πράκτορας έχουν αντίκτυπο στις μελλοντικές ενέργειες και ανταμοιβές του πράκτορα.

2.2.1 Ορισμός Μαρκοβιανής Διαδικασίας Αποφάσεων

Μία MDP ορίζεται ως μια πλειάδα τεσσάρων στοιχείων (S, A, P^a, R) [5], όπου το κάθε ένα αναπαριστά αντίστοιχα:

- S το σύνολο των καταστάσεων

$$\mathrm{S} = \{S_0, S_1, S_2, S_3, \dots, S_v\}$$

- A το σύνολο των ενεργειών

$$A = \{\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_v\}$$

- P^a ο πίνακας μεταβάσεων με τις πιθανότητες για να μεταβούμε στην κατάσταση s' την στιγμή t+1, δεδομένου ότι την στιγμή t βρισκόμαστε στην κατάσταση s και εκτελούμε την ενέργεια a

$$P_{ss}^a = \Pr[S_t = s' | S_{t-1} = s, A_{t-1} = a]$$

- R το σύνολο των άμεσων ανταμοιβών

$$R = \{R_0, R_1, R_2, R_3, \dots, R_v\}$$
$$\mu\varepsilon\ R_s^a = E[R_t | S_{t-1} = s, A_{t-1} = a]$$

2.2.1.1 Μαρκοβιανή Ιδιότητα

Η χρήση των MDPs στην ενισχυτική μάθηση γίνεται λόγω της σημαντικής μαρκοβιανής ιδιότητας που έχουν. Σύμφωνα με την ιδιότητα αυτή για τις μελλοντικές αποφάσεις μπορούμε να βασιστούμε μόνο στην τωρινή κατάσταση που βρισκόμαστε. Δηλαδή το μέλλον εξαρτάται μόνο από την παρούσα κατάσταση και είναι ανεξάρτητο από το παρελθόν.

$$\Pr[S_{t+1} | S_t] = \Pr[S_{t+1} | S_1, S_2, .., S_t] \qquad (2)$$

2.2.1.2 Άθροισμα Ανταμοιβών

Όπως αναφέραμε ο πράκτορας προσπαθεί να αυξήσει την συνολική του ανταμοιβή της οποίας το άθροισμα μέχρι μια τελική κατάσταση T το ορίζουμε ως [5]:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T \qquad (3)$$

Αυτό το άθροισμα όμως δίνει την ίδια βαρύτητα σε όλες τις ανταμοιβές, κάτι το οποίο πρακτικά δεν είναι και τόσο σωστό διότι οι πιο πρόσφατες αμοιβές έχουν περισσότερη βαρύτητα.

Για αυτόν τον λόγο στο άθροισμα των ανταμοιβών προσθέτουμε τον όρο έκπτωσης γ, του οποίου ο ρόλος είναι να μειώσει την βαρύτητα των μελλοντικών ανταμοιβών. Έτσι το άθροισμα γίνεται ως εξής:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{n=0}^{\infty} \gamma^n R_{t+n+1}\,, \text{όπου } \gamma \in [0,1] \qquad (4)$$

Ο συντελεστής γ παίρνει τιμές από 0 μέχρι 1, αναλόγως με την βαρύτητα που θέλουμε να δώσουμε στις μελλοντικές ανταμοιβές.

2.2.1.3 Συνάρτηση Τιμών

Πολύ σημαντικός είναι και ο ορισμός της συνάρτησης τιμών (Value Function), σκοπός της οποίας είναι να αποδώσει την αξία μιας κατάστασης στην οποία βρίσκεται ο πράκτορας. Ο υπολογισμός αυτής της αξίας γίνεται με βάση τις αναμενόμενες μελλοντικές ανταμοιβές και το άθροισμα τους [5]. Η συνάρτηση τιμών ορίζεται ως εξής:

$$\mathrm{V}(s) = E[G_t | S_t = s] = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \qquad (5)$$

Και αναδρομικά γράφεται :

$$\mathrm{V(s)} = R_{t+1} + \gamma\, V(S_{t+1}) \qquad (6)$$

Ο σωστός ορισμός και υπολογισμός της συνάρτησης τιμών είναι σημαντικοί καθώς είναι ο τρόπος με τον οποίο ο πράκτορας μπορεί να καταλάβει εάν η ενέργεια που εκτέλεσε, επέφερε θετικό ή αρνητικό αποτέλεσμα στην κατάσταση του ως προς το περιβάλλον.

2.2.1.4 Πολιτική

Εξίσου σημαντική με την συνάρτηση τιμών είναι και η πολιτική που ακολουθεί ο πράκτορας. Η πολιτική είναι ο τρόπος με τον οποίο επιλέγει ο πράκτορας ποια από τις ενέργειες να εκτελέσει, και συγκεκριμένα η πολιτική αντιστοιχεί όλες τις ενέργειες με πιθανότητες επιλογής τους βάση κάποιας κατάστασης [5]. Ο λόγος που χρησιμοποιούμε μια πολιτική είναι

για να εξισορροπήσουμε την εξερεύνηση (exploration) και την εκμετάλλευση (exploitation) του πράκτορα. Πιο αναλυτική περιγραφή για την πολιτική, την εξερεύνηση και την εκμετάλλευση θα κάνουμε πιο κάτω σε αυτό το κεφάλαιο. Η πολιτική συμβολίζεται ως $\pi(a|s)$ και αντιστοιχεί στην πιθανότητα να επιλεγεί η ενέργεια At = a εάν St = s τη χρονική στιγμή t [5].

$$\pi(a|s) = P[A_t = a \mid S_t = s] \quad (7)$$

2.2.1.5 Συνάρτηση Τιμής-Ενέργειας

Η συνάρτηση Τιμής-Ενέργειας ορίζεται ως η αναμενόμενη τιμή των ανταμοιβών ξεκινώντας από την κατάσταση s, παίρνοντας την ενέργεια a και ακολουθώντας την πολιτική π.

$$Q_\pi(s,a) = E_\pi[G_t|S_t = s, A_t = a] \quad (8)$$

2.3 Μέθοδοι μάθησης

Ένας από τους πιο σημαντικούς παράγοντες για την επίλυση προβλημάτων ενισχυτικής μάθησης είναι η μέθοδος με την οποία θα προσπαθήσουμε να επιλύσουμε το συγκεκριμένο πρόβλημα. Αρκετές προσπάθειες έχουν γίνει για την επίλυση διαφόρων προβλημάτων ενισχυτικής μάθησης, τα οποία είχαν διαφορετικά χαρακτηριστικά, με αποτέλεσμα να αναπτυχθούν διάφορες μέθοδοι επίλυσης τους. Η σωστή επιλογή της μεθόδου και των αλγορίθμων επίλυσης είναι εξίσου σημαντική με την ορθή δημιουργία του περιβάλλοντος. Εάν δεν χρησιμοποιηθούν οι κατάλληλες μέθοδοι και δεν οριστεί σωστά το περιβάλλον, αποτέλεσμα θα είναι η ανικανότητα του πράκτορα για την σωστή και βέλτιστη λύση του προβλήματος. Οι μέθοδοι που θα αναλύσουμε πιο κάτω βασίζονται στην επίλυση προβλημάτων ενισχυτικής μάθησης τα οποία μοντελοποιήθηκαν με MDPs.

2.3.1 Δυναμικός Προγραμματισμός (DP)

Ο όρος δυναμικός προγραμματισμός (dynamic programming – DP) αναφέρεται σε αλγόριθμους που μπορούν να χρησιμοποιηθούν για τον υπολογισμό της βέλτιστης πολιτικής δεδομένου ενός τέλειου μοντέλου του περιβάλλοντος σαν MDP. Δυστυχώς όμως η πρακτική εφαρμογή και χρήση αυτών των αλγορίθμων είναι πολύ περιορισμένη καθώς προϋποθέτουν

ένα τέλειο μοντέλο και έχουν μεγάλη υπολογιστική πολυπλοκότητα. Αν και η χρήση τους δεν είναι μεγάλη, είναι πολύ σημαντικοί από θεωρητικής άποψης αφού παρέχουν το απαραίτητο υπόβαθρο για την κατανόηση των πρακτικά εφαρμοσμένων μεθόδων και αλγορίθμων ενισχυτικής μάθησης. Οι μέθοδοι και οι αλγόριθμοι αυτοί έχουν ακριβώς τον ίδιο στόχο με τους αλγόριθμους δυναμικού προγραμματισμού με μικρότερο υπολογιστικό κόστος και χωρίς να υπάρχει ένα τέλειο μοντέλο για το περιβάλλον [5].

Για την χρήση δυναμικού προγραμματισμού θεωρούμε ότι το περιβάλλον μας αναπαρίσταται με μια πεπερασμένη MDP. Συνεπώς τα σύνολα καταστάσεων S, ενεργειών A, ανταμοιβών R και οι πιθανότητες μετάβασης P είναι όλα πεπερασμένα. Σε αυτού του είδους τα προβλήματα κύριος σκοπός του δυναμικού προγραμματισμού και γενικά της ενισχυτικής μάθησης είναι η χρήση συναρτήσεων τιμών για την οργάνωση και τη δομή της αναζήτησης των καλύτερων πολιτικών. Η εύρεση της καλύτερης πολιτικής μπορεί να διαχωριστεί στην αξιολόγηση της πολιτικής (policy evaluation) και στην βελτίωση της πολιτικής (policy improvement). Η αξιολόγηση της πολιτικής αναφέρεται στον επαναληπτικό υπολογισμό των συναρτήσεων τιμών για μια σταθερή πολιτική, ενώ η βελτίωση πολιτικής αναφέρεται στον υπολογισμό μιας βελτιωμένης πολιτικής για μια σταθερή συνάρτηση αξίας [5].

Βάση των πιο πάνω προκύπτει το εξής θεώρημα βελτίωσης πολιτικής (policy improvement theorem) [10]:

Θεώρημα 1:

Για κάθε MDP ισχύει ότι:

- Υπάρχει μια βέλτιστη πολιτική π* τέτοια ώστε $\pi^* \geq \pi, \forall \pi$
- Κάθε βέλτιστη πολιτική επιτυγχάνει την βέλτιστη συνάρτηση τιμών εάν επιτυγχάνει και την βέλτιστη συνάρτηση τιμής-ενέργειας

$$Q_{\pi^*}\big(s, \pi(s)\big) \geq V_\pi(s), \forall s \in S \Rightarrow V_{\pi^*}(s) \geq V_\pi(s)$$

Βάση του Θεώρημα 1: και με την βοήθεια της εξίσωσης του Bellman [11] μπορούμε να υπολογίσουμε τις δύο αυτές συναρτήσεις. Δηλαδή με την εξίσωση (3) υπολογίζουμε την άμεση ανταμοιβή και με την εξίσωση (4) την ανταμοιβή από το σημείο που ήμαστε και μετά.

$$V_{\pi\prime}(s) = max_a \sum_{s',r} P_{ss}^a(s', r\,|s, a)[r + \gamma V_{\pi\prime}(s')] \tag{9}$$

$$Q_{\pi\prime}(s) = \sum_{s',r} P_{ss}^a(s', r\,|s, a)[r + \gamma\, max_{a\prime}\, Q_{\pi\prime}(s', a')] \tag{10}$$

Επειδή τα περισσότερα προβλήματα ενισχυτικής μάθησης στον πραγματικό κόσμο δεν διαθέτουν ένα τέλειο μοντέλο και το μέγεθος τους καθιστά πολύ μεγάλη υπολογιστική πολυπλοκότητα, πρακτικά δεν γίνεται χρήση δυναμικού προγραμματισμού. Έτσι γίνεται χρήση άλλων μεθόδων και αλγορίθμων για την ενισχυτική μάθηση αντί του δυναμικού προγραμματισμού.

2.3.2 Μέθοδοι Monte Carlo (MC)

Σε αντίθεση με τον δυναμικό προγραμματισμό οι μέθοδοι Monte Carlo δεν προϋποθέτουν πλήρη γνώση του περιβάλλοντος. Η διαφορά τους είναι ότι για την εκμάθηση χρειάζονται μόνο δείγματα εμπειρίας (experience) από τις καταστάσεις, τις ενέργειες και τις ανταμοιβές του περιβάλλοντος. Αυτό σημαίνει ότι δεν χρειάζεται να γνωρίζουμε ολόκληρο τον πίνακα πιθανοτήτων μετάβασης. Ο τρόπος που λειτουργούν αυτές οι μέθοδοι είναι καθώς αλληλοεπιδρούν με το περιβάλλον τους παράγουν δείγματα εμπειρίας σε μορφή επεισοδίων. Μετά το πέρας κάθε επεισοδίου αναδρομικά ανανεώνουν τις εκτιμήσεις των συναρτήσεων τιμών και πολιτικών με βάση τον εμπειρικό μέσο όρο από τις ανταμοιβές του επεισοδίου.

$$V(S_t) = V(S_t) + \alpha[\, G_t - V(S_t)\,] \tag{11}$$

2.3.3 Μάθηση Χρονικών Διαφορών (Temporal-Difference Learning -TD)

Ένας πολύ καλός τρόπος μάθησης είναι ο συνδυασμός του δυναμικού προγραμματισμού με τις μεθόδους Monte Carlo, και αυτός είναι η μάθηση Χρονικών Διαφορών (TD). Σύμφωνα με την μάθηση TD, η ενημέρωση των εκτιμήσεων γίνεται εν μέρη βάση των εκτιμήσεων που ήδη έχει εις γνώση του ο πράκτορας, χωρίς να περιμένει για ένα οριστικό αποτέλεσμα (boostraping). Ο τρόπος με τον οποίο γίνεται αυτό είναι με το να μετακινείται προς την σωστή κατεύθυνση η συνάρτηση τιμών χρησιμοποιώντας την άμεση ανταμοιβή αλλά και τις ήδη αποθηκευμένες τιμές [12]. Αποτέλεσμα είναι η πιο κάτω συνάρτηση τιμών:

$$V(S_t) = V(S_t) + \alpha[\, R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\,] \tag{12}$$

2.3.4 Εξερεύνηση και Εκμετάλλευση

Όπως αναφέραμε και πριν σε αυτό το κεφάλαιο, η χρήση των πολιτικών στην ενισχυτική μάθηση γίνεται για να εξισορροπηθεί και να βελτιστοποιηθεί η εξερεύνηση και η εκμετάλλευση του πράκτορα. Με τον όρο εξερεύνηση εννοούμε την εκτέλεση ενεργειών και τη μετάβαση σε καταστάσεις, που δεν έχουμε επισκεφθεί ακόμα, καθώς μπορεί αυτές να οδηγούν στην συμπεριφορά για το βέλτιστο άθροισμα ανταμοιβών. Με τον όρο εκμετάλλευση εννοούμε την επιλογή ενεργειών που μας οδηγούν σε ήδη εξερευνημένες καταστάσεις που μας δίνουν το καλύτερο άθροισμα ανταμοιβών που έχουμε βρει μέχρι εκείνη στη στιγμή. Όπως καταλαβαίνουμε η ισορροπία μεταξύ των δύο είναι σημαντική. Εάν επικεντρωθούμε πολύ στην εξερεύνηση χάνουμε το στόχο που είναι να αυξήσουμε την ανταμοιβή και εάν επικεντρωθούμε στην εκμετάλλευση, πιθανών να μην βρούμε τις καλύτερες ανταμοιβές βάση των οποίων θα πρέπει να κτίσουμε την βέλτιστη πολιτική μας. Φυσικά η ισορροπία μεταξύ αυτών των δύο, δεν είναι σταθερή και πολλές φορές εξαρτάται από την φύση του προβλήματος ενισχυτικής μάθησης. Ακόμη υπάρχουν δύο μέθοδοι για την εφαρμογή πολιτικών, η On-policy και η Off-policy. Στην πρώτη χρησιμοποιείται η ίδια πολιτική για την εξερεύνηση και τον υπολογισμό της συνάρτησης τιμής-ενέργειας, ενώ στη δεύτερη χρησιμοποιείται μια πολιτική για την εξερεύνηση και μια άλλη πολιτική για τον υπολογισμό της συνάρτησης τιμής-ενέργειας.

2.3.4.1 Ε-Άπληστη Πολιτική

Μια από τις ποιο απλές πολιτικές που προσπαθούν να βρουν αυτήν την ισορροπία είναι η Ε-Άπληστη πολιτική (e-greedy policy). Σύμφωνα με αυτή την πολιτική βάση μιας πιθανότητας ε επιλέγουμε άπληστα την ενέργεια με την καλύτερη τιμή της συνάρτησης τιμής-ενέργειας, αλλιώς επιλέγουμε μια τυχαία ενέργεια. Φυσικά λόγο της απλοϊκότητας της πολιτικής αυτής υπάρχει μια αρκετά μεγάλη τυχαιότητα στην εξερεύνηση και στην επιλογή των ενεργειών [13].

2.3.4.2 Boltzman policy

Η πολιτική Boltzman προσπαθεί να απαλείψει την μεγάλη τυχαιότητα της Ε-Άπληστης πολιτικής με το να αποδίδει πιθανότητα επιλογής σε κάθε ενέργεια ξεχωριστά . Η πιθανότητα που θα δοθεί σε κάθε ενέργεια βασίζεται στην τιμή που έχει η συνάρτηση τιμής-ενέργειας για αυτήν και υπολογίζεται με τον εξής τρόπο:

$$\pi(s,a) = \frac{e^{\frac{Q(s,a)}{T}}}{\sum_i^A e^{\frac{Q(s,a^i)}{T}}} \tag{13}$$

Όπου A είναι όλες οι ενέργειες που μπορούν να εκτελεστούν στην κατάσταση s, και T είναι η παράμετρος θερμοκρασίας η οποία είναι υπεύθυνη για την εξερεύνηση. Όσο μεγαλύτερη είναι αυτή η παράμετρος τόσο μεγαλύτερη είναι και η εξερεύνηση.

Με αύτη την πολιτική μειώνεται η τυχαιότητα μεταξύ της επιλογής των ενεργειών, δίνοντας μεγαλύτερη πιθανότητα στις ενέργειες με την καλύτερη τιμή βάσει της συνάρτησης τιμής-ενέργειας [14], [15].

2.3.4.3 Max - Boltzmann policy

Η συγκεκριμένη πολιτική χρησιμοποιεί την Ε-Άπληστη πολιτική μαζί με την Boltzman πολιτική. Σύμφωνα με την Max Boltzmann επιλέγεται η άπληστη ενέργεια με πιθανότητα 1-ε αλλιώς επιλέγεται μια ενέργεια βάσει των πιθανοτήτων της πολιτικής Boltzmann. Με την εισαγωγή αυτής της υπερπαραμέτρου (hyperparameter) μπορούμε να ελέγξουμε καλύτερα την εξερεύνηση του περιβάλλοντος [13].

2.3.5 Q-learning

Μία μεγάλη καινοτομία στην ενισχυτική μάθηση ήταν η ανάπτυξη του αλγορίθμου Q-learning. Ο αλγόριθμος αυτός εντάσσεται στις κατηγορίες Off-policy και TD Control και έχει σκοπό να μάθει να δρα βέλτιστα σε ένα περιβάλλον κατασκευασμένο από MDP βάσει της εμπειρίας που ανακτά από αυτό. Ο αλγόριθμος αυτός βασίζεται στην πιο κάτω εξίσωση [5], [16]:

$$Q(S_t, A_t) = Q(S_t, A_t) + a[R_{t+1} + \gamma\, max_a Q(S_{t+1}, a) - Q(S_t, A_t)] \qquad (14)$$

Στην περίπτωση του Q-learning η συνάρτηση τιμής-ενέργειας Q, προσεγγίζει την βέλτιστη Q’ με την άπληστη πολιτική $[max_a Q(S_{t+1}, a)]$. Για την εξερεύνηση όμως μπορεί να χρησιμοποιηθεί κάποια άλλη πολιτική όπως αυτές που αναφέραμε πιο πάνω.

Με όλα όσα έχουμε αναφέρει μέχρι τώρα μπορούμε να διατυπώσουμε τον αλγόριθμο Q-learning [5]:

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

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

Εικόνα 2-5: Αλγόριθμος Q-learning

Αν και ο αλγόριθμος Q-learning επιτυγχάνει την βέλτιστη συνάρτηση τιμής-ενέργειας υποθέτει πεπερασμένο πλήθος καταστάσεων, και δεν κάνει χρήση γενικεύσεων που μπορούν να υπάρχουν, με αποτέλεσμα να καθιστά δύσκολη τη χρήση του σε πολλά προβλήματα.

2.4 Βαθιά Ενισχυτική Μάθηση (Deep Reinforcement Learning)

Κατά την ανάπτυξη της ενισχυτικής μάθησης για την υλοποίηση της έγινε χρήση νευρωνικών δικτύων με αποτέλεσμα να προκύψει ο όρος Βαθιά Ενισχυτική μάθηση. Η δυνατότητα γενίκευσης ήταν η ανάγκη που οδήγησε στην χρήση των νευρωνικών δικτύων καθώς αυτά χρησιμοποιούνται για την δημιουργία τεχνητής νοημοσύνης. Η χρήση τους μας διευκολύνει στην επεξεργασία μεγάλου όγκου δεδομένων και στην εξαγωγή χρήσιμων πληροφοριών από αυτά. Τα τελευταία χρόνια υπήρξαν πολλές καινοτομίες στην χρήση της βαθιάς μάθησης με πρωτοπόρο την εταιρία DeepMind. Η DeepMind μέσο της βαθιάς ενισχυτικής μάθησης κατάφερε να εκπαιδεύσει πράκτορες που μπορούν να παίζουν παιχνίδια της κονσόλας Atari, οι οποίοι μάλιστα ξεπερνούν σε απόδοση κάθε ανθρώπινο παίχτη. Επίσης η πιο σημαντική μέχρι τώρα καινοτομία ήταν το σύστημα AlphaGo της εταιρίας αυτής το οποίο έμαθε να παίζει το παιχνίδι Go, ένα πολύπλοκο παιχνίδι που θεωρείται πιο δύσκολο και από το σκάκι. Το AlphaGo ήταν το πρώτο υπολογιστικό πρόγραμμα που κατάφερε να νικήσει επαγγελματίες παίχτες του Go και θεωρείται ο καλύτερος παίχτης Go στην ιστορία [17], [18]. Αν και οι εφαρμογές που αναφέραμε αποτελούν παιχνίδια, έδωσαν το έναυσμα για την χρήση

της βαθιάς ενισχυτικής μάθησης για την εφαρμογή της σε πιο πρακτικά προβλήματα του πραγματικού κόσμου.

2.4.1 Τεχνητά Νευρωνικά Δίκτυα

Η βασική ιδέα για τα τεχνητά νευρωνικά δίκτυα είναι η αναπαράσταση του τρόπου λειτουργείας του ανθρώπινου εγκεφάλου και των νευρώνων του. Τα τελευταία χρόνια το συγκεκριμένο επιστημονικό πεδίο είχε τεράστια ανάπτυξη και η χρήση τους έχει καθιερωθεί σε πάρα πολλούς τομείς. Καθοριστική για την ανάπτυξη τους ήταν η ισχυροποίηση των υπολογιστικών συστημάτων που χρησιμοποιούμε και η δυνατότητα τους να επεξεργάζονται και να λαμβάνουν υπόψη μεγάλους όγκους δεδομένων.

2.4.1.1 Χαρακτηριστικά Τεχνητών Νευρωνικών Δικτύων

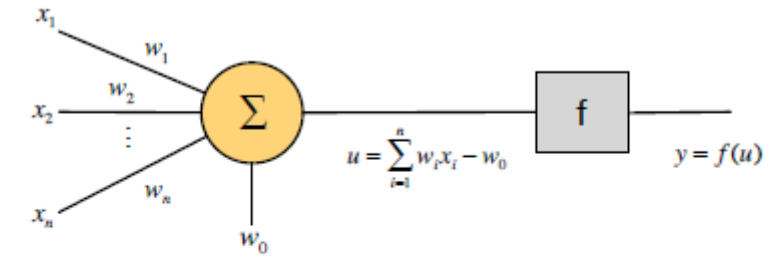
Η δημιουργία ενός νευρωνικού δικτύου βασίζεται στην συνένωση και επικοινωνία νευρώνων, όπως και ο ανθρώπινος εγκέφαλος. Ο κάθε νευρώνας αποτελεί έναν γραμμικό συνδυαστή ο οποίος παίρνει σαν είσοδο σήματα με κάποιες τιμές xi και τις προσθέτει μαζί με έναν σταθερό όρο b ή w_0 (bias). Ακολούθως βάση μιας συνάρτησης, την λεγόμενη συνάρτηση ενεργοποίησης, ο νευρώνας βγάζει στην έξοδο του σαν σήμα την τιμή 0 εάν δεν ενεργοποιηθεί και την τιμή 1 εάν ενεργοποιηθεί.

Τα νευρωνικά δίκτυα κατασκευάζονται από επίπεδα νευρώνων (layers), τα οποία συνδέονται μεταξύ τους έτσι ώστε το σήμα εξόδου των νευρώνων ενός επιπέδου να είναι το σήμα εισόδου για το επόμενο επίπεδο. Το πρώτο επίπεδο το ονομάζουμε επίπεδο εισόδου, το τελευταίο επίπεδο εξόδου και τα ενδιάμεσα κρυφό επίπεδο. Στις ενώσεις μεταξύ των επιπέδων μπορούμε να αποδώσουμε βάρη (weights), των οποίων ρόλος είναι να δώσουν βαρύτητα για το πόσο σημαντικά είναι τα σήματα στην είσοδο των νευρώνων. Τα σήματα εισόδου και τα βάρη τους μπορούν να γραφτούν σε μορφή πινάκων X και W αντίστοιχα [19]. Έτσι μαθηματικά μπορούμε να εκφράσουμε την λειτουργεία ενός νευρώνα με τον πιο κάτω τρόπο:

$$y = f[\sum_{i=1}^{n} w_i x_i - w_0] \tag{15}$$

Όπου y είναι η έξοδος του νευρώνα και όπου f η συνάρτηση ενεργοποίησης.

Στη Εικόνα 2-6: Λειτουργεία νευρώνα, βλέπουμε την λειτουργεία ενός νευρώνα όπως την περιγράψαμε πιο πάνω. Ο νευρώνας δέχεται τα σήματα εισόδου x_i με τα βάρη w_i τα οποία αθροίζει μαζί με τον σταθερό όρο w_0. Ακολούθως περνά το άθροισμα αυτό στην συνάρτηση ενεργοποίησης η οποία με την σειρά της υπολογίζει το σήμα εξόδου.



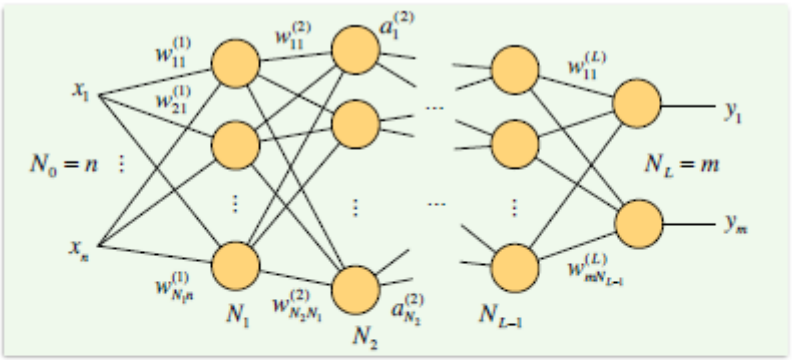
- x_i, w_i, y — Είσοδοι, συναπτικά βάρη, έξοδος
- u — Διέγερση (γινόμενο διανυσμάτων εισόδου-βαρών)
- f — Συνάρτηση ενεργοποίησης
- w_0 — Κατώφλι ενεργοποίησης

Εικόνα 2-6: Λειτουργεία νευρώνα

Αντίστοιχα για τα νευρωνικά δίκτυα η εξίσωση (10) παίρνει ακόμη μία παράμετρο που είναι το επίπεδο στο οποίο την χρησιμοποιούμε.

$$a_i^K = f[\sum_{j=1}^{N_{K-1}} w_{ij}^K a_j^{K-1} - w_{i0}^K] \tag{16}$$

Στην Εικόνα 2-7 παρουσιάζουμε την συνολική εικόνα ενός τεχνητού νευρωνικού δικτύου. Κάθε επίπεδο αποτελείτε από πολλούς νευρώνες των οποίων η έξοδος, γίνεται η είσοδος για το επόμενο επίπεδο. Στο τελευταίο επίπεδο λαμβάνουμε την έξοδο ολόκληρου του νευρωνικού.



- N_i — Πλήθος νευρώνων στο στρώμα i
- $w_{ij}^{(K)}$ — Βάρος από το νευρώνα j του στρώματος K-1 στο νευρώνα i του στρώματος K
- $a_i^{(K)}$ — Ενεργοποίηση νευρώνα i του στρώματος K

Εικόνα 2-7: Λειτουργεία Νευρωνικού Δικτύου

2.4.1.2 Εκπαίδευση Τεχνητών Νευρωνικών Δικτύων

Η δημιουργία τεχνητών νευρωνικών δικτύων είναι πλέον τετριμμένη και σχετικά απλή, αλλά ο τρόπος με τον οποίο εκπαιδεύονται έχει μεγάλο ενδιαφέρον και εξαιρετική σημασία στην λειτουργία τους. Η εκπαίδευση των νευρωνικών γίνεται με σωστή αναπροσαρμογή των βαρών των διασυνδέσεων τους. Ο αλγόριθμος που χρησιμοποιείται περισσότερο για την εκπαίδευση είναι αυτός της οπίσθιας διάδοσης (backpropagation) και εκτελείται αφού γίνει πρόβλεψη από το νευρωνικό δίκτυο, την οποία συγκρίνουμε με την επιθυμητή τιμή. Η λογική του αλγορίθμου βασίζεται στην ύπαρξη δύο ειδών σημάτων στο νευρωνικό δίκτυο, το λειτουργικό σήμα που είναι οι είσοδοι και έξοδοι των νευρώνων όπως εξηγήσαμε πιο πάνω και το σήμα σφάλματος. Το σήμα σφάλματος έχει αντίθετη φορά από το λειτουργικό σήμα δηλαδή από το τέλος του δικτύου προς την αρχή και σκοπός του είναι να μεταφέρει το σφάλμα κάθε νευρώνα έτσι ώστε να πραγματοποιηθεί η εκπαίδευση του. Το σφάλμα υπολογίζεται με την βοήθεια της συνάρτησης απώλειας (loss function) η οποία υπολογίζει την διαφορά της εξόδου των νευρώνων με την επιθυμητή, και μηδενίζεται όταν δεν υπάρχουν σφάλματα [19].

$$L(\hat{y}, y) = L(f(X, W), y) \tag{17}$$

Όπου L είναι η συνάρτηση απώλειας, $\hat{y}$ και y είναι αντίστοιχα η προβλεπόμενη και επιθυμητή έξοδος.

Αφού υπολογιστεί το σφάλμα σε κάποιο νευρώνα τότε ξεκινάμε να προσαρμόζουμε τα βάρη των σημάτων εισόδου ανάλογα με την συνεισφορά που είχαν στο σφάλμα. Για τον καθορισμό της συνεισφοράς αυτής χρησιμοποιούμε την παράγωγο του μέσου τετραγωνικού σφάλματος ως προς το συγκεκριμένο βάρος (gradient descent). Η μέθοδος που αναφέραμε αποτελεί τον βασικό τρόπο λειτουργείας της οπίσθιας μετάδοση και υπάρχουν πολλές παραλλαγές αυτής, με διάφορες συναρτήσεις ενεργοποίησης και συναρτήσεις απώλειας.

2.4.2 Deep Q-Networks (DQN)

Η χρήση τεχνητών νευρωνικών δικτυών για ενισχυτική μάθηση αντιμετωπίζει αρκετές δυσκολίες. Ο κύριος λόγος είναι ο τρόπος λειτουργίας της ίδια της ενισχυτικής μάθησης καθώς πολλές φορές έχουμε σποραδικές ανταμοιβές οι οποίες μπορεί να είναι θορυβώδης και καθυστερημένες. Επίσης ενώ έχουμε μια ακολουθία από ισχυρά συσχετισμένες καταστάσεις η βαθιά ενισχυτική μάθηση υποθέτει ότι τα στοιχεία της ακολουθίας είναι ανεξάρτητα και ομοιόμορφα κατανεμημένα κάτι που δεν ισχύει γιατί καθώς ο πράκτορας μαθαίνει πραγματοποιεί και νέες συμπεριφορές [18].

Ένας τρόπος όμως με τον οποίο μπορούμε να εφαρμόσουμε ενισχυτική μάθηση μαζί με τεχνητά νευρωνικά δίκτυα είναι οι αλγόριθμοι Deep Q-Netowrks (DQN). Οι αλγόριθμοι αυτοί βασίζονται στην τροποποίηση του κλασικού αλγορίθμου Q-learning έτσι ώστε να μπορεί να λειτουργά μαζί με ένα τεχνητό νευρωνικό δίκτυο το οποίο εκπαιδεύεται για να προσεγγίσει την βέλτιστη συνάρτηση τιμής-ενέργειας Q:

$$Q(s, a, w) \approx Q^*(s, a) \tag{18}$$

Η παραλλαγή στο Q-learning γίνεται με τη πρόσθεση της παραμέτρου των βαρών (w ή θ) του νευρωνικού στις εξισώσεις της. Επίσης το νευρωνικό μας δίκτυο το αποκαλούμε Q-δίκτυο και πρέπει να ορίσουμε ανάλογα και σε αυτό την συνάρτηση απώλειας και τις εξόδους των νευρώνων αντίστοιχα με τον εξής τρόπο:

$$L_i(w_i) = E_\pi[(y_i - Q(s, a, w))^2] \tag{19}$$

$$y_i = E[R + \gamma \, max_a Q(s', a', w_{i-1}) | s, a \,] \tag{20}$$

Ο αλγόριθμος DQN χαρακτηρίζεται ως model-free και off-policy καθώς δεν χρειάζεται πλήρη εικόνα για το περιβάλλον του πάρα μόνο εμπειρίες από αυτό και χρησιμοποιεί διαφορετική πολιτική για την εξερεύνηση και προσαρμογή της συνάρτησης τιμής-ενέργειας. Επίσης καλό είναι να αναφέρουμε ότι στον συγκεκριμένο αλγόριθμο γίνεται χρήση της μεθόδου αναπαραγωγής εμπειριών (experience replay). Δηλαδή ο πράκτορας χρησιμοποιεί δεδομένα από προηγούμενες του εμπειρίες για την ανανέωση των βαρών του Q-δικτύου, και σκοπός είναι να αντιμετωπιστεί η συσχέτιση μεταξύ διαδοχικών δειγμάτων που λαμβάνει.

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

Εικόνα 2-8: Αλγόριθμος DQN

2.5 *Εφαρμογή Ενισχυτικής Μάθησης για λήψη αποφάσεων*

Όπως έχουμε αναφέρει ήδη τα τελευταία χρόνια στην Ενισχυτική Μάθηση έχει παρατηρηθεί τεράστια ανάπτυξη, ειδικά στην Βαθιά Ενισχυτική Μάθηση με αποτέλεσμα να δημιουργούνται καινοτόμα μοντέλα και τρόποι εκπαίδευσης. Λόγο αυτής της ανάπτυξης ξεκίνησε η χρήση της ενισχυτικής μάθησης για προβλήματα διαδοχικής λήψης αποφάσεων σε ένα μεγάλο εύρος πεδίων όπως φυσικές και κοινωνικές επιστήμες και τη μηχανική. Η χρήση της ενισχυτικής μάθησης για λήψη αποφάσεων χρησιμοποιώντας κυρίως MDPs απορρέει από την ίδια την ενισχυτική μάθηση, η οποία βασίζεται στην εκπαίδευση του πράκτορα στο να παίρνει αποφάσεις για το ποιες ενέργειες να εκτελέσει. Μερικά πεδία στα οποία χρησιμοποιείται ήδη η ενισχυτική μάθηση είναι:

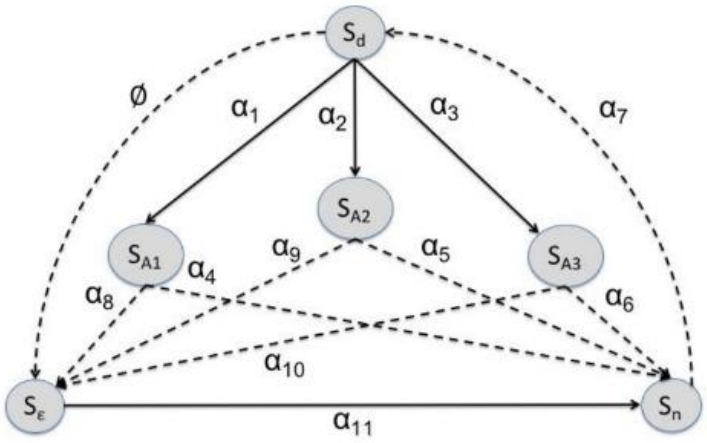
- Παιχνίδια
- Ρομποτική
- Επεξεργασία φυσικής γλώσσας
- Όραση υπολογιστών
- Διοίκηση Επιχειρήσεων
- Οικονομικά
- Υγεία
- Εκπαίδευση
- Βιομηχανία
- Ευφυή συστήματα Μέσων Μεταφοράς
- Αθλήματα

Όσων αφορά την λήψη αποφάσεων σε πραγματικά σενάρια μεγάλο ενδιαφέρον έχουν οι εφαρμογές στην διοίκηση επιχειρήσεων και στην βιομηχανία. Για την διοίκηση επιχειρήσεων η χρήση της ενισχυτικής μάθησης έγινε στην σύσταση διαφημίσεων, στην διαχείριση πελατών και γενικά στο μάρκετινγκ. Παραδείγματα είναι η χρήση προσωποποιημένων συστάσεων για άρθρα, βάσει του περιεχόμενου τους και στοιχεία των χρηστών. Στον τομέα της βιομηχανίας η χρήση της ενισχυτικής μάθησης έγινε στα πλαίσια του όρου Βιομηχανία 4.0 (Industry 4.0) που χαρακτηρίζεται ως η 4η βιομηχανική επανάσταση. Σκοπός είναι ο ψηφιακός μετασχηματισμός της παραγωγής και των συναφών βιομηχανιών με την τεχνητή νοημοσύνη γενικά και τη ενισχυτική μάθηση ιδιαίτερα να είναι ο τρόπος με τον οποίο θα επιτευχθεί. Συγκεκριμένα έχει γίνει χρήση ενισχυτικής μάθησης για την συνεργασία ρομπότ και ανθρώπων για την γραμμή παραγωγής και πολλές προσπάθειες για να γεφυρωθεί το κενό μεταξύ της ακαδημαϊκής έρευνας και των πραγματικών προβλημάτων στη βιομηχανία [1].

3 Προτεινόμενη Προσέγγιση

3.1 Συνολική παρουσίαση μεθόδου

Η προτεινόμενη μέθοδος μοντελοποίησης της διαδικασίας λήψης αποφάσεων με τη βοήθεια της ενισχυτικής μάθησης που προτείνουμε έχει ως κύριο σκοπό την προληπτική σύσταση αποφάσεων για την αποφυγή ενός ανεπιθύμητου συμβάντος, με την ιδιότητα να μπορεί να προσαρμόζεται δυναμικά ανάλογα με το συγκεκριμένο πρόβλημα για το οποίο χρησιμοποιείται. Η μέθοδος μας βασίζεται στην εκπαίδευση ενός πράκτορα ενισχυτικής μάθησης ο οποίος έχει ως στόχο να μεταβεί σε μία επιθυμητή κατάσταση ξεκινώντας από μια κατάσταση κινδύνου, λαμβάνοντας κάποιες αποφάσεις και αποφεύγοντας το ανεπιθύμητο συμβάν. Οι αποφάσεις αυτές διαφέρουν ανάλογα με το πρόβλημα που ζητείται να επιλύσει ο πράκτορας και να βρει την βέλτιστη απόφαση. Η βέλτιστη αυτή απόφαση επιτυγχάνεται όταν ο πράκτορας επιλέγει την απόφαση η οποία έχει το μικρότερο κόστος. Το κόστος αυτό είναι ένας τρόπος για να μπορούμε να συγκρίνουμε τις αποφάσεις μεταξύ τους το οποίο θα αναλύσουμε αργότερα σε αυτό το κεφάλαιο. Επίσης, ο πράκτορας έχει την δυνατότητα να μην πάρει καμία απόφαση με αποτέλεσμα να συμβεί το ανεπιθύμητο συμβάν. Για τη μετάβαση από το ανεπιθύμητο συμβάν στην επιθυμητή κατάσταση επίσης υπάρχει κάποιο κόστος.



Εικόνα 3-1: Γράφος μοντέλου με δυνατότητα επιλογής τριών διαφορετικών αποφάσεων.

Πίνακας 3-1: Παράρτημα Εικόνας 3-1

Sn	Επιθυμητή κατάσταση
Sd	Κατάσταση κινδύνου
Se	Ανεπιθύμητο συμβάν
SAi	Κατάσταση επιλογής Απόφασης A_i
Συνεχόμενες Ακμές	Μετάβαση με κόστος
Διακεκομμένες Ακμές	Μετάβαση χωρίς κόστος

3.2 Μοντελοποίηση προβλήματος απόφασης

Για να μοντελοποιήσουμε το πρόβλημα απόφασης του μοντέλου μας χρησιμοποιήσαμε την διαδικασία Markov Decision Process (MDP) για πεπερασμένες καταστάσεις και ενέργειες. Αφού θα χρησιμοποιήσουμε ενισχυτική μάθηση για την λήψη των αποφάσεων μας θα αναφερόμαστε στο πρόβλημα που θέλουμε να λύσουμε ως περιβάλλον, στις αποφάσεις ως ενέργειες και στο κόστος λήψης μιας απόφασης ως ανταμοιβή. Όπως αναφέραμε και στο Κεφάλαιο 2 οι MDPs είναι ένας τυποποιημένος τρόπος με τον οποίο μπορούμε να αναπαραστήσουμε πλήρως τις καταστάσεις, τις ενέργειες και τις σχέσεις μεταξύ τους, όπως αυτές ορίζονται από το περιβάλλον που θέλουμε να κατασκευάσουμε.

3.2.1 Κατασκευή MDP

Αρχικά ξεκινάμε με το να ορίσουμε το σύνολο των καταστάσεων μας (states). Στο δικό μας περιβάλλον έχουμε 3 σταθερές καταστάσεις και ν καταστάσεις λήψης αποφάσεων όπου ν είναι το πλήθος των διαθέσιμων αποφάσεων.

$$S = \{ S_n, S_d, S_e, S_{A1}, \dots , S_{Av} \}$$

S_n = επιθυμητή κατάσταση

S_d = κατάσταση κινδύνου

S_e = κατάσταση ανεπιθύμητου συμβάν

S_{Ai} = κατάσταση επιλογής απόφασης A_i

Ακολούθως, ορίζουμε το σύνολο των ενεργειών (actions). Λόγο των τριών σταθερών καταστάσεων που ορίσαμε θα έχουμε και εδώ 3 σταθερές ενέργειες α_d, α_e και α_n οι οποίες είναι η μετάβαση στις καταστάσεις S_d, S_e και S_n αντίστοιχα. Επιπρόσθετα, για ν διαθέσιμες αποφάσεις θα έχουμε 3ν ενέργειες. Για κάθε απόφαση A_i θα έχουμε μια ενέργεια α_i με την οποία θα μεταβαίνουμε από την κατάσταση κινδύνου στην κατάσταση λήψης απόφασης S_{ai} και από αυτήν 2 ενέργειες α_{ie} και α_{in} με τις οποίες θα μεταβαίνουμε στις καταστάσεις S_e και S_n αντίστοιχα. Έτσι το σύνολο ενεργειών έχει πλήθος 3+3ν ενεργειών με ν διαθέσιμες αποφάσεις.

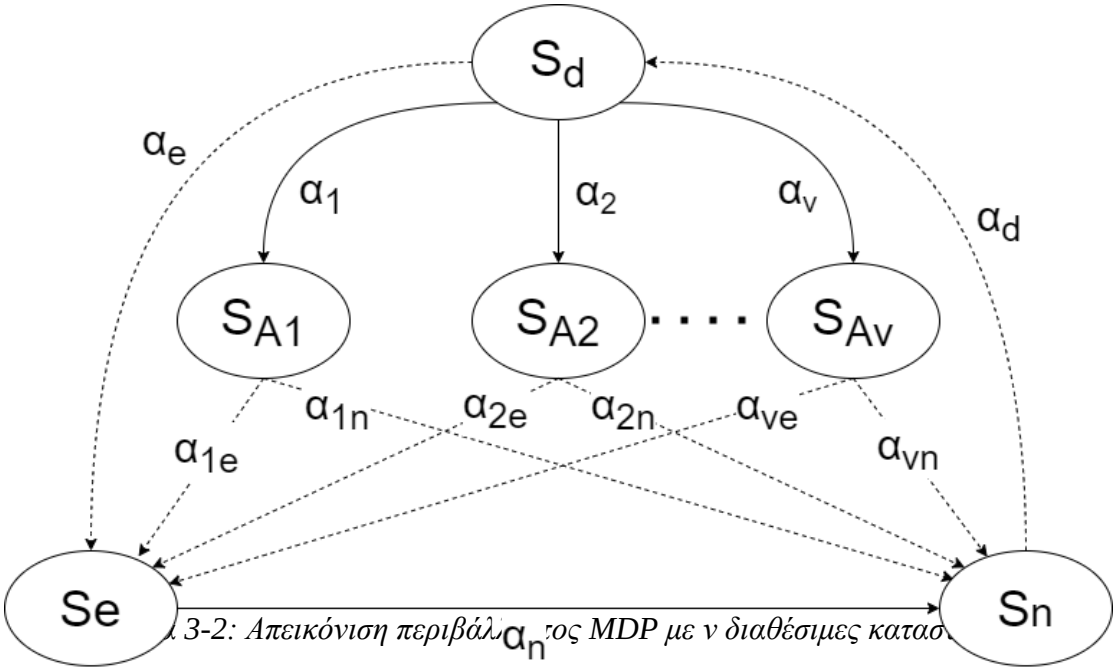
$$A = \{ a_d, a_e, a_n, a_1, a_{1e}, a_{1n}, \dots a_v, a_v, a_{vn}\}$$

Ακόμη, χρειάζεται να ορίσουμε τις ανταμοιβές κάθε ενέργειας που υπάρχει στο σύνολο ενεργειών που ορίσαμε πριν.

$$R = \{ R_d, R_e, R_n, R_1, R_{1e}, R_{1n}, \dots R_v, R_v, R_{vn}\}$$

Έχοντας ορίσει τα πιο πάνω σύνολα S, A, και R είμαστε σε θέση να ορίσουμε και τον πίνακα μεταβατικών καταστάσεων πιθανότητας P και να εκφράσουμε το περιβάλλον μας με μια MDP. Τέλος αφού η μοντελοποίηση μας πλήρη όλες τις προϋποθέσεις και ιδιότητες μιας MDP μπορούμε να αναπαραστήσουμε το πρόβλημα μας λήψης αποφάσεων με μια MDP.

$$MDP = \{S, A, P, R\}$$



Εικόνα 3-2: Απεικόνιση περιβάλλοντος MDP με ν διαθέσιμες καταστ

Το πιο πάνω διάγραμμα θα μπορούσε να χρησιμοποιηθεί για το παράδειγμα που δώσαμε στο Κεφάλαιο 2 με το καροτσάκι. Στο συγκεκριμένο παράδειγμα κάθε μια από τις ενδιάμεσες καταστάσεις θα ήταν η κίνηση η οποία θα εκτελούσαμε, να μετακινηθούμε δεξιά, αριστερά ή να μην κινηθούμε. Ομοίως το ανεπιθύμητο συμβάν θα ήταν να απομακρυνθούμε από το τελικό σημείο ενώ η τελική κατάσταση θα ήταν να φτάσουμε σε αυτό.

3.2.2 Δημιουργία Περιβάλλοντος Ενισχυτικής Μάθησης

Λόγο του ότι θα χρησιμοποιήσουμε ενισχυτική μάθηση για το μοντέλο μας χρειάζεται να κατασκευάσουμε το περιβάλλον στο οποίο θα δρα ο πράκτορας μας και το οποίο θα βασίζεται στην MDP που ορίσαμε πριν. Η MDP και το περιβάλλον μας όμως δεν είναι σταθερά, και αλλάζουν ανάλογος με το πρόβλημα ή καλύτερα το σενάριο για το οποίο καλείτε το μοντέλο μας να πάρει αποφάσεις. Με τον τρόπο με τον οποίο ορίσαμε την MDP η κατασκευή του περιβάλλοντος μας εξαρτάται από το πλήθος των ενεργειών που μπορούν να εκτελεστούν, βάση των οποίων θα κατασκευαστούν οι καταστάσεις, οι ενέργειες και οι ανταμοιβές του περιβάλλοντος μας. Παραστατικά μπορούμε να φανταστούμε το περιβάλλον μας σαν το ταμπλό ενός επιτραπέζιου παιχνιδιού. Για παράδειγμα μπορούμε να πάρουμε το σκάκι ως το περιβάλλον μας και τον πράκτορα μας σαν αυτόν που παίζει το παιχνίδι.

3.2.2.1 Καταστάσεις

Οι καταστάσεις του περιβάλλοντος μας είναι οι θέσεις στις οποίες μπορούν να κινηθούν τα πιόνια μας. Στην δική μας περίπτωση φυσικά οι ‘’θέσεις’’ αυτές δεν είναι σταθερές όπως το σκάκι, αλλά έχουν αριθμό ν+3 για ν διαθέσιμες αποφάσεις. Ο πράκτορας μας λαμβάνει γνώση της κατάστασης του μέσο των παρατηρήσεων που στο δικό μας περιβάλλον οι παρατηρήσεις συμπίπτουν με τις καταστάσεις και χαρακτηρίζονται με ακεραίους αριθμούς με τον εξής τρόπο:

Κατάσταση	**Αριθμός παρατήρησης**
Sn	0
Sd	1
SAi	i+1
Se	-1

Ακόμη οι πιο πάνω τιμές των καταστάσεων μας είναι και οι τιμές που βλέπει ο πράκτορας μας στην παρατήρηση του. Δηλαδή όταν εκτελεστεί κάποια ενέργεια και ο πράκτορας μας κινηθεί στην κατάσταση Se σαν παρατήρηση ως προς το περιβάλλον θα βλέπει την τιμή -1.

3.2.2.2 Ενέργειες

Στο σκάκι οι ενέργειες είναι οι κινήσεις που μπορεί να εκτελέσει ο παίκτης με τα πιόνια του. Στην δική μας περίπτωση μπορούμε να θεωρήσουμε ότι έχουμε ένα μόνο πιόνι το οποίο με συγκεκριμένες κινήσεις μπορεί να κινηθεί στις καταστάσεις μας. Οι κινήσεις αυτές ορίζονται από τον σύνολο ενεργειών της MDP που κατασκευάσαμε. Επίσης στο σκάκι οι ενέργειες που μπορεί να εκτελέσει ένα πιόνι ορίζονται από το είδος του πιονιού ενώ στο δικό μας περιβάλλον οι ενέργειες ορίζονται από την κατάσταση στην οποία βρισκόμαστε.

3.2.2.3 Ανταμοιβές

Στο παράδειγμα με το σκάκι δεν υπάρχει ξεκάθαρη αντιστοίχιση με την άμεση ανταμοιβή καθώς δεν υπάρχει κάποιο κόστος ή κέρδος με την εκτέλεση μιας κίνησης. Στο δικό μας περιβάλλον όμως κάποιες ενέργειες έχουν άμεση ανταμοιβή για τον πράκτορα μας. Οι ενέργειες που έχουν άμεση ανταμοιβή είναι αυτές που μας οδηγούν στην λήψη μιας απόφασης και η μετάβαση από το ανεπιθύμητο συμβάν στην επιθυμητή κατάσταση. Οι ανταμοιβές αυτές ορίζονται αριθμητικά και μπορούν να είναι θετικές ή αρνητικές. Σύμφωνα με τον τρόπο που κατασκευάσαμε το περιβάλλον μας οι ανταμοιβές πρέπει να είναι θετικές όταν το επιθυμητό είναι η ελάχιστη από αυτές και αρνητικές όταν το επιθυμητό είναι η μέγιστη από αυτές. Δηλαδή το μοντέλο μας θεώρει ως καλύτερη απόφαση αυτή με την μεγαλύτερη ανταμοιβή.

3.3 Μοντέλο επίλυσης με βαθιά ενισχυτική μάθηση – DeepRL αλγόριθμοι

Το περιβάλλον μας είναι κατασκευασμένο στο να μπορούν να τρέξουν πολλοί αλγόριθμοι βαθιάς ενισχυτικής μάθησης και όχι κάποιος συγκεκριμένα. Αν και υπάρχουν πάρα πολλοί διαφορετικοί αλγόριθμοι για βαθιά ενισχυτική μάθηση στην δική μας υλοποίηση χρησιμοποιήσαμε τον αλγόριθμο DQN μαζί με ένα νευρωνικό δίκτυο 3 κρυφών επιπέδων για την εκπαίδευση του πράκτορα μας. Η επιλογή του DQN έγινε για τρείς βασικού λόγους. Καταρχάς η ευκολία χρήσης του και η σχετική του απλότητα ήταν βοηθητικά έτσι ώστε να κατασκευαστεί πρώτα απ' όλα σωστά και γρήγορα. Οι δύο άλλοι λόγοι ήταν το ότι έχουμε πεπερασμένο αριθμό καταστάσεων και πολλά επεισόδια κατά την διάρκεια της εκπαίδευσης, κάτι το οποίο ταιριάζει με τον τρόπο λειτουργείας του DQN.

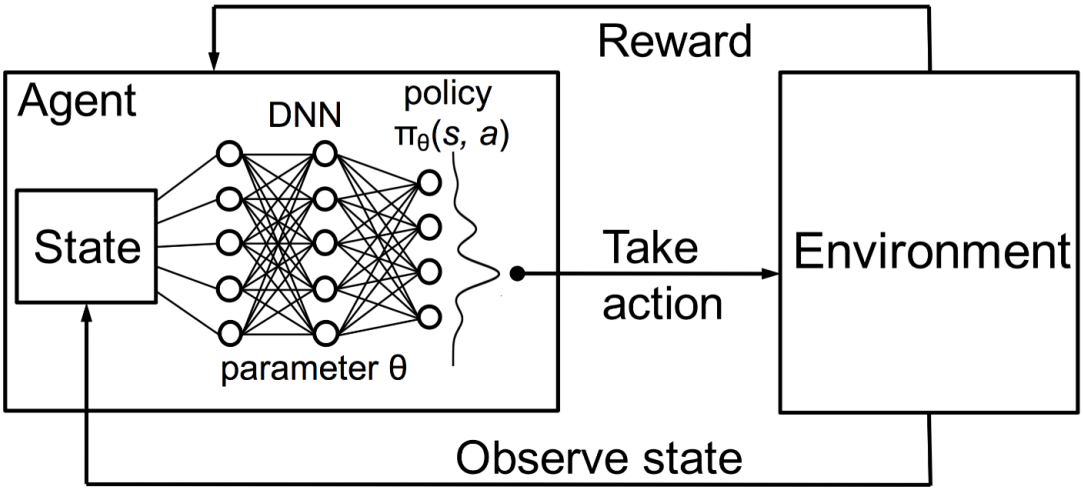
Ο DQN μαζί με το νευρωνικό δίκτυο έχουν ως σκοπό να προσεγγίσουν την συνάρτηση Q η οποία αξιολογεί τις διαθέσιμες ενέργειες ανάλογα με την κατάσταση που βρίσκεται εκείνη

την στιγμή ο πράκτορας. Ακόμη χρησιμοποιήσαμε την τεχνική αναπαραγωγής εμπειριών, όπου ο πράκτορας κρατά στην μνήμη του παρελθοντικές καταστάσεις και ενέργειες τις οποίες λαμβάνει υπόψη για την εκπαίδευση του.

Επιπρόσθετα μεγάλη βαρύτητα για την εκπαίδευση του πράκτορα φέρει η επιλογή της πολιτικής βάση της οποίας ο πράκτορας θα διαλέγει ποια ενέργεια θα εκτελέσει. Υπάρχουν πολλές διαφορετικές πολιτικές οι οποίες σκοπό έχουν να εξισορροπήσουν την «εκμετάλλευση» και «εξερεύνηση» του πράκτορα ανάλογα με το περιβάλλον στο οποίο αυτός δρα. Πιο συνηθισμένη και απλή είναι η ε-άπληστη τεχνική, όμως εμείς χρησιμοποιήσαμε τρείς διαφορετικές πολιτικές στα πειράματα μας.

3.3.1 Λήψη αποφάσεων με βαθιά ενισχυτική μάθηση

Μια πιο αναλυτική λειτουργία του μοντέλου που κατασκευάσαμε φαίνεται στην πιο κάτω εικόνα όπου μπορούμε να δούμε καλύτερα το πως αλληλοεπιδρά ο πράκτορας με το περιβάλλον μας. Το περιβάλλον αντιπροσωπεύει το πρόβλημα απόφασης και ο πράκτορας τον αλγόριθμο βαθιάς ενισχυτικής μάθησης.



Εικόνα 3-3: Αναπαράσταση των επιμέρους στοιχείων του Μοντέλου [20]

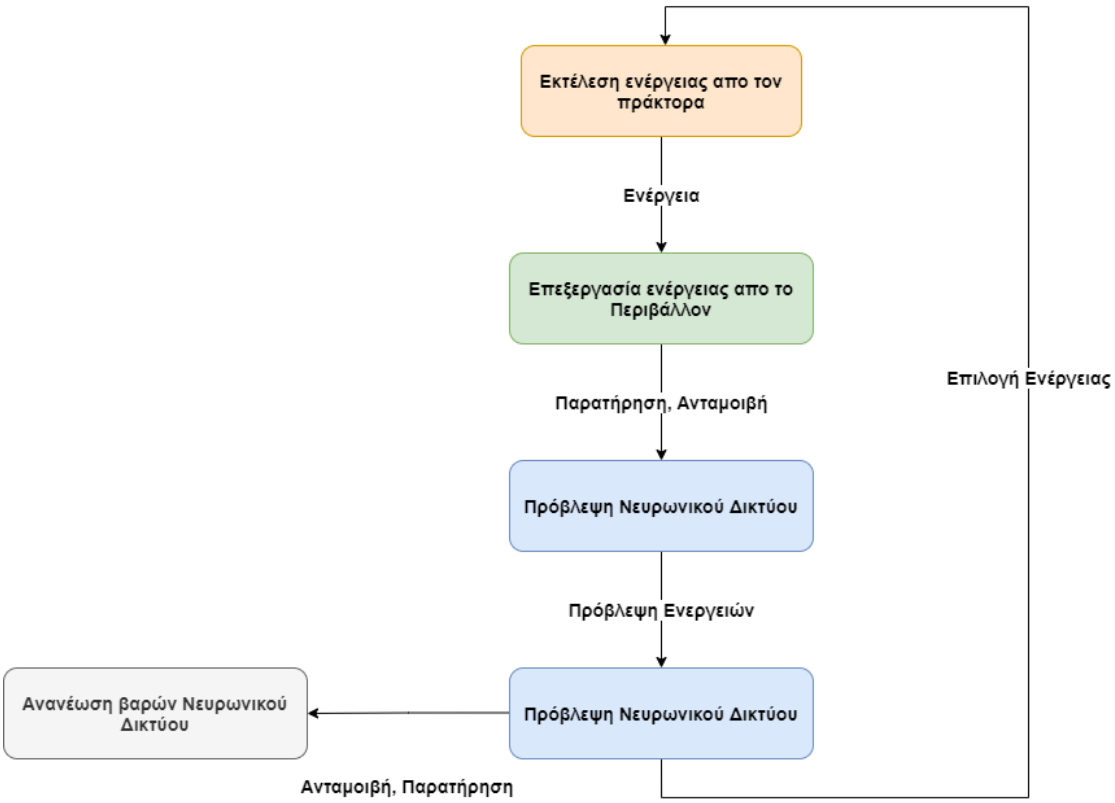
Συγκεκριμένα το περιβάλλον αναπαριστά το πρόβλημα που θέλουμε να επιλύσουμε περιλαμβάνοντας τα χαρακτηριστικά του και την λειτουργεία του στέλνοντας τις απαραίτητες πληροφορίες στον πράκτορα. Ο πράκτορας με την σειρά του αποτελείτε από το τεχνητό νευρωνικό δίκτυο και την πολιτική του. Το πρώτο είναι υπεύθυνο για την πρόβλεψη της

καλύτερης ενέργειας που μπορεί να εκτελεστεί, ενώ η πολιτική αποφασίζει τελικά ποια ενέργει θα εκτελεστεί και να σταλθεί στο περιβάλλον.

3.3.2 Συνολική Λειτουργία του μοντέλου με βαθιά ενισχυτική μάθηση

Η συνολική λειτουργεία του μοντέλου μας γίνεται με τα εξής επαναλαμβανόμενα βήματα:

1. Ο πράκτορας δίνει στο περιβάλλον την ενέργεια που θέλει να εκτελέσει
2. Το περιβάλλον επεξεργάζεται την ενέργεια του πράκτορα και αναλόγως με την κατάσταση με την οποία βρίσκεται του απαντά με την παρατήρηση που θα έχει ο πράκτορας και την ανταμοιβή που θα πάρει αφού εκτελεστεί η ενέργεια του.
3. Ο πράκτορας μας οδηγεί την παρατήρηση που παίρνει ως είσοδο στο νευρωνικό του δίκτυο το οποίο είναι υπεύθυνο να αποδώσει τιμές στις επόμενες πιθανές κινήσεις που μπορεί να εκτελέσει ο πράκτορας. Η καλύτερη ενέργεια που προτείνει το νευρωνικό δίκτυο έχει και την μεγαλύτερη τιμή από τις υπόλοιπες.
4. Ο πράκτορας παίρνει την έξοδο του νευρωνικού δικτύου, τις τιμές των ενεργειών δηλαδή, και βάσει τις πολιτικής που χρησιμοποιεί επιλέγει πια ενέργεια τελικά θα εκτελέσει.
5. Χρησιμοποιώντας την ανταμοιβή που πήρε από το περιβάλλον στην αρχή, λαμβάνοντας υπόψη σε ποια κατάσταση ήταν και τι ανταμοιβή περίμενε, αναπροσαρμόζει τα βάρη του νευρωνικού. Δηλαδή σε αυτή την φάση γίνεται ουσιαστικά η εκμάθηση.
6. Τέλος ο πράκτορας δίνει στο περιβάλλον την ενέργεια, που πήρε απόφαση να εκτελέσει και ξεκινά ξανά η πιο πάνω διαδικασία.



Εικόνα 3-4: Συνολική Λειτουργεία Μοντέλου

Σαν τελικό μοντέλο μετά την εκπαίδευση του πράκτορα μας μπορεί να χρησιμοποιηθεί μόνο το νευρωνικό δίκτυο. Δίνοντας του σαν παρατήρηση την κατάσταση κινδύνου S_d θα βγάζει την μεγαλύτερη τιμή στην έξοδο του, στην ενέργεια με το λιγότερο κόστος που εξερεύνησε με πιθανότητα να είναι και η βέλτιστη, αναλόγως με το αν την έχει εξερευνήσει ή όχι. Επιπρόσθετα έχουμε την δυνατότητα κατά την διάρκεια της εκπαίδευσης να αποθηκεύσουμε τα βάρη των ακμών του νευρωνικού δικτύου με τα οποία αυτό έκανε πρόβλεψη για την καλύτερη απόφαση μέχρι την συγκεκριμένη στιγμή. Έτσι μετά το πέρας της εκπαίδευσης μπορούμε να επαναφέρουμε τα συγκεκριμένα βάρη στο νευρωνικό μας δίκτυο και έτσι να μας προβλέπει την καλύτερη απόφαση που βρήκε κατά την εκπαίδευση.

3.4 Παραδείγματα μοντελοποίησης

Όπως αναφέραμε και στο κεφάλαιο 2 η ενισχυτική μάθηση χρησιμοποιείται όλο και περισσότερο ειδικά στον κλάδο των επιχειρήσεων για προβλέψεις, βελτιστοποιήσεις λήψη αποφάσεων κ.α. Σε αυτή την ενότητα λοιπόν θα παραθέσουμε μερικά παραδείγματα χρήσης του μοντέλου μας σε πραγματικά σενάρια. Βασική προϋπόθεση για να μπορέσει να χρησιμοποιηθεί το μοντέλο μας είναι η ύπαρξη αντιστοίχισης μεταξύ των αποφάσεων και του κόστους τους. Το κόστος αυτό μπορεί να είναι οτιδήποτε, φτάνει να μπορούμε να συγκρίνουμε με αυτό τις αποφάσεις που μπορούν να παρθούν. Δηλαδή το κόστος μπορεί να είναι χρηματικό,

κόστος σε χρόνο κ.ο.κ. ή μπορεί να είναι συνδυαστικό λαμβάνοντας υπόψη διάφορες παραμέτρους αρκεί στο τέλος να έχουμε μια συγκρίσιμη τιμή για κάθε απόφαση. Επιπρόσθετα το κόστος αυτό μπορεί να παίρνει θετικές τιμές όταν θέλουμε να το μεγιστοποιήσουμε ή αρνητικό όταν θέλουμε να το ελαχιστοποιήσουμε.

Για τα παραδείγματα που θα παραθέσουμε πιο κάτω η λειτουργία του μοντέλου μας θα βασίζεται σε όσα αναφέραμε στο προηγούμενο υποκεφάλαιο. Δηλαδή το περιβάλλον μας θα κατασκευάζεται με τρόπο ώστε να προσομοιώνει το πρόβλημα το οποίο καλείτε να επιλύσει το μοντέλο μας. Η προσομοίωση αυτή γίνεται φτιάχνοντας τόσες καταστάσεις όσες είναι και οι διαθέσιμες αποφάσεις που μπορούν να παρθούν. Ο πράκτορας με την σειρά του θα είναι υπεύθυνος στο να εξερευνήσει το συγκεκριμένο περιβάλλον και να καταφέρει να μας προτείνει την βέλτιστη απόφαση που πρέπει να παρθεί, έτσι ώστε να ελαχιστοποιηθεί το κόστος της ενέργειας που θα εκτελεστεί.

3.4.1 Παράδειγμα 1 – Χρήση για Industry 4.0

Όπως αναφέραμε και στο Κεφάλαιο 2 η χρήση της ενισχυτικής μάθησης χρησιμοποιείται στην βιομηχανία για την επίτευξη του Industry 4.0. Ας δώσουμε ένα παράδειγμα χρήσης του δικού μας μοντέλου σε μια γραμμή παραγωγής σε μια βιομηχανία. Σκοπός του μοντέλου μας θα είναι η αυτόματη λήψη αποφάσεων και η αντίστοιχη εκτέλεση ενεργειών για την σωστή και αποδοτική λειτουργεία ενός εργοστασίου. Ας υποθέσουμε αρχικά ότι το μοντέλο μας είναι υπεύθυνο για την ομαλή λειτουργεία της γραμμή παραγωγής σε ένα εργοστάσιο. Πιο συγκεκριμένα για θέμα απλοϊκότητας, ότι είναι υπεύθυνο για την λειτουργία ενός λέβητα στο εργοστάσιο. Έστω λοιπόν ότι ανιχνεύεται κάποιο πρόβλημα στον λέβητα (κατάσταση κινδύνου) και πρέπει να παρθεί μια απόφαση και να εκτελεστεί μια ενέργεια για την αποφυγή έκρηξης ή καλύτερα της ολικής καταστροφής του (ανεπιθύμητο συμβάν).

Ας ορίσουμε 3 πιθανές ενέργειες που μπορούν να εκτελεστούν:

1. Να θέσουμε προσωρινά εκτός λειτουργείας τον λέβητα.
2. Να αλλάξουμε ένα εξάρτημα που πιθανών να προκαλεί το πρόβλημα.
3. Να αντικαταστήσουμε τον λέβητα με καινούριο.
4. Να μην παρθεί απόφαση.

Το σημαντικό σε κάθε πρόβλημα λήψης απόφασης είναι ο σωστός ορισμός του κόστους κάθε απόφασης. Έτσι στη συνέχεια θα δώσουμε δύο τρόπους για το πώς μπορεί να μοντελοποιηθεί το κόστος αυτό και θα αντιστοιχήσουμε τις πιο πάνω ενέργειες με τα κόστη τους σε μορφή πίνακα. Τα κόστη αυτά είναι ενδεικτικά και σκοπός τους είναι να δείξουν πως επηρεάζεται η λήψη της βέλτιστης απόφασης από αυτά.

Σε πρώτη φάση ας υποθέσουμε ότι το κόστος είναι μόνο χρηματικό, χωρίς να μας ενδιαφέρει το πως θα επηρεαστεί το εργοστάσιο.

Πίνακας 3-2: Κόστη παραδείγματος 1α

Ενέργεια	Κόστος
1	0
2	100
3	2000
4	2000

Σύμφωνα με τον πιο πάνω πίνακα η βέλτιστη λύση επιτυγχάνεται με το να εκτελεστεί η πρώτη ενέργεια η οποία έχει μηδενικό κόστος στην περίπτωση μας.

Μοντελοποιώντας διαφορετικά το κόστος κάθε ενέργειας ας υποθέσουμε ότι ως κόστος ενέργειας λαμβάνουμε υπόψη και το χρηματικό κόστος όταν σταματά η γραμμή παραγωγής του εργοστασίου.

Πίνακας 3-3: Κόστη παραδείγματος 1β

Ενέργεια	Κόστος
1	1000
2	400
3	4000
4	5000

Με τα νέα κόστη των ενεργειών μας δεν συμφέρει η πρώτη απόφαση, καθώς το κόστος στην παραγωγή όταν είναι κλειστός ο λέβητας είναι μεγαλύτερο από το να πάρουμε την δεύτερη απόφαση και να αλλάξουμε το εξάρτημα.

Αφού έχουμε δείξει πώς επηρεάζει η μοντελοποίηση του κόστους των ενεργειών του μοντέλου μας μπορούμε να περιγράψουμε πιο περίπλοκα σενάρια εφαρμογής του. Ο τρόπος κατασκευής του μοντέλου μας είναι τέτοιος ώστε να μπορεί να προσαρμοστεί σε πολλά και διαφορετικά μεταξύ τους σενάρια προβλημάτων. Τα σενάρια αυτά μπορεί να είναι αρκετά πιο πολύπλοκα από το παράδειγμα του λέβητα που περιγράψαμε. Δηλαδή το μοντέλο μας μπορεί

να είναι υπεύθυνο ακόμα και για όλη την γραμμή παραγωγής. Αυτό σημαίνει ότι η μοντελοποίηση του προβλήματος θα έχει πολύ περισσότερες ενέργειες. Οι ενέργειες αυτές μπορεί να αποτελούνται από διάφορες αποφάσεις που μπορούν να παρθούν. Δηλαδή μια ενέργεια μπορεί να αποτελείτε από ένα σύνολο αποφάσεων όπως το να αλλαχτεί κάποιο εξάρτημα στον λέβητα, να γίνει παραγγελία για πρώτες ύλες και μετακίνηση εργατών σε άλλο σημείο της γραμμής παραγωγής. Όπως καταλαβαίνουμε σκοπός αυτών των ενεργειών είναι η ομαλή και αποδοτική λειτουργία του εργοστασίου κάτι το οποίο αντιλαμβάνεται ο πράκτοράς μας σαν κόστος και προσπαθεί να ελαχιστοποιήσει. Όπως αναφέραμε και στην αρχή της παραγράφου το κόστος αυτό μπορεί να είναι συνάρτηση πολλών παραμέτρων και ακόμα να αλλάζει δυναμικά. Επίσης για τον υπολογισμό του κόστους μπορεί να λαμβάνεται υπόψη και ο χρόνος ο οποίος θα ληφθεί, με αποτέλεσμα ο πράκτοράς να μπορεί να πάρει τις ανάλογες αποφάσεις την κατάλληλη χρονική στιγμή. Λόγω της πολυπλοκότητας και του πλήθους των δεδομένων που πρέπει να ληφθούν υπόψη για μεγάλης κλίμακας προβλήματα η χρήση του μοντέλου μας θα μπορούσε να διαχειριστεί καλύτερα την λήψη αποφάσεων από έναν άνθρωπο.

3.4.2 Παράδειγμα 2 – Χρηματιστήριο και η αγορά μετοχών

Αρκετές προσπάθειες έχουν γίνει για χρήση ενισχυτικής μάθησης για την συναλλαγή μετοχών στο χρηματιστήριο με αρκετά καλά αποτελέσματα. Στο συγκεκριμένο παράδειγμα το μοντέλο μας μπορεί να χρησιμοποιηθεί για την λήψη αποφάσεων για την διαχείριση μετόχων. Το περιβάλλον σε αυτή την περίπτωση θα κατασκευαστεί βάσει του χρηματιστηρίου και θα έχει τις βασικές του λειτουργείες και ιδιότητες όπως η αγορά και πώληση μετοχών. Το σημαντικό όμως όπως και σε άλλες περιπτώσεις θα είναι ο σωστός υπολογισμός του κόστους κάθε πιθανής απόφασης. Το κόστος αυτό μπορεί να παίρνε υπόψη πολλές παραμέτρους κάτι το οποίο μπορεί να είναι δύσκολο για έναν άνθρωπο. Οι παράμετροι αυτοί μπορεί να είναι οι τιμές την συγκεκριμένη στιγμή, προβλέψεις για μελλοντικές τιμές, προηγούμενο ιστορικό και ακόμα μελλοντικό κέρδος από την κράτηση μετοχών. Ακόμη η κάθε ενέργεια μπορεί να αποτελείτε από ένα σύνολο αποφάσεων για την αγορά, πώληση και κράτηση πολλών μετοχών ταυτόχρονα. Επίσης το ανεπιθύμητο συμβάν μπορεί να είναι η χρεοκοπία ή η ζημία από μια συγκεκριμένη μετοχή. Για περαιτέρω κατανόηση θα παραθέσουμε το πιο κάτω απλό παράδειγμα με ενδεικτικές ενέργειες και κόστη.

Ενδεικτικές ενέργειες:

1. Αγορά μετοχής M1
2. Πώληση μετοχής M2
3. Αγορά μετοχής M1 και πώληση μετοχής M2
4. Αγορά μετοχής M1 και να κρατήσουμε την μετοχή M3

Πίνακας 3-4: Κόστη παραδείγματος 2

Ενέργεια	Κόστος
1	-200
2	250
3	-450
4	-350

Τα κόστη που φαίνονται πιο πάνω αναπαριστούν την ζημία που θα έχουμε με αποτέλεσμα οι αρνητικές τιμές που προκύπτουν να είναι κέρδος. Σύμφωνα με τις τιμές του πιο πάνω πίνακα η ενέργεια 2 θα μας επιφέρει ζημία ενώ οι υπόλοιπες ενέργειες θα μας επιφέρουν κέρδος με την ενέργεια 3 να είναι η πιο επικερδή. Φυσικά οι τιμές αυτές μπορούν να αλλάξουν λαμβάνοντας υπόψη περισσότερες παραμέτρους για τον υπολογισμό του κόστους ή καθώς αλλάζουν οι τιμές των μετοχών. Αποτέλεσμα του σωστού υπολογισμού των τιμών του κόστους θα έχει ως αποτέλεσμα να μάθει ο πράκτορας μας ποιες αποφάσεις είναι καλύτερο να πάρει έτσι ώστε να μεγιστοποιήσει το κέρδος από τις μετοχές.

3.4.3 Παράδειγμα 3 – Συστήμα συστάσεων

Ομοίως με τα προηγούμενα παραδείγματα το μοντέλο μας μπορεί να χρησιμοποιηθεί για την σύσταση σε χρήστες κάποιου συστήματος. Στο συγκεκριμένο παράδειγμα ενδιαφέρον θα έχει ο υπολογισμός του κόστους, που στην συγκεκριμένη περίπτωση θα αναπαριστά το κατά πόσο είναι πιθανών να συσταθεί στον χρήστη κάποιο προϊόν, ταινία, διαφήμιση κ.ο.κ. . Λαμβάνοντας διάφορα δεδομένα από κάποιο χρήστη θα μπορούσε να κατασκευάζεται σαν κόστος ποιο προϊόν ή ταινία ή ακόμη και διαφήμιση θα ήταν καλύτερα να προτείνουμε στο χρήστη και ανεπιθύμητο συμβάν την μη σύσταση ή την μην λάβει υπόψη ο χρήστης την σύσταση. Αναλυτικά το κόστος που αναφέραμε θα μπορούσε να είναι το κέρδος που θα έχουμε από τις διαφημίσεις, τις προβολές που θα έχει η ταινία, ή το πόσο κοντά είναι αυτό που θέλουμε να προτείνουμε στο χρήστη βάσει των στοιχείων που έχουμε για αυτόν. Η σύσταση αυτή μπορεί να μην αποσκοπεί σε ένα συγκεκριμένο προϊόν αλλά μια κατηγορία προϊόντων και να προσαρμόζεται βάσει των συνήθειών του χρήστη. Έτσι το μοντέλο μας να εκτελεί την κατάλληλη ενέργεια για την σύσταση στον συγκεκριμένο χρήστη.

Ενδεικτικές αποφάσεις

1. Σύσταση προϊόντος Α

2. Σύσταση κατηγορίας προϊόντος Β
3. Σύσταση διαφήμισης Γ

Πίνακας 3-5: Κόστη παραδείγματος 3

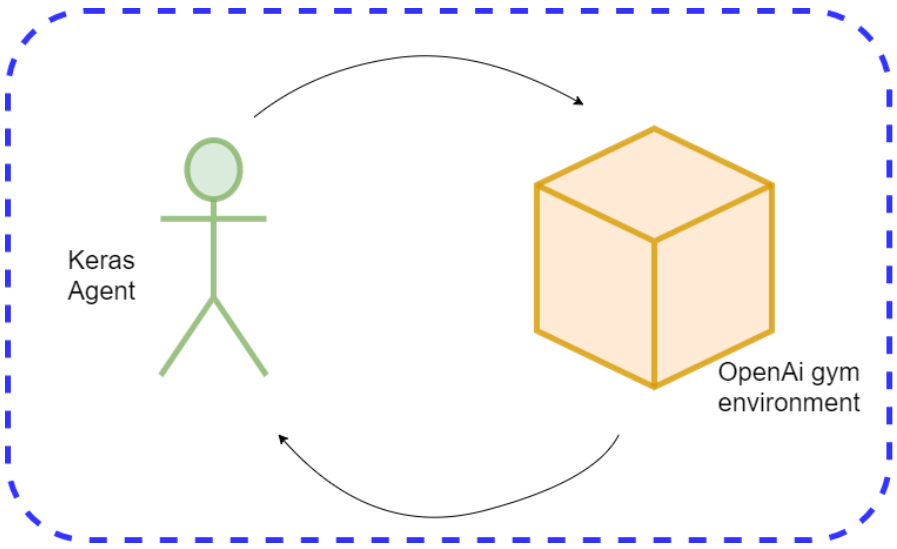
Ενέργεια	Κόστος
1	-20
2	-30
3	-15

Εκτός από τα 3 παραδείγματα που παραθέσαμε πιο πάνω το μοντέλο μας μπορεί να προσαρμοστεί και να μοντελοποίηση ότι προβλήματα λήψης απόφασης του δοθεί, φτάνει να υπάρχει αντιστοίχιση κόστους-απόφασης. Φυσικά τα συγκεκριμένα παραδείγματα που δώσαμε είναι αρκετά απλά και βλέποντας τον πίνακα μπορούμε πολύ εύκολα να συμπεράνουμε την βέλτιστη ενέργεια. Ο σκοπός του μοντέλου μας όμως είναι να βρίσκει την βέλτιστη αυτή ενέργεια όταν το πλήθος και η πολυπλοκότητα τον ενεργειών και του κόστους τους είναι πολύ μεγάλη. Κάτι το οποίο θα ήταν δύσκολο για ένα άνθρωπο, καθώς δεν θα μπορούσε εύκολα να διαχειριστεί τις πολλές πληροφορίες των ενεργειών και τον κοστών τους.

4 Υλοποίηση

4.1 Εισαγωγή

Η υλοποίηση του μοντέλου μας βασίζεται στην δημιουργία δύο επιμέρους στοιχειών, του περιβάλλοντος και του πράκτορα. Συγκεκριμένα χρησιμοποιήσαμε τον OpenAI gym για την δημιουργία του περιβάλλοντος μας και το Keras για την δημιουργία του πράκτορα μας. Τα 2 αυτά στοιχεία αλληλοεπιδρούν μεταξύ τους ανταλλάζοντας πληροφορίες με σκοπό την υλοποίηση του μοντέλου που θέλουμε να κατασκευάσουμε. Στα επόμενα υποκεφάλαια θα αναλύσουμε αρχικά την υλοποίηση του περιβάλλοντος μας στο OpenAI gym και ακολούθως το πως χρησιμοποιήσαμε το Keras για την υλοποίηση του πράκτορα μας. Τέλος θα εξηγήσουμε το πως αυτά αλληλοεπιδρούν μεταξύ τους και πως τα αξιολογούμε.

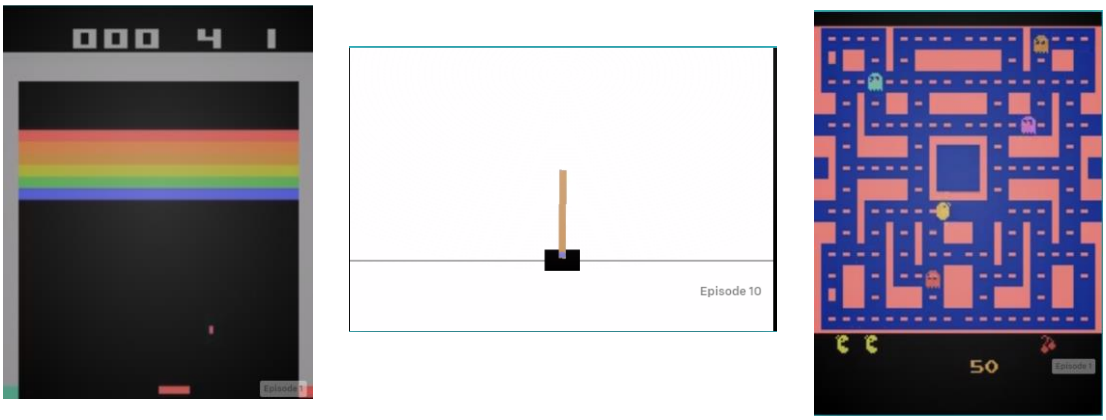


Εικόνα 4-1: Αναπαράσταση στοιχειών της προσέγγισης

4.2 Υλοποίηση προσαρμοσμένου περιβάλλοντος σε OpenAI

4.2.1 Τι είναι το OpenAI Gym

Το OpenAI είναι ένα toolkit για την ανάπτυξη και σύγκριση αλγορίθμων Ενισχυτικής Μάθησης όπου με ένα τυποποιημένο τρόπο μπορούμε να ορίσουμε το περιβάλλον μας και να εκπαιδεύσουμε με ευκολία τον πράκτορα μας [21]. Η επιλογή για το συγκεκριμένο toolkit έγινε λόγο της ευκολίας χρήσης έτοιμων αλγορίθμων ενισχυτικής μάθησης σε περιβάλλοντα που κατασκευάζονται με αυτό. Αν και το OpenAI gym παρέχει πληθώρα από έτοιμα περιβάλλοντα για διάφορες χρήσεις όπως προσομοιώσεις χρηματιστηρίου και παιχνιδιών, εμείς κατασκευάσαμε το δικό μας προσαρμοσμένο περιβάλλον (custom environment) για να προσομοιώσουμε το μοντέλο μας. Μερικά παραδείγματα από έτοιμα περιβάλλοντα στο OpenAi gym είναι παιχνίδια από την γνωστή κονσόλα Atari (Breakout, MsPacman, Pong-ram). Ακόμη παραδείγματα από προσαρμοσμένα περιβάλλοντα είναι το CartPole που σκοπός του είναι η ισορροπία μια ράβδου πάνω σε ένα καροτσάκι και το FrozenLake όπου σκοπός είναι ο πράκτορας να φτάσει σε ένα τερματικό σημείο «περπατώντας» πάνω σε μια παγωμένη λίμνη με τρύπες που σε οδηγούν μέσα στο νερό. Στα παιχνίδια που αναφέραμε σκοπός τον πρακτόρων είναι η εξερεύνηση των περιβαλλόντων και η συγκέντρωση της μεγαλύτερης ανταμοιβής.



Εικόνα 4-2: Εικόνες από τα παιχνίδια BreakOut,CartPole και MsPacman αντίστοιχα

Η κατασκευή του προσαρμοσμένου περιβάλλοντος μας έγινε με βάση την MDP που κατασκευάσαμε και εξηγήσαμε στο προηγούμενο κεφάλαιο. Ακόμα καλό είναι να αναφέρουμε ότι το περιβάλλον μας αφού γίνει καταχώριση του στο OpenAI gym θα είναι διαθέσιμο για να το χρησιμοποιήσει όποιος θέλει για να μοντελοποιήσει βάσει της δικής μας προσέγγισης ένα δικό του πρόβλημα. Το περιβάλλον μας και όλοι οι αλγόριθμοι υλοποιήθηκαν στην γλώσσα Python η όποια είναι η πιο δημοφιλής στην ενισχυτική μάθηση και το OpenAI Gym.

Αρχικά χρειάστηκε να δημιουργήσουμε το περιβάλλον μας με τις κατάλληλες καταστάσεις, ενέργειες και κανόνες σύμφωνα με την τυποποιημένη υλοποίηση του OpenAI Gym. Η υλοποίηση ξεκινά με την δήλωση του περιβάλλοντος μας ως κλάση στην Python.

Ακολούθως χρειάζεται η υλοποίηση των ακόλουθων τεσσάρων συναρτήσεων :

- __init__
- Step
- Reset
- Render

Οι πιο πάνω είναι απαραίτητες για την λειτουργία του περιβάλλοντος στο OpenAI Gym. Πέραν αυτών μπορούν να υλοποιηθούν και άλλες συναρτήσεις για την σωστή λειτουργία του περιβάλλοντος.

4.2.2 Ανάλυση συναρτήσεων OpenAI Gym

1. **__init__**

   Η πρώτη και πιο σημαντική συνάρτηση του περιβάλλοντος, καθώς εδώ γίνεται η αρχικοποίηση και δημιουργία του περιβάλλοντος. Πιο συγκεκριμένα γίνεται ορισμός και δημιουργία των παρατηρήσεων (observation_space) και των ενεργειών (action_space) του πράκτορα μαζί με τις υπόλοιπές παραμέτρους του περιβάλλοντος. Στην δική μας περίπτωση οι υπόλοιπες παράμετροι ήταν η κατάσταση στην οποία βρισκόμαστε, τα κόστη των αποφάσεων και το κόστος του ανεπιθύμητου συμβάν.

2. **Step**

   Αυτή η συνάρτηση είναι υπεύθυνη για την διαχείριση των ενεργειών του πράκτορα. Παίρνει ως παράμετρο την ενέργεια (action) και υπολογίζει σε ποια κατάσταση (state) θα βρεθεί ο πράκτορας σύμφωνα με αυτή. Επίσης επιστρέφει τέσσερις τιμές σε μορφή τούπλας, (observation, reward, done, info).

   - **Observation**: η παρατήρηση του πράκτορα για το περιβάλλον μετά από την ενέργεια του
   - **Reward:** η ανταμοιβή του από το την ενέργεια του
   - **Done:** μια Boolean μεταβλητή η οποία είναι True ένα ο πράκτορας έχει φτάσει στην τελική και επιθυμητή κατάσταση, αλλιώς επιστρέφει False.
   - **Info:** διαγνωστικές πληροφορίες χρήσιμες για τον εντοπισμό σφαλμάτων

3. **Reset**

   Με αυτή την συνάρτηση επαναφέρουμε τον περιβάλλον μας στις αρχικές του συνθήκες. Δηλαδή επιστρέφουμε στην κατάσταση κινδύνου και μηδενίζουμε την συνολική ανταμοιβή.

4. **Render**

Εδώ γίνεται γραφική απόδοση του περιβάλλοντος την συγκεκριμένη κατάσταση στην οποία καλείται.

Επίσης αξίζει να κάνουμε και αναφορά στα spaces (χώρους) του OpenAI gym, τα οποία είναι τρόπος με τον οποίο δηλώνονται οι ενέργειες και οι παρατηρήσεις για το περιβάλλον μας.

- Discrete (n) – Διακριτή δήλωση χώρου, n τιμών
- Box (low = k, high = m) – Δήλωση συνεχούς χώρου με όρια k και m.

Ο ρόλος των spaces είναι ουσιαστικά ο ορισμός της εισόδου και τις εξόδου του περιβάλλοντος. Δηλαδή ο τρόπος με τον οποίο θα δέχεται την ενέργεια που πρέπει να εκτελεστεί το περιβάλλον και ο τρόπος με τον οποίο θα βγάζει στην έξοδο του την παρατήρηση του πράκτορα για το περιβάλλον. Ο σωστός ορισμός τους είναι σημαντικός καθώς αποτελούν τις πληροφορίες που ανταλλάζει ο πράκτορας με το περιβάλλον. Για παράδειγμα στο περιβάλλον του CartPole το space των κινήσεων είναι Discrete (2), να κινηθεί το καροτσάκι δεξιά και αριστερά ενώ η παρατήρηση είναι Box(4) και περιλαμβάνει την θέση και την ταχύτητα του καροτσιού όπως επίσης την γωνία της ράβδου και την γωνιακή της ταχύτητα. Αφού κατασκευαστεί ένα περιβάλλον με τον πιο πάνω τρόπο είναι έτοιμο για να μπορούν να τρέξουν σε αυτό αλγόριθμοι ενισχυτικής μάθησης

4.2.3 Υλοποίηση περιβάλλοντος

Το προσαρμοσμένο περιβάλλον που θέλουμε να κατασκευάσουμε είναι η βάση του μοντέλου μας, και για αυτόν τον λόγο η σωστή δημιουργία του είναι πάρα πολύ σημαντική. Έτσι ξεκινήσαμε την υλοποίηση μας με το να κατασκευάσαμε το περιβάλλον μας σύμφωνα με όσα αναφέραμε πιο πάνω για το OpenAI gym. Λόγω του ότι το περιβάλλον μας δεν είναι σταθερό και οι ενδιάμεσες καταστάσεις μπορεί να διαφέρουν κάθε φορά για να δημιουργηθεί το περιβάλλον μας πρέπει απαραίτητα να του περάσουμε δύο μεταβλητές. Η πρώτη είναι μια λίστα με τα κόστη των διαθέσιμων αποφάσεων η οποία αποτελείται από θετικούς ή αρνητικούς ακεραίους και η δεύτερη μεταβλητή είναι το κόστος όταν δεν εκτελέσουμε κάποια ενέργεια, δηλαδή να μεταβούμε στην κατάσταση ανεπιθύμητου συμβάν Se.

Κατά την δημιουργία του περιβάλλοντος μας, καλείται η συνάρτηση __init__ στην οποία γίνεται η αρχικοποίηση και δημιουργία του περιβάλλοντος σύμφωνα με τις δύο μεταβλητές εισόδου. Κατά την αρχικοποίηση δηλώνουμε το action_space και observation_space του περιβάλλοντος μας, όπως επίσης μεταβλητές για την κατάσταση την οποία βρισκόμαστε και τα δοσμένα κόστη ενεργειών. Το action_space δηλώνεται ως Discrete με μέγεθος όσες είναι οι ενδιάμεσες καταστάσεις + 1 και το observation_space δηλώνεται ως Box με τιμές -1 μέχρι

όσες είναι οι ενδιάμεσες καταστάσεις +1, πιο συγκεκριμένα αποδίδουμε τις ακόλουθες τιμές στις καταστάσεις μας:

Πίνακας 4-1: Τιμές καταστάσεων

Κατάσταση	Αριθμός παρατήρησης
S_n	0
S_d	1
S_{Ai}	i+1
S_e	-1

Παραδοχή

Στην δική μας περίπτωση η κάθε κατάσταση δεν έχει τον ίδιο αριθμό ενεργειών (actions) αυτό όμως δεν μας το επιτρέπει το OpenAI gym και οι έτοιμοι αλγόριθμοι βαθιάς ενισχυτικής μάθησης . Έτσι στην δημιουργία του περιβάλλοντος μας για κάθε κατάσταση έχουμε τον ίδιο αριθμό ενεργειών όσα έχει και η κατάσταση κινδύνου Sd, που είναι ο μέγιστος αριθμός. Για να περιορίσουμε όμως τις ενέργειες που επιλέγει ο πράκτορας μας σε κάθε ενέργεια που δεν θα έπρεπε να υπάρχει, δίνουμε ένα μεγάλο κόστος ίσο με το άθροισμα όλων το κότσων των ενδιάμεσων καταστάσεων απόφασης και μεταφερόμαστε την τελική κατάσταση. Με αυτόν τον τρόπο «αναγκάζουμε» τον πράκτορα μας να μην εκτελεί αυτές τις ενέργειες. Το κόστος μη επιτρεπτής ενέργειας το αποκαλούμε τιμωρία (penalty) και η ανταμοιβή του συμβολίζεται με Rp .

Πίνακας 4-2: Αντιστοίχισης Τιμών Ανταμοιβών

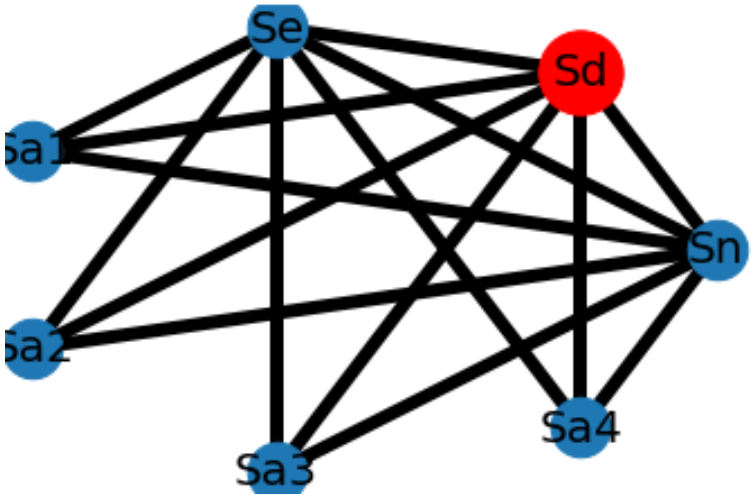
Κατάσταση	Ενέργεια	Επόμενη κατάσταση	Αμοιβή
Sd	0	Se	0
	1 μέχρι ν	SAi	RAi
Se	0	Sn	Re
	1 μέχρι ν	Sn	Rp
Sn	0	Sd	0
	1 μέχρι ν	Sn	Rp
SAi	0	Se	0
	1	Sn	0
	1 μέχρι ν	Sn	Rp

Μετά από την δημιουργία του περιβάλλοντος μας χρειάζεται να υλοποιήσουμε τις υπόλοιπες συναρτήσεις για την σωστή λειτουργεία του.

Ξεκινάμε με την υλοποίηση της συνάρτησης Step η οποία παίρνει σαν όρισμα την αριθμό της της ενέργειας που θα εκτελεστεί. Σε αυτήν την συνάρτηση δεδομένης της κατάστασης που βρισκόμαστε υπολογίζεται ποια θα είναι η ανταμοιβή μας, και αν η ενέργεια αυτή δεν είναι επιτρεπτή σαν ανταμοιβή παίρνουμε την τιμή Rp. Επιπρόσθετα υπολογίζεται η επόμενη κατάσταση στην οποία θα βρεθούμε. Αφού έχουν υπολογιστεί τα πιο πάνω η συνάρτηση Step μας επιστρέφει την τούπλα (observation,reward,done,info).

Ακολούθως υλοποιούμε την συνάρτηση Reset όπου απλά μηδενίζουμε τις μέχρι τώρα ανταμοιβές μας και μετακινούμαστε στην κατάσταση κινδύνου η οποία είναι και η αρχική μας κατάσταση.

Τέλος υλοποιούμε την συνάρτηση Render. Στην περίπτωση μας αύτη η συνάρτηση δεν παίζει ρόλο στην λειτουργεία του περιβάλλοντος μας και θα μπορούσαμε να την αφήσουμε κενή. Με την βοήθεια όμως της βιβλιοθήκης networkx αναπαριστούμε το περιβάλλον μας σαν γράφο, όπως δείξαμε στο Κεφάλαιο 3 και σημειώνουμε την κατάσταση στην οποία βρισκόμαστε με διαφορετικό χρώμα. Την συνάρτηση αυτήν την χρησιμοποιήσαμε μόνο στην αρχή της υλοποίησης όταν κάναμε δοκιμές στο περιβάλλον μας για να διαπιστώσουμε ένα δούλευε σωστά και όπως θέλαμε.



Εικόνα 4-3: Αναπαράσταση περιβάλλοντος με την βιβλιοθήκη networkx

Αφού τελειώσαμε με την υλοποίηση του περιβάλλοντος μας πριν προχωρήσουμε σε βαθιά μάθηση κάναμε κάποιες δοκιμές στο περιβάλλον μας τρέχοντας ένα απλό αλγόριθμο Q-learning. Έχοντας επιβεβαιώσει ότι το περιβάλλον μας λειτουργά σωστά με τον αλγόριθμο Q-learning είμασταν σε θέση να χρησιμοποιήσουμε αλγορίθμους βαθιάς μάθησης.

4.3 Υλοποίηση DeepRL αλγορίθμων μέσω του Keras

Με το περιβάλλον μας έτοιμο το μόνο που έμενε ήταν να τρέξουμε σε αυτό αλγορίθμους βαθιάς ενισχυτικής μάθησης. Για την εκπαίδευση του πράκτορα με βαθιά μάθηση χρησιμοποιήσαμε Deep Q-learning με την βοήθεια του Keras χρησιμοποιώντας τον DQN Agent.

4.3.1 Τι είναι το Keras

Το Keras είναι μια βιβλιοθήκη της Python η οποία παρέχει έτοιμα εργαλεία και αλγορίθμους με τους οποίους μπορούμε εύκολα να δημιουργήσουμε, να εκπαιδεύσουμε και να εξάγουμε αποτελέσματα για βαθιά μάθηση [22]. Η βιβλιοθήκη Keras έχει έτοιμους αλγορίθμους για τα περισσότερα που αναφέραμε στο Κεφάλαιο 2 (Θεωρητικό υπόβαθρο) για την ενισχυτική μάθηση. Στην δική μας περίπτωση θα χρησιμοποιήσουμε αλγορίθμους ενισχυτικής μάθησης και βαθιάς ενισχυτικής μάθησης. Συγκεκριμένα από την βιβλιοθήκη Keras χρησιμοποιήσαμε τους έτοιμους αλγορίθμους για τα ακόλουθα:

Πίνακας 4-3:Πίνακας παραμέτρων Keras

Νευρωνικό Δίκτυο	Είδος	Sequential
	Επίπεδα	Dense
		Flatten
	Ενεργοποίηση	Relu
		Sigmoid
		Softmax
Πράκτορας	Είδος	DQN Agent
	Βελτιστοποιητές (Optimizers)	Adam
		SGD
	Πολιτικές	BoltzmannQPolicy
		MaxBoltzmannQPolicy
		EpsGreedyQPolicy

Οι αλγόριθμοι του Keras ήταν υπεύθυνοι για την δημιουργία του πράκτορα, ο οποίος αλληλοεπιδρά με το περιβάλλον μας. Η δημιουργία αυτού του πράκτορα γίνεται με σχετική ευκολία έχοντας δεδομένο ότι το περιβάλλον το οποίο θα χρησιμοποιήσουμε είναι φτιαγμένο

με βάση το OpenAI gym και ότι λειτουργά σωστά. Το μόνο που χρειάζεται να κατασκευάσουμε είναι δύο πράγματα, τον νευρωνικό μας και τον πράκτορα μας.

4.3.2 Κατασκευή Νευρωνικού

Το νευρωνικό κατασκευάζεται με μια συνάρτηση που υλοποιήσαμε η οποία παίρνει σαν όρισμα στο σχήμα (shape), τις διαστάσεις δηλαδή του action_space και observation_space του περιβάλλοντος μας. Αυτά τα 2 τα χρειάζεται έτσι ώστε να οριστούν σωστά η είσοδος και η έξοδος του νευρωνικού μας. Καθώς οι διαστάσεις του action_space εξαρτούνται από το πλήθος των ενεργειών μας κάθε φορά που αλλάζει το περιβάλλον μας το νευρωνικό δίκτυο πρέπει να ξανά δημιουργείται.

4.3.3 Κατασκευή Πράκτορα

Στην δική μας υλοποίηση χρησιμοποιήσαμε τον DQNAgent από την βιβλιοθήκη Keras για τον πράκτορα μας. Η κατασκευή και λειτουργεία του πράκτορα προϋποθέτει την ύπαρξη του περιβάλλοντος και του νευρωνικού. Αν και σαν ορίσματα μπορεί να πάρει αρκετές παραμέτρους για την δημιουργία του εμείς χρησιμοποιήσαμε τις εξής παραμέτρους:

- Το νευρωνικό μας δίκτυο
- Την μνήμη που θα έχει από προηγούμενες ενέργειες.
- Την πολιτική την οποία θα χρησιμοποιεί.
- Τον αριθμό των ενεργειών που μπορεί να εκτελέσει.
- Τα βήματα που εκτελεί σαν «προθέρμανση» πριν να ξεκινήσει η εκπαίδευση του.
- Τον ρυθμό με τον οποίο θα μαθαίνει.

Τέλος για την λειτουργία του πράκτορα πρέπει να τον κάνουμε compile με τον βελτιστοποιητή και τις μετρικές που θέλουμε. Ο ρόλος του βελτιστοποιητή είναι να ελέγχει και να βελτιώνει την απώλεια (loss) του νευρωνικού δικτύου. Οι μετρικές με την σειρά τους είναι αποτελέσματα από την εκπαίδευση του πράκτορα και θα τα αναλύσουμε στην συνέχεια αυτού του κεφαλαίου.

4.4 Εκπαίδευση και αξιολόγηση του μοντέλου

Έχοντας υλοποιήσει πλέον όλα τα επιμέρους κομμάτια του μοντέλου μας που αναλύσαμε πιο πάνω είμαστε σε θέση να εκπαιδεύσουμε και να αξιολογήσουμε το μοντέλο μας. Σε αυτό το σημείο θα μας διευκολύνει το Keras, που είναι και ο λόγος που το επιλέξαμε για την υλοποίηση μας. Η εκπαίδευση και αξιολόγηση μέσο του Keras είναι ιδιαίτερα εύκολη και μπορεί να πραγματοποιηθεί ακόμη και με μόνο μια γραμμή κώδικα.

4.4.1 Εκπαίδευση πράκτορα

Η εκπαίδευση του πράκτορα γίνεται με την βοήθεια της συνάρτησης fit. Η συνάρτηση αυτή παίρνει τρία ορίσματα το περιβάλλον, τα βήματα εκπαίδευσης και τις επιστροφές (callbacks). Τα βήματα εκπαίδευσης είναι τα πόσα βήματα θα πραγματοποιηθούν κατά την εκπαίδευση, δηλαδή πόσες φορές θα κληθεί η συνάρτηση Step του περιβάλλοντος μας. Τα callbacks είναι έτοιμες ή προσαρμοσμένες συναρτήσεις που εκτελούνται κατά την διάρκεια της εκπαίδευσης. Αυτές οι συναρτήσεις μπορεί να χρησιμοποιηθούν για να σταματήσει η εκπαίδευση σε κάποιο σημείο, για την αποθήκευση των βαρών του μοντέλου, για την αποθήκευση παραμέτρων, η αλλαγή παραμέτρων κ.α. . Στην δική μας περίπτωση φτιάξαμε δύο δικές μας συναρτήσεις. Η πρώτη ήταν να αποθηκεύουμε αποτελέσματα των μετρικών από την εκπαίδευση σε μορφή json για μεταγενέστερη χρήση και αξιολόγηση. Η δεύτερη ήταν για την αποθήκευση του νευρωνικού που πρότεινε την καλύτερη λύση μέχρι την στιγμή που καλούταν η συγκεκριμένη συνάρτηση.

4.4.2 Αξιολόγηση μοντέλου

Όπως αναφέραμε και στην εκπαίδευση του πράκτορα η αξιολόγηση του μοντέλου μας έγινε από τις τιμές των μετρικών που εξαγάγαμε από τα json αρχεία κατά την διάρκεια της εκπαίδευσης.

4.4.2.1 Μετρικές

Το Keras από μόνο του παρέχει μετρικές για την εκπαίδευση του πράκτορα. Από μόνος του ο DQN Agent παρέχει αποτελέσματα για αρκετές μετρικές, όμως εμείς προσθέσαμε ακόμα δύο τις mae και accuracy. Αναλυτικά όλες οι μετρικές που χρησιμοποιήσαμε φαίνονται στον Πίνακας 4-4 πιο κάτω και ακολούθως εξηγούμε μερικές από αυτές:

Πίνακας 4-4: Μετρικές που χρησιμοποιήσαμε

Μετρική	Εξήγηση
Loss	Σφάλμα πρόβλεψης του νευρωνικού
Mae	Μέσο απόλυτο σφάλμα του νευρωνικού
Accuracy	Ακρίβεια του νευρωνικού
Mean_q	Μέση αναμενόμενη ανταμοιβή
Episode reward	Ανταμοιβή επεισοδίου
Episode steps	Βήματα ανά επεισόδιο
Duration	Διάρκεια εκπαίδευσης επεισοδίου

- **Loss**

  Το σφάλμα που υπολογίζεται είναι σε σχέση με την πρόβλεψη της επόμενης ενέργειας και υπολογίζεται με τον εξής τρόπο [23]:

  $$J(\theta) = \left|\left|\delta_{\tau+1}\right|\right|^2 = \ ||R_{t+1} + \gamma\theta^T\Phi(S_{t+1}) - \ \theta^T\Phi(S_t)||^2$$

  Όπου δ είναι το σφάλμα χρονικής διαφοράς (TD error), θ^T ο ανάστροφος πίνακας των βαρών του νευρωνικού και Φ(S) το διάνυσμα καταστάσεων.

- **Mae**

  Το μέσο απόλυτο σφάλμα μετρά το μέσο μέγεθος σφαλμάτων στις προβλέψεις του νευρωνικού μεταξύ των προβλεπόμενων $\hat{y}$ και πραγματικών τιμών y_i [24] .

  $$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n}$$

- **Accuracy**

  Η ακρίβεια υπολογίζεται ως ποσοστό των επιτυχών προβλέψεων του νευρωνικού

  $$Accurancy = \frac{Επιτυχείς\ Προβλέψεις}{Συνολικές\ Προβλέψεις}$$

- **Mean-q**

  Είναι η μέση ανταμοιβή που περιμένει ο πράκτοράς στην κατάσταση στην οποία βρίσκεται

Οι πρώτες 3 μετρικές είναι για το νευρωνικό μας δίκτυο. Λόγο του ότι χρησιμοποιούμε ενισχυτική μάθηση και λόγο της φύσης του δικού μας περιβάλλοντος, δεν ήταν αντιπροσωπευτικές για το μοντέλο μας, έτσι δεν τους δώσαμε μεγάλη βαρύτητα για την αξιολόγηση του μοντέλου . Το κύριο πρόβλημα παρατηρήθηκε όταν ο πράκτορας εξερευνούσε μια καινούργια κατάσταση του περιβάλλοντος με αποτέλεσμα να παρατηρούμαι μεγάλη ασυνέχεια και αστάθεια στις τιμές αυτών των μετρικών. Για τον πιο πάνω λόγο δώσαμε πολύ μεγαλύτερη βαρύτητα στις υπόλοιπες μετρικές, διότι μας ενδιέφερε κυρίως η εξερεύνηση των διαθέσιμων αποφάσεων και η εύρεση της βέλτιστης από αυτές.

4.4.2.2 Αξιολόγηση εκτός των Μετρικών

Πέραν από τις τιμές των ίδιων των μετρικών υπολογίσαμε και άλλες τιμές για την αξιολόγηση του μοντέλου μας. Οι τιμές αυτές υπολογίστηκαν βάσει των τιμών των μετρικών και κύριο σκοπό είχαν να μα δώσουν μια πιο ξεκάθαρη εικόνα για το τι συνέβαινε κατά την διάρκεια της εκπαίδευσης. Οι τιμές που υπολογίστηκαν ήταν οι εξής:

- Αριθμός μη επιτρεπτών κινήσεων
  - Πόσες μη επιτρεπτές κινήσεις εκτελέστηκαν
- Μέσος όρος ανταμοιβών

    - Ο μέσος όρος ανταμοιβών ανά επεισόδιο
- Μέσος όρος βημάτων
    - Πόσα βήματα εκτελέστηκαν κατά μέσο όρο ανά επεισόδιο
- Εξερευνημένες αποφάσεις
    - Πόσες από τις αποφάσεις που έχουμε εξερευνήθηκαν
- Κόστος καλύτερης απόφασης
    - Το κόστος της καλύτερης απόφασης που βρέθηκε

5 *Αξιολόγηση*

5.1 Εισαγωγή

Στο παρόν κεφάλαιο θα αναλύσουμε τα πειράματα που εκτελέσαμε και τα αποτελέσματα που εξαγάγαμε από αυτά. Για την αξιολόγηση του μοντέλου μας εκτελέσαμε μια σειρά από διαφορετικά πειράματα αλλάζοντας τις παραμέτρους του πράκτορα και του περιβάλλοντος. Μεγαλύτερη έμφαση δώσαμε στις παραμέτρους του πλήθους τον διαθέσιμων αποφάσεων, των κοστών τους και της πολιτικής του πράκτορα.

Η ιδέα για την αλλαγή τον παραμέτρων ήταν να αξιολογήσουμε το μοντέλο μας σε διαφορά σενάρια εφαρμογής, αλλάζοντας δηλαδή το περιβάλλον στο οποίο ο πράκτορας εκπαιδευόταν παρατηρώντας την απόδοση και τα αποτελέσματα του μοντέλου μας .

5.2 Αξιολόγηση μοντέλων επίλυσης DeepRL - Δοκιμές με διαφορετικές παραμέτρους στο νευρωνικό δίκτυο για εύρεση του βέλτιστου δικτύου

Αρχικά ξεκινήσαμε με τον πειραματισμό στον πράκτορα μας και συγκεκριμένα στο νευρωνικό του δίκτυο. Σκοπός αυτών των πειραμάτων ήταν να βρούμε τις παραμέτρους του νευρωνικού που μας δίνουν καλύτερα αποτελέσματα. Οι παράμετροι που εξετάσαμε ήταν το πλήθος των κρυμμένων επιπέδων του νευρωνικού, η συνάρτηση ενεργοποίησης των νευρώνων, ο βελτιστοποιητής (optimizer) και ο ρυθμός μάθησης του. Πιο αναλυτικά οι παράμετροι αυτοί φαίνονται στον ακόλουθο πίνακα:

Πίνακας 5-1: Παράμετροι νευρωνικού

Είδος Παραμέτρου	Τιμές
Κρυμμένα επίπεδα	2
	3
Συνάρτηση ενεργοποίησης	Relu
	Sigmoid
	Softmax
Βελτιστοποιητής	Adam
	SGD
Ρυθμός μάθησης	0.0001
	0.001
	0.005

Εκτελέσαμε πειράματα για όλους τους πιο πάνω συνδυασμούς παραμέτρων, δηλαδή 36 συνολικά ενώ κρατούσαμε σταθερές τις υπόλοιπες παραμέτρους του μοντέλου μας. Οι σταθερές παράμετροι ήταν οι εξής:

Πίνακας 5-2: Πίνακας σταθερών παραμέτρων

Παράμετρος	Τιμή
Αριθμός ενεργειών	50
Εύρος κοστών αποφάσεων	Μικρό (11-110)
Κόστος ανεπιθύμητου συμβάν	220
Πολιτική	BoltzmanQPolicy
Βήματα εκπαίδευσης	20000

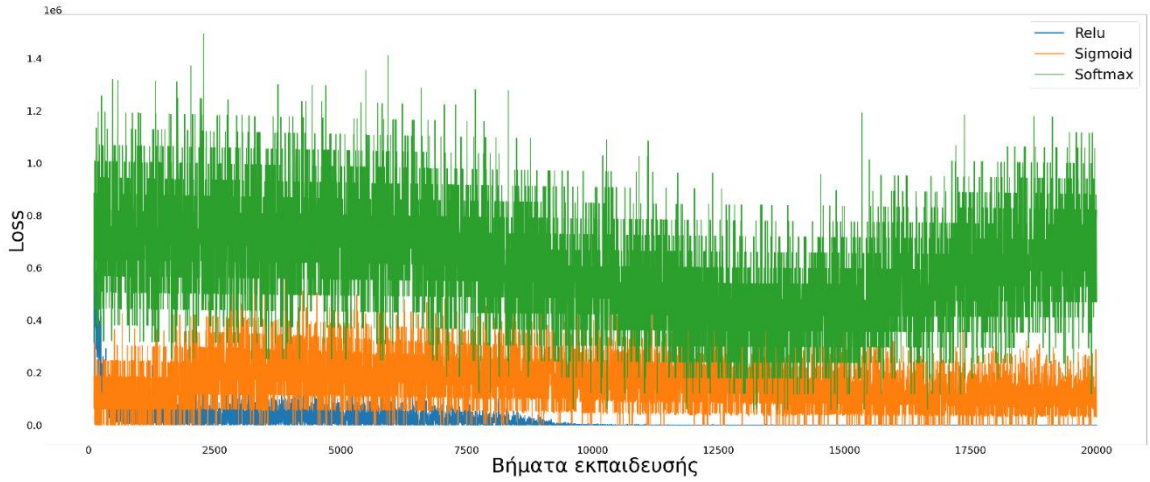
Για ευκολία σύγκρισης τον αποτελεσμάτων των πειραμάτων μας ακολουθήσαμε την εξής στρατηγική για την επιλογή του καλύτερου νευρωνικού μας. Από τους συνδυασμούς βελτιστοποιητών και αριθμό κρυφών επιπέδων πήραμε τα νευρωνικά με τα καλύτερα αποτελέσματα για κάθε συνάρτηση ενεργοποίησης. Έτσι είχαμε τέσσερα νευρωνικά για κάθε συνάρτηση ενεργοποίησης. Ακολούθως επιλέξαμε το καλύτερο από αυτά τα τέσσερα με αποτέλεσμα να έχουμε το καλύτερο από κάθε συνάρτηση ενεργοποίησης. Έτσι στο τέλος

συγκρίναμε το καλύτερο από κάθε συνάρτηση ενεργοποίησης και καταλήξαμε στις παραμέτρους που θα χρησιμοποιήσουμε για το νευρωνικό δίκτυο μας στα υπόλοιπα μας πειράματα. Πιο κάτω παραθέτουμε τις παραμέτρους των καλύτερων νευρωνικών και τις γραφικές των μετρικών τους.

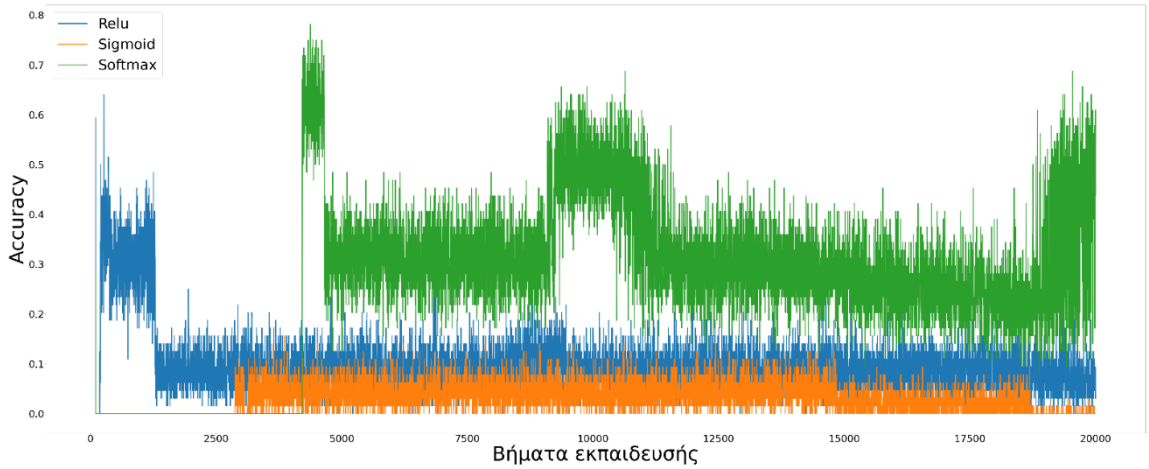
Πίνακας 5-3: Πίνακας καλύτερων νευρωνικών

Συνάρτηση ενεργοποίησης	**Κρυμμένα επίπεδα**	**Βελτιστοποιητής**	**Ρυθμός μάθησης**
Relu	3	Adam	0.001
Sigmoid	3	Adam	0.005
Softmax	2	Adam	0.005

Από όλα σχεδόν τα πειράματα που εκτελέσαμε, εκτός από μόνο μερικά δεν υπήρχε ξεκάθαρα κάποιο νευρωνικό δίκτυο το οποίο να ξεχώρισε αισθητά από τα υπόλοιπα. Αυτό οφείλεται κυρίως στην φύση του περιβάλλοντος και του τρόπου λειτουργείας των αλγορίθμων που χρησιμοποιήσαμε, καθώς υπάρχει τυχαιότητα στην επιλογή αποφάσεων κατά την εξερεύνηση του περιβάλλοντος από τον πράκτορα μας. Έτσι περισσότερη έμφαση δώσαμε στην εύρεση της βέλτιστης απόφασης, στην εν συνεχεία επιλογή της, στην αποφυγή επιλογής μη επιτρεπτών ενεργειών όπως επίσης και την εξερεύνηση του περιβάλλοντος μας. Έχοντας αυτά υπόψη καταλήξαμε στην επιλογή των πιο πάνω νευρωνικών για κάθε συνάρτηση ενεργοποίησης. Ακολούθως παραθέτουμε τις γραφικές των μετρικών για τα πιο πάνω νευρωνικά.

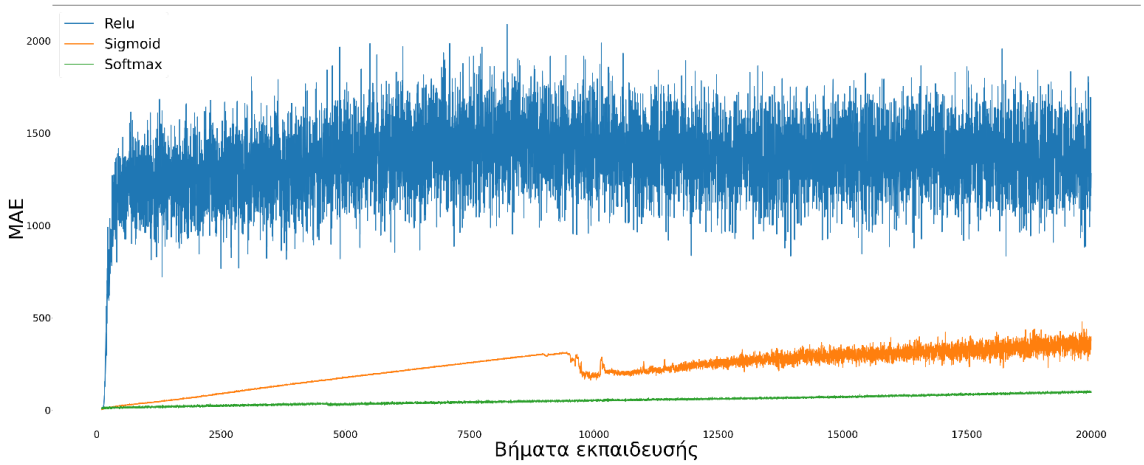


Εικόνα 5-1: Σφάλμα πρόβλεψης του νευρωνικού

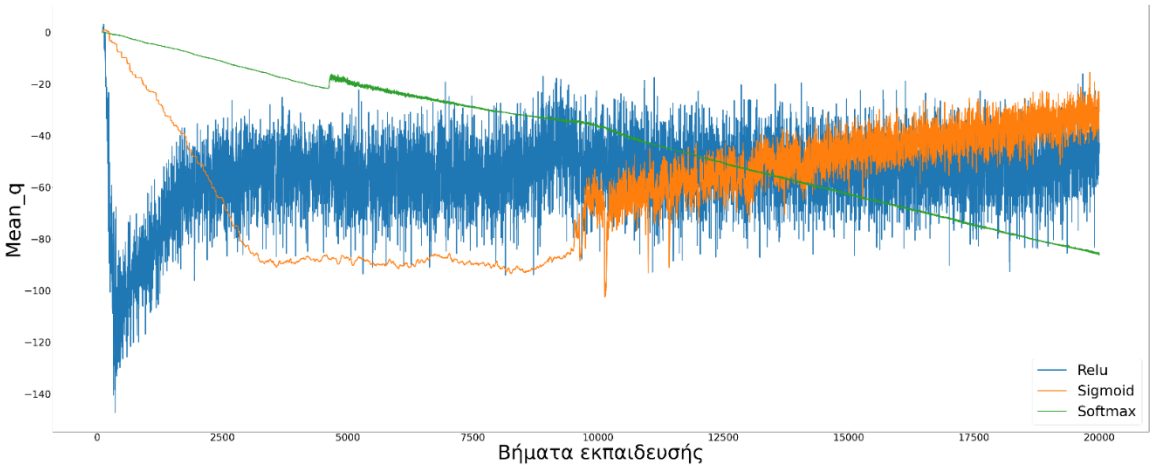


Εικόνα 5-2: Ακρίβεια του νευρωνικού

Στις πιο πάνω εικόνες (Εικόνα 5-1Εικόνα 5-2) μπορούμε να παρατηρήσουμε την συμπεριφορά του νευρωνικού μας δικτύου και πιο συγκεκριμένα το σφάλμα πρόβλεψης του για την επόμενη ενέργεια και το ποσοστό ακρίβειας των σωστών προβλέψεων. Για το σφάλμα θέλουμε να το ελαχιστοποιήσουμε ενώ την ακρίβεια να την μεγιστοποιήσουμε. Παρατηρούμε όμως αρκετές αστάθειές λόγο της εξερεύνησης νέων ενεργειών ιδιαίτερα στο νευρωνικό με την συνάρτηση ενεργοποίησης Softmax αν και έχει την μεγαλύτερη ακρίβεια. Τα άλλα δυο νευρωνικά έχουν αρκετά χαμηλό σφάλμα πρόβλεψης αλλά και ακρίβειας.

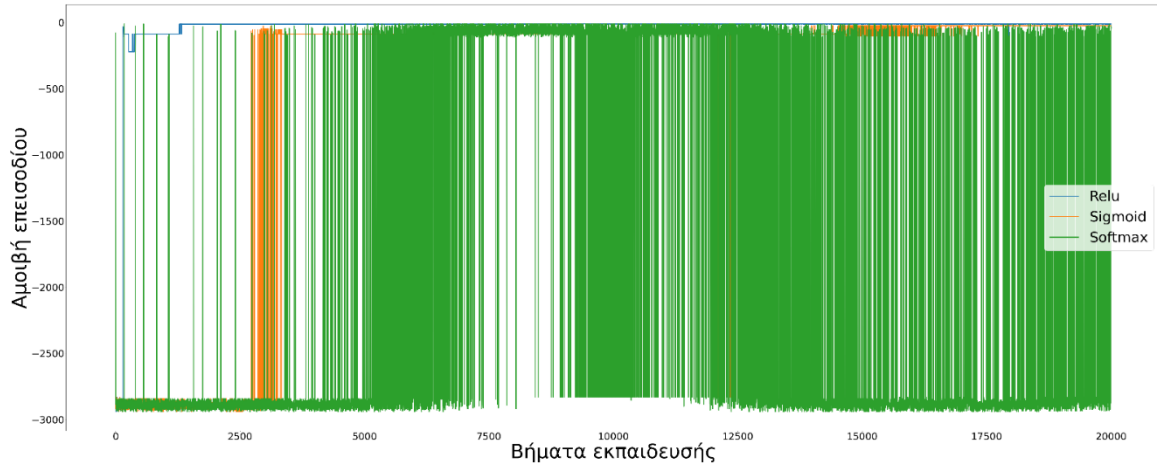


Εικόνα 5-3: Μέσο απόλυτο σφάλμα του νευρωνικού



Εικόνα 5-4: Μέση αναμενόμενη ανταμοιβή

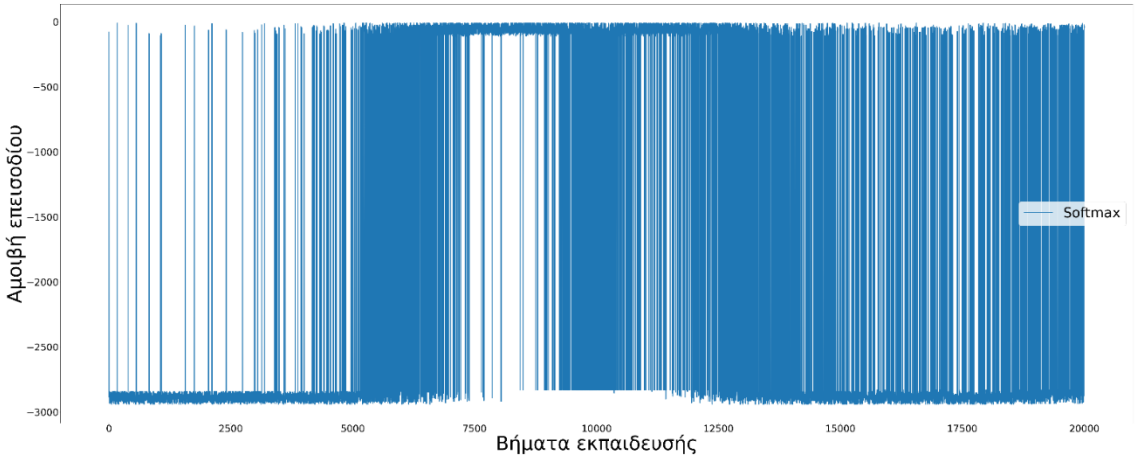
Στις δύο πιο πάνω εικόνες (Εικόνα 5-3Εικόνα 5-4) βλέπουμε το μέσο απόλυτο σφάλμα (MAE) και την μέση ανταμοιβή αντιστοιχία (Mean-q). Το μέσο απόλυτο σφάλμα είναι το τετράγωνο της διαφοράς της πρόβλεψης με την πραγματική τιμή και η μέση ανταμοιβή είναι η ανταμοιβή που περιμένουμε να πάρουμε. Μπορούμε να παρατηρήσουμε ότι το νευρωνικό με την συνάρτηση ενεργοποίησης Softmax αυξάνει συνεχώς την αναμενόμενη ανταμοιβή κάτι το οποίο σημαίνει ότι δεν εκπαιδεύεται σωστά. Αντίθετος τα άλλα δυο νευρωνικά φαίνονται να συγκλείνουν κοντά στο 0, όπου θα είναι και η βέλτιστη ανταμοιβή και ενέργεια που πρέπει να εκτελέσουν.



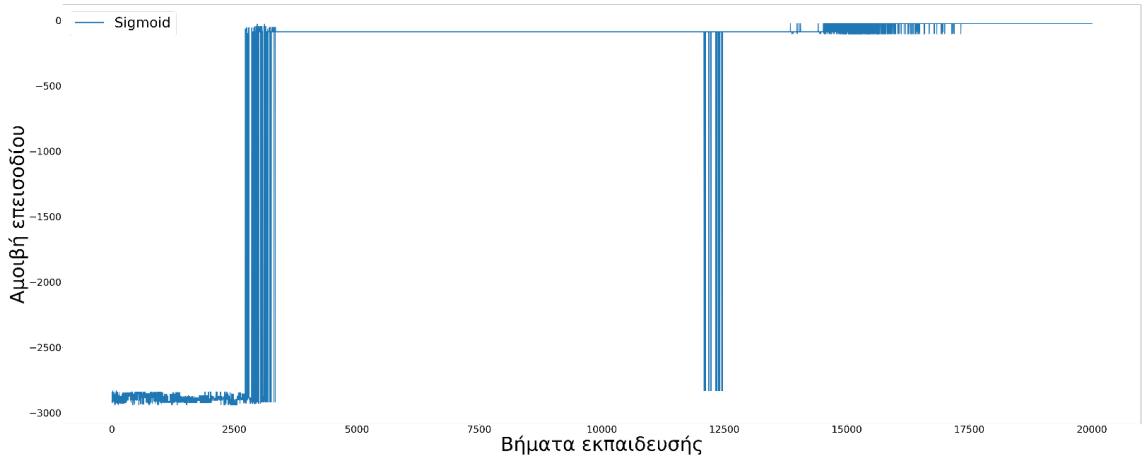
Εικόνα 5-5: Ανταμοιβή ανά επεισόδιο

Στην πιο πάνω Εικόνα 5-5 μπορούμε να δούμε την ανταμοιβή ανά επεισόδιο και των 3 νευρωνικών μαζί. Στόχος μας είναι η ανταμοιβή όσο πιο κοντά στο 0 γίνεται που είναι το κόστος που θα έχουμε με τις ενέργειες που θα εκτελέσουμε. Οι 3 εικόνες (Εικόνα 5-7Εικόνα

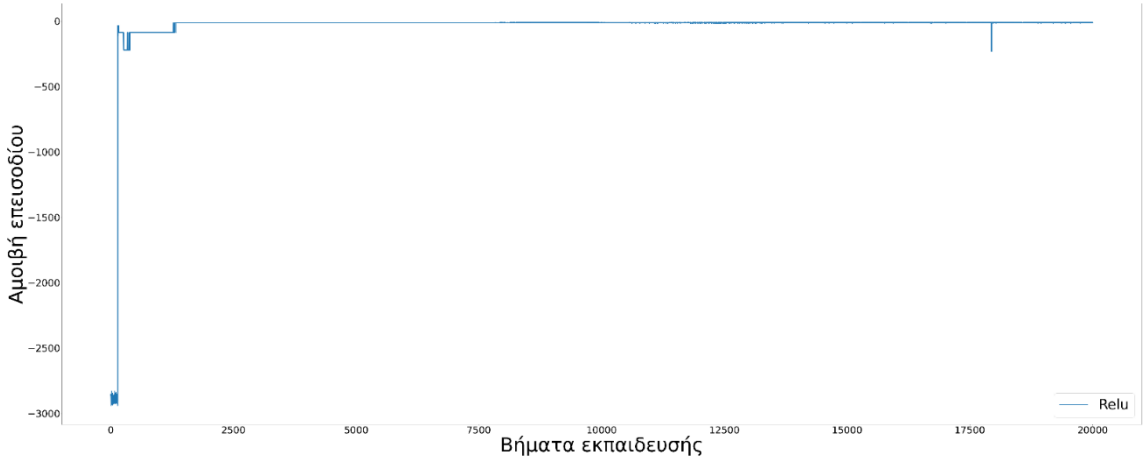
5-6Εικόνα 5-8) που ακολουθούν παρουσιάζουν αυτές τις ανταμοιβές για κάθε νευρωνικό ξεχωριστά.



Εικόνα 5-8: Ανταμοιβή ανά επεισόδιο - Softmax

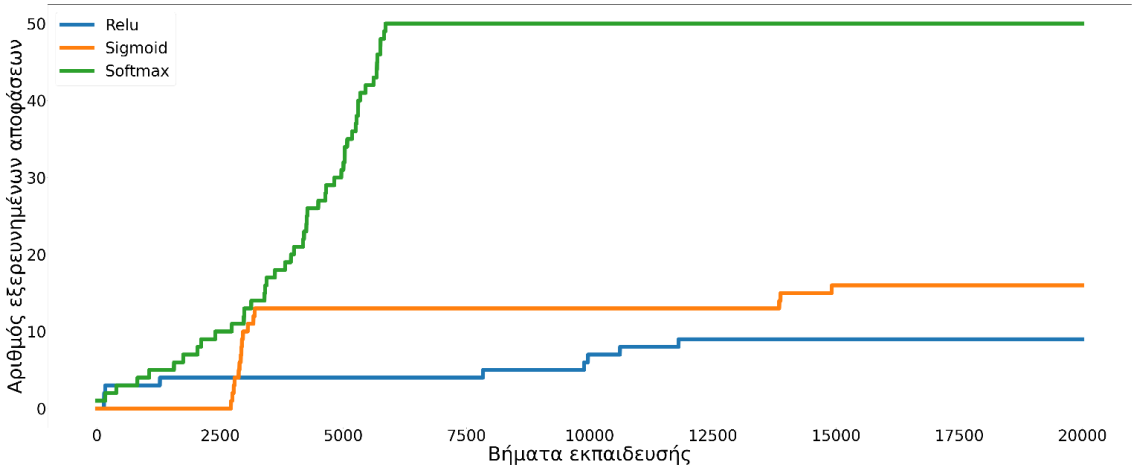


Εικόνα 5-7: Ανταμοιβή ανά επεισόδιο - Sigmoid

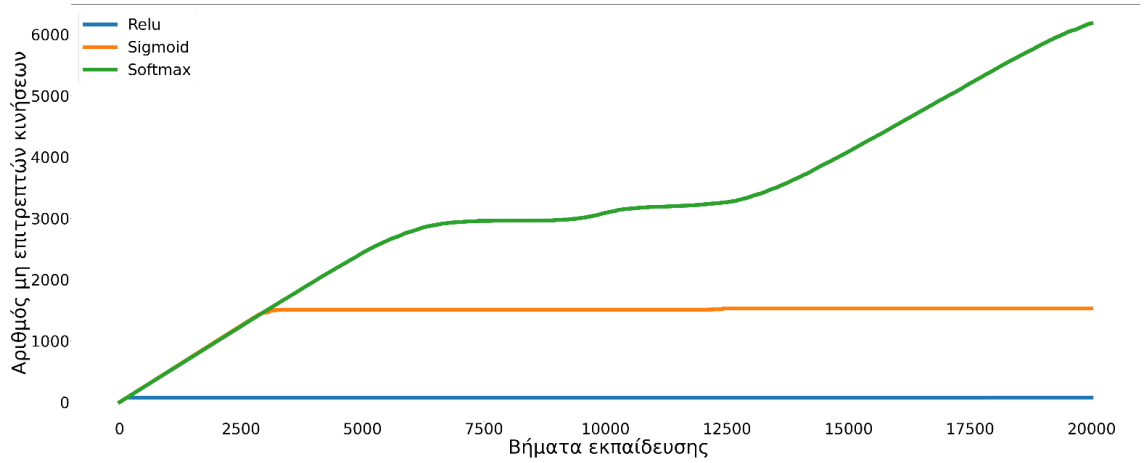


Εικόνα 5-6: Ανταμοιβή ανά επεισόδιο - Relu

Για τα νευρωνικά που υλοποιήθηκαν με Relu και Sigmoid παρατηρούμε μια συνέχεια στην επιλογή των ενεργειών με χαμηλό κόστος, ενώ το νευρωνικό με την Softmax παρατηρούμε δυσκολία στην επιλογή των ενεργειών με χαμηλό κόστος.



Εικόνα 5-9: Εξερευνημένες αποφάσεις



Εικόνα 5-10: Μη επιτρεπτές ενέργειες

Οι δύο τελευταίες εικόνες (Εικόνα 5-9Εικόνα 5-10) μας δείχνουν το πλήθος των εξερευνημένων αποφάσεων και το πλήθος των μη επιτρεπτών κινήσεων αντίστοιχα για κάθε συνάρτηση ενεργοποίησης. Εμφανές είναι ότι το νευρωνικό με την συνάρτηση Softmax έχει την μεγαλύτερη εξερεύνηση αλλά και είναι πιο επιρρεπές στην εκτέλεση μη επιτρεπτών κινήσεων. Τα άλλα δυο νευρωνικά έχουν περίπου την ίδια εξερεύνηση αλλά το νευρωνικό με την συνάρτηση ενεργοποίησης Relu έχει αισθητά τις λιγότερες μη επιτρεπτές κινήσεις. Πέραν από τις γραφικές καλό θα είναι να αναφέρουμε το κόστος των καλύτερων αποφάσεων που πάρθηκαν από τα νευρωνικά έχοντας δεδομένου ότι η βέλτιστη απόφαση είχε κόστος 10.

Πίνακας 5-4: Βέλτιστα κόστη ανά συνάρτηση ενεργοποίησης

Συνάρτηση Ενεργοποίησης Νευρωνικού	Κόστος καλύτερης απόφασης
Rely	10
Sigmoid	23
Softmax	10

Τέλος επιλέξαμε ένα από τα 3 υποψήφια νευρωνικά δίκτυα για να χρησιμοποιήσουμε στα υπόλοιπα μας πειράματα.

Παρατηρήσεις για κάθε ένα από τα 3 νευρωνικά δίκτυα:

- **Softmax**

  Αν και βρήκε την βέλτιστη απόφαση και είχε την μεγαλύτερη εξερεύνηση στο περιβάλλον, έδειξε μεγάλη αστάθεια στις υπόλοιπες μετρικές και ειδικά στην επιλογή απόφασης καθώς πολλές φορές επέλεγε μη επιτρεπτές κινήσεις.
- **Sigmoid**

  Δεν βρήκε την βέλτιστη απόφαση και οι μετρικές του ήταν μεταξύ των μετρικών των άλλον 2 νευρωνικών δικτύων. Γενικά είχε καλά αποτελέσματα και έδειχνε να συγκλίνει προς την βέλτιστη απόφαση.
- **Relu**

  Έδειξε συνολικά την καλύτερη εικόνα καθώς βρήκε την βέλτιστη λύση και έδειξε συνέχεια στην επιλογή της βέλτιστης απόφασης ενώ παράλληλα απέφυγε τις μη επιτρεπτές κινήσεις.

Έχοντας υπόψη τα ποιο πάνω καταλήξαμε στην επιλογή του νευρωνικού δικτύου με την συνάρτηση ενεργοποίησης Relu, και το χρησιμοποιήσαμε για όλα τα επόμενα πειράματα που θα αναφέρουμε.

5.3 Αξιολόγηση μοντέλου με διαφορετικές παραμέτρους στο περιβάλλον και στην πολιτική του πράκτορα

Έχοντας λοιπόν επιλέξει το νευρωνικό δίκτυο με το οποίο θα εκπαιδεύσουμε τον πράκτορα μας ξεκινήσαμε τα υπόλοιπα πειράματα για το μοντέλο μας. Στα πειράματα που

ακολουθούν σκοπός μας είναι να παρατηρήσουμε την απόδοση του μοντέλου μας σε διάφορα περιβάλλοντα ανάλογα με την πολιτική με την οποία χρησιμοποιεί ο πράκτορας μας. Όσων αφορά το περιβάλλον εξετάσαμε την απόδοση του μοντέλου μας σε διαφορετικό αριθμό πλήθους διαθέσιμών αποφάσεων, πλήθους ενεργειών δηλαδή, και διαφορετικού εύρος τιμών που είχαν τα κόστη κάθε απόφασης. Συγκεκριμένα για τα κόστη χρησιμοποιήσαμε τρία εύρη κοστών, από τα οποία παίρναμε τυχαία μοναδική τιμή για κάθε κόστος

Πίνακας 5-5: Κόστη Ενεργειών

Εύρος κόστους	**Τιμές για n αριθμό αποφάσεων**	
	Κάτω όριο	Άνω όριο
Ακριβές	10	n +11
Μικρό	10	2n +10
Μεγάλο	10	10n + 10

Στον πίνακα που ακολουθεί δείχνουμε πιο αναλυτικά όλες τις τιμές χρησιμοποιήσαμε για να εξετάσουμε τις πιο πάνω παραμέτρους:

Πίνακας 5-6: Παράμετροι Πειραμάτων

Παράμετρος	**Τιμή**
Αριθμός διαθέσιμών αποφάσεων	20
	50
	500
	1000
	2000
Εύρος κόστους	Ακριβές
	Μικρό
	Μεγάλο
Πολιτική	Boltzmann Q policy
	Max Boltzman Q policy
	Epsilon greedy Q policy

Επιπρόσθετα για κάθε συνδυασμό των πιο πάνω παραμέτρων εκπαιδεύσαμε τον πράκτορα μας σε 3 φορές με 10 000, 50 000 και 100 000 βήματα ανά εκπαίδευση. Οι πιο πάνω συνδυασμοί των παραμέτρων έχουν ως αποτέλεσμα να έχουμε συνολικά 135 εκπαιδεύσεις του μοντέλου μας. Καθώς οι διαφορές σε μερικές εκπαιδεύσεις δεν είναι μεγάλες θα παραθέσουμε τα αποτελέσματα μερικών πειραμάτων που εκτελέσαμε που είναι πιο ενδεικτικές. Συγκεκριμένα θα παραθέσουμε 2 πειράματα, ένα για ένα για 500 αποφάσεις και ένα για 2000 αποφάσεις. Σε κάθε πείραμα θα δίνουμε τις τιμές για κάθε πολιτική και εύρος κόστους για 100 000 βήματα εκπαίδευσης.

5.3.1 Πείραμα 1

Οι τιμές των παραμέτρων που χρησιμοποιήθηκαν στο συγκεκριμένο πείραμα για την εκπαίδευση του πράκτορα, φαίνονται στον Πίνακας 5-7. Στο συγκεκριμένο πείραμα εξετάζουμε την απόδοση του μοντέλου μας σε ένα μεσαίο αριθμό διαθέσιμων αποφάσεων. Τα αποτελέσματα του πειράματος φαίνονται στον Πίνακας 5-8.

Πίνακας 5-7 : Παράμετροι Πειράματος 1

Παράμετρος	**Τιμή**
Αριθμός διαθέσιμών ενεργειών	500
Βήματα εκπαίδευσης	100 000
Εύρος κόστους	Ακριβές (Just)
	Μικρό (Small)
	Μεγάλο (Big)
Πολιτικές	Boltzmann Q policy
	Max Boltzman Q policy
	Epsilon greedy Q policy

Πίνακας 5-8 : Αποτελέσματα Πειράματος 1

Πολιτική	**Boltzmann**			**Max Boltzman**			**Epsilon greedy**		
Εύρος Κόστους	Ακριβές	Μικρό	Μεγάλο	Ακριβές	Μικρό	Μεγάλο	Ακριβές	Μικρό	Μεγάλο
Αριθμός μη επιτρεπτών κινήσεων	48681	42901	17064	16368	3573	2102	6638	8126	10279
Μέσος όρος αμοιβών	-253	-518	-2162	-202	-254	-783	-441	-198	-1234
Μέσος όρος Βημάτων	2.0160	2.1634	2.0211	2.2616	2.0247	2.0000	2.0132	2.0158	2.0359
Αποφάσεις που εξερεύνησε	406	462	499	165	146	467	500	500	499
Κόστος καλύτερης απόφασης	-10	-11	-17	-10	-19	-49.0	-10	-10	-29.0
Φορές που επιλέγηκε	2	5	74	22	17171	2	87	136	10
Κόστος βέλτιστης απόφασης	-10	-11	-17	-10	-10	-49	-10	-10	-29
Βρέθηκε η βέλτιστη απόφαση	☑	☑	☑	☑	✗	☑	☑	☑	☑

Συμπεράσματα αποτελεσμάτων:

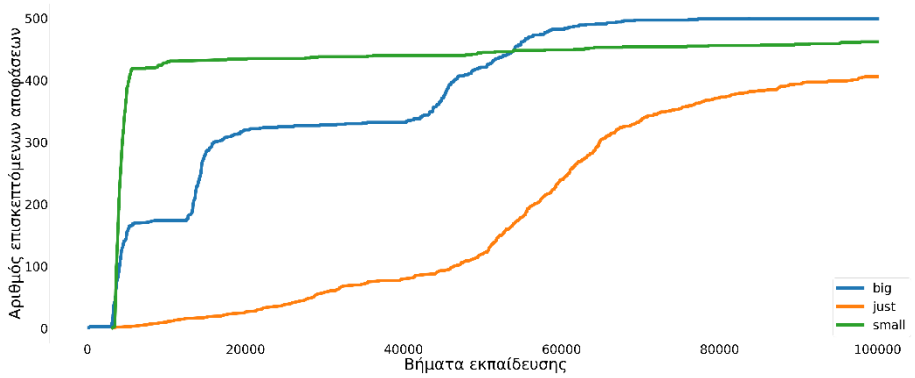
- **Πολιτικές**

Σύμφωνα με τα πιο πάνω αποτελέσματα μπορούμε να παρατηρήσουμε ότι την καλύτερη εξερεύνηση την έχει η Epsilon greedy πολιτική. Πολύ καλή εξερεύνηση έχει επίσης και η Boltzman πολιτική ενώ παρατηρούμε δυσκολία στην εξερεύνηση για την πολιτική Max Boltzman, με αρκετά μικρότερες τιμές στο πλήθος εξερευνημένων αποφάσεων σε σχέση με τες άλλες πολιτικές. Αποτέλεσμα τις μικρής εξερεύνησης είναι οι πολλές φορές που επιλέχτηκε η καλύτερη απόφαση στο Μικρό εύρος τιμών, με μεγάλη διαφορά από όλες τις υπόλοιπες τιμές και το γεγονός ότι δεν βρέθηκε ποτέ η βέλτιστη απόφαση. Σε αντίθεση με την εξερεύνηση η πολιτική Max Boltzman έχει τον μικρότερο αριθμό μη επιτρεπτών κινήσεων με εξαίρεση το Ακριβές εύρος τιμών. Τέλος όσων αφορά στην επιμονή επιλογής της βέλτιστης απόφασης βλέπουμε καλύτερα αποτελέσματα στην

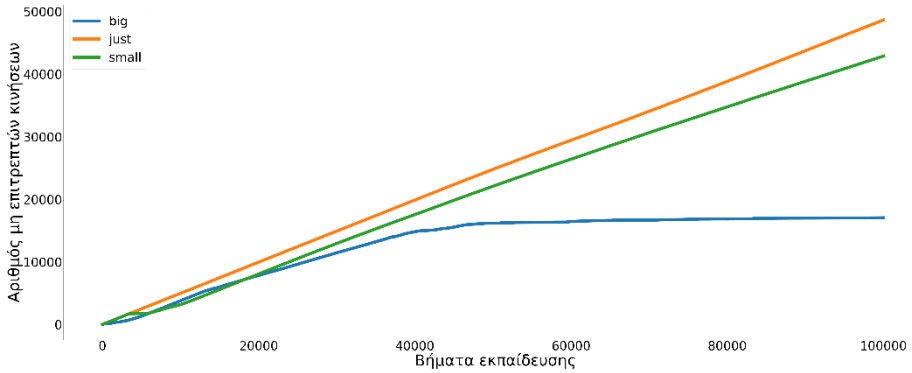
πολιτική Epsilon greedy για το Ακριβές και Μικρό εύρος τιμών ενώ στην πολιτική Boltzmann για το Μεγάλο εύρος τιμών.

- **Εύρος κόστους**

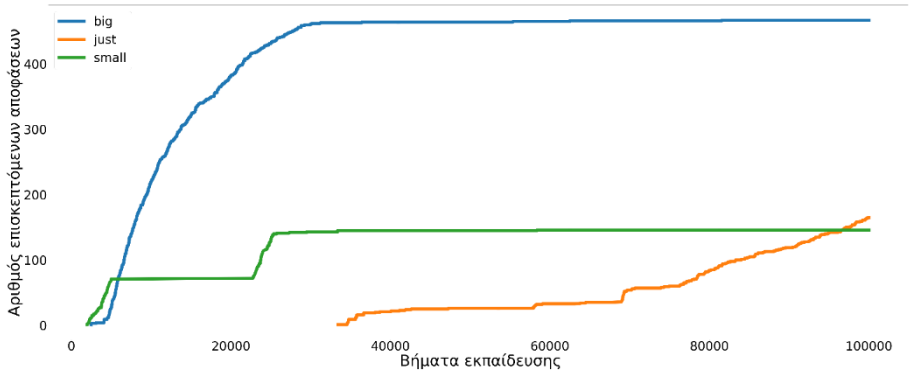
Τα αποτελέσματα επηρεάζονται και από το εύρος κόστους. Η μέση ανταμοιβή μειώνεται κάτι πολύ λογικό καθώς και τα κόστη αυξάνονται. Παρατηρούμε όμως και αύξηση στην εξερεύνηση καθώς το εύρος αυξάνεται όπως επίσης και αύξηση στον μέσο όρο βημάτων. Τέλος παρατηρούμε μείωση στην επιλογή λάθος κινήσεων, πιθανός στην μεγαλύτερη διαφορά που έχουν τα κόστη των ενεργειών με τις μη επιτρεπτές κινήσεις .



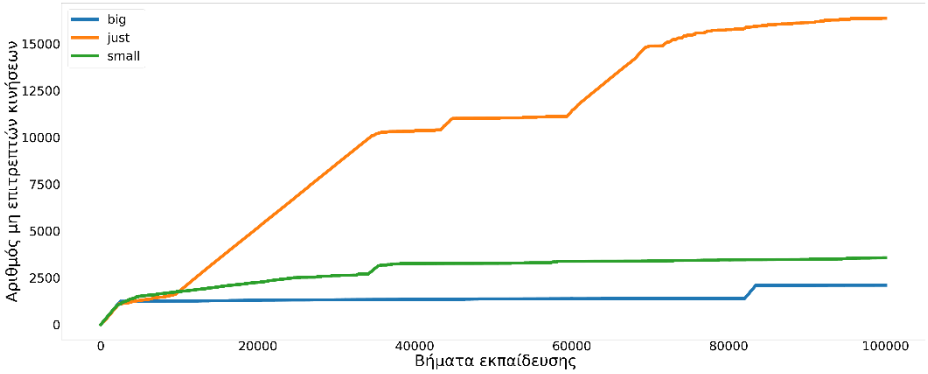
Εικόνα 5-11: Γραφική εξερεύνησης - Boltzmann



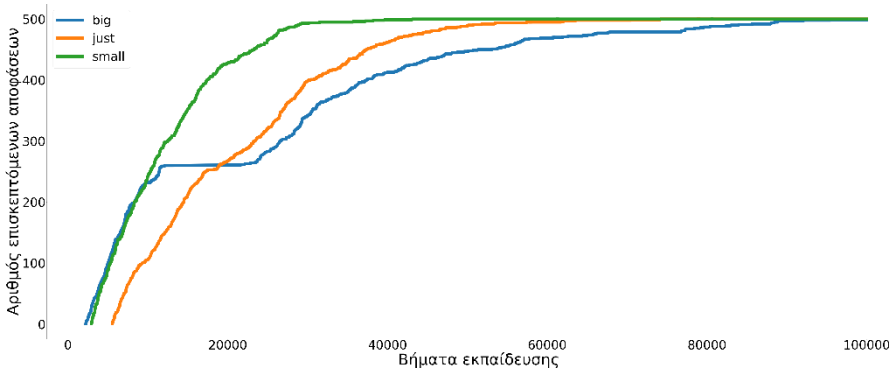
Εικόνα 5-12: Γραφική λάθος ενεργειών - Boltzmann



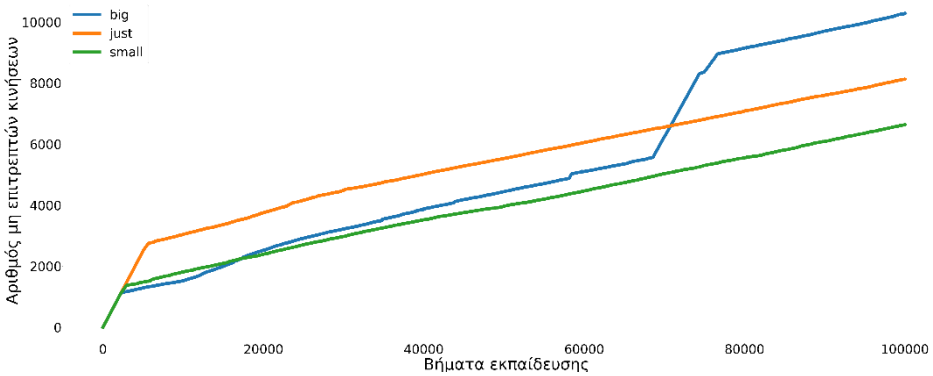
Εικόνα 5-13: Γραφική εξερεύνησης – Max Boltzmann



Εικόνα 5-14: Γραφική λάθος ενεργειών - Max Boltzmann



Εικόνα 5-15: Γραφική εξερεύνησης - Epsilon greedy



Εικόνα 5-16: Γραφική λάθος ενεργειών – Epsilon greedy

Συμπεράσματα γραφικών

Με τις γραφικές μπορούμε να επαληθεύσουμε όσα αναφέραμε και πιο πάνω. Η πολιτική Epsilon Greedy βλέπουμε ότι σχετικά γρήγορα εξερευνά πολλές αποφάσεις φτάνοντας στο τέλος να τις έχει επισκεφτεί σχεδόν όλες. Ακόμη μπούμε να διακρίνουμε την καλή εξερεύνηση της πολιτικής Boltzmann και την δυσκολία στην εξερεύνηση για την πολιτική Max Boltzmann. Επίσης διακρίνουμε μια αύξουσα τάση όσων αφορά τις μη επιτρεπτές κινήσεις για τις πολιτικές Epsilon greedy και Boltzmann ενώ μια κάπως σταθερότητα για την πολιτική Max Boltzmann . Ακόμα το ακριβές εύρος κόστους φαίνεται να περιορίζει την εξερεύνηση και να ωθεί στην εκτέλεση μη επιτρεπτών κινήσεων, αντιθέτως το μεγάλο εύρος κόστους περιορίζει την εκτέλεση μη επιτρεπτών κινήσεων και βοηθά στην εξερεύνηση. Το μικρό εύρος κόστους αν και δεν έχει ξεκάθαρη συμπεριφορά όσων αφορά την εκτέλεση μη επιτρεπτών κινήσεων, παρατηρούμε ότι σχετικά νωρίς σταμάτα την εξερεύνηση η οποία συνεχίζει με πολύ μικρό ρυθμό.

5.3.2 Πείραμα 2

Οι τιμές των παραμέτρων που χρησιμοποιήθηκαν στο συγκεκριμένο πείραμα για την εκπαίδευση του πράκτορα, φαίνονται στον Πίνακας 5-9 : Παράμετροι Πειράματος 2. Στο συγκεκριμένο πείραμα εξετάζουμε την απόδοση του μοντέλου μας σε ένα μεγάλο αριθμό διαθέσιμων αποφάσεων. Τα αποτελέσματα του πειράματος φαίνονται στον Πίνακας 5-10 : Αποτελέσματα Πειράματος .

Πίνακας 5-9 : Παράμετροι Πειράματος 2

Παράμετρος	**Τιμή**
Αριθμός διαθέσιμών ενεργειών	2000
Βήματα εκπαίδευσης	100 000
Εύρος κόστους	Ακριβές (Just)
	Μικρό (Small)
	Μεγάλο (Big)
Πολιτικές	Boltzmann Q policy
	Max Boltzman Q policy
	Epsilon greedy Q policy

Πίνακας 5-10 : Αποτελέσματα Πειράματος 2

Πολιτική	**Boltzmann**			**Max Boltzman**			**Epsilon greedy**		
Εύρος Κόστους	Ακριβές	Μικρό	Μεγάλο	Ακριβές	Μικρό	Μεγάλο	Ακριβές	Μικρό	Μεγάλο
Αριθμός μη επιτρεπτών κινήσεων	14583	34235	38427	29012	25333	27403	35166	36480	30265
Μέσος όρος αμοιβών	-212	-1099	-3987	-1038	-2326	-4076	-29.0	-2220	-18947
Μέσος όρος Βημάτων	2.0145	2.0134	2.0267	2.3188	2.0611	2.3029	2.4476	2.3914	2.1986
Αποφάσεις που εξερεύνησε	1452	916	497	308	991	68	1	371	56
Κόστος καλύτερης απόφασης	-11.0	-15.0	-20.0	-11.0	-12.0	-326.0	-29.0	-21.0	-851.0
Φορές που επιλεκτικέ	2	13	13	1	1	1	1	1	1
Κόστος βέλτιστης απόφασης	-10	-11	-11	-10	-12	-16	-10	-15	-14

Βρέθηκε η βέλτιστη απόφαση	✗	✗	✗	✗	☑	✗	✗	✗	✗

Συμπεράσματα:

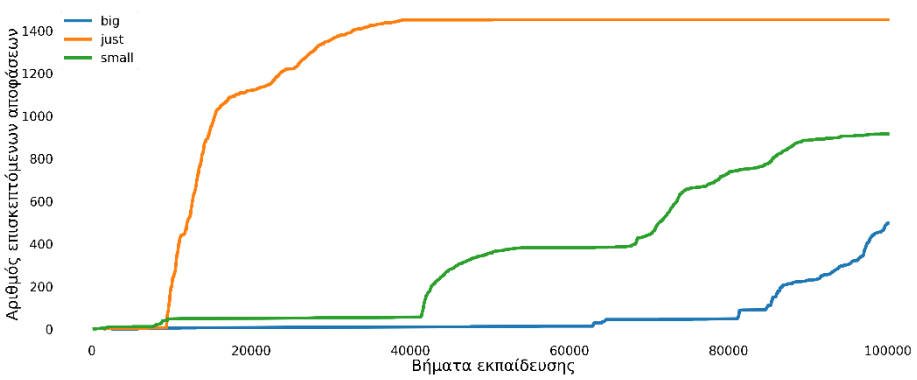
Αυξάνοντας τις ενέργειες που μπορεί να πάρει ο πράκτορας μπορούμε να διακρίνουμε καλύτερα τις διαφορές μεταξύ των πολιτικών και των ευρών κόστους .

- **Πολίτικες**

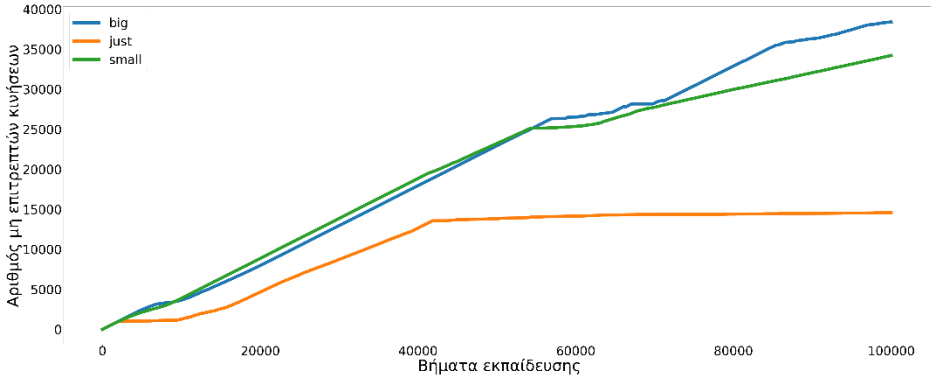
Από τις τιμές του πιο πάνω πίνακα μπορούμε να διακρίνουμε δυσκολία στις πολιτικές Max Boltzman και Epsilon Greedy για το ακριβές και μεγάλο εύρος κόστους και αδυναμία στην συνεχή επιλογή της καλύτερης ενέργειας που βρέθηκε. Επίσης η πολιτική Epsilon Greedy για το ακριβές εύρος τιμών κατάφερε να εκτελέσει μόνο μια επιτρεπτή κίνηση το οποίο είναι αποτέλεσμα της ανικανότητας της πολιτικής στο να εφαρμόσει μια αποδοτική στρατηγική για το περιβάλλον. Η πολιτική Max Boltzman ήταν η μόνη που κατάφερε να βρει την βέλτιστη ενέργεια στο μικρό εύρος τιμών έχοντας κάνει σχετικά καλή εξερεύνηση σε σχέση με τα άλλα δύο εύρη τιμών. Τέλος η πολιτική Boltzman έδειξε τα καλύτερα αποτελέσματα καθώς πέτυχε συνολικά την καλύτερη εξερεύνηση και έδειξε συνέχεια στην επιλογή των καλύτερων ενεργειών που βρήκε.

- **Εύρη Κόστους**

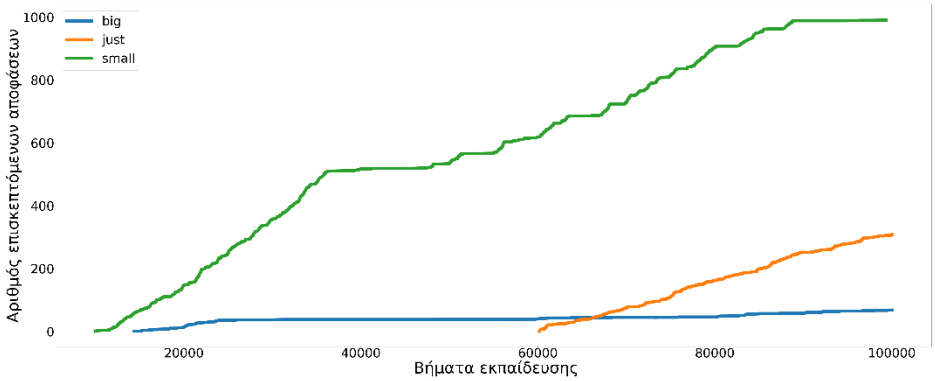
Όπως είναι λογικό όσο αυξάνεται το εύρος τιμών αυξάνεται και το κόστος της μέσης ανταμοιβής. Το εύρος κόστους επηρεάζει και τα βήματα ανά επεισόδιο καθώς βλέπουμε να αυξάνονται στην πολιτική Boltzman και να μειώνονται στην πολιτική Epsilon Greedy καθώς το εύρος αυξάνεται. Το εύρος τιμών φαίνεται να επηρεάζει και την εξερεύνηση με αποτέλεσμα όσο πιο μεγάλο να είναι τόσο πιο λίγη να είναι η εξερεύνηση. Εξαίρεση είναι όμως η εξερεύνηση των πολιτικών Max Boltzman και Epsilon Greedy για το μικρό εύρος τιμών όπου σημείωσαν αρκετά περισσότερη εξερεύνηση από τις υπόλοιπες περιπτώσεις.



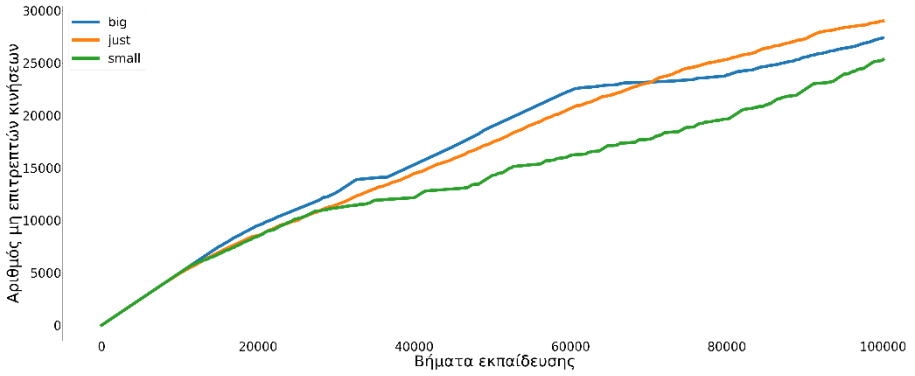
Εικόνα 5-17: Γραφική εξερεύνησης - Boltzmann

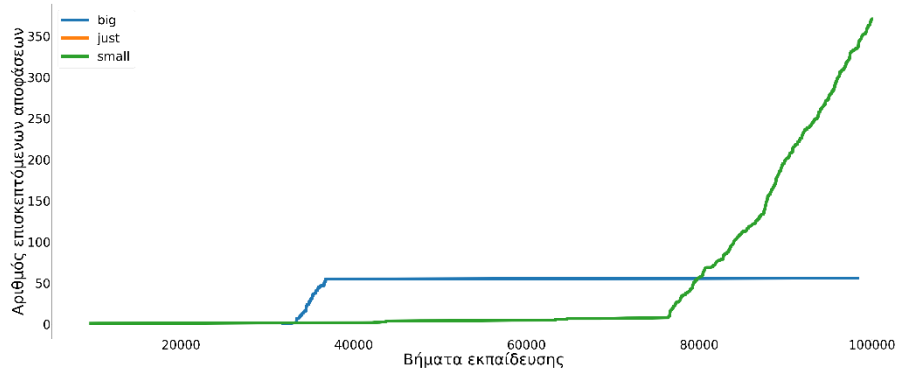


Εικόνα 5-18: Γραφική λάθος ενεργειών - Boltzmann

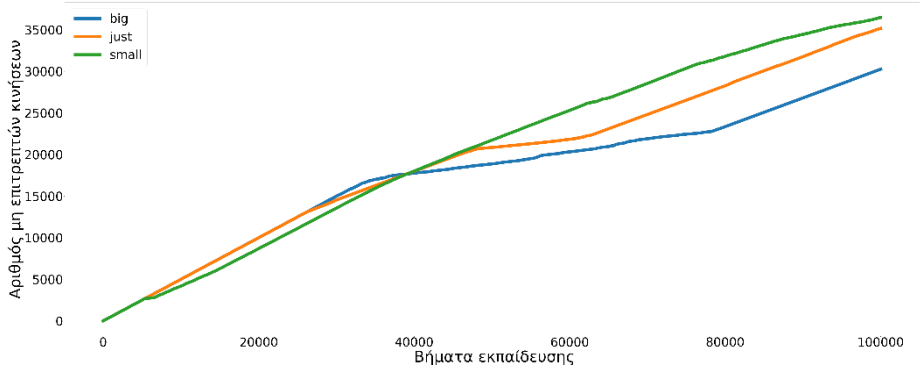


Εικόνα 5-19: Γραφική εξερεύνησης – Max Boltzmann





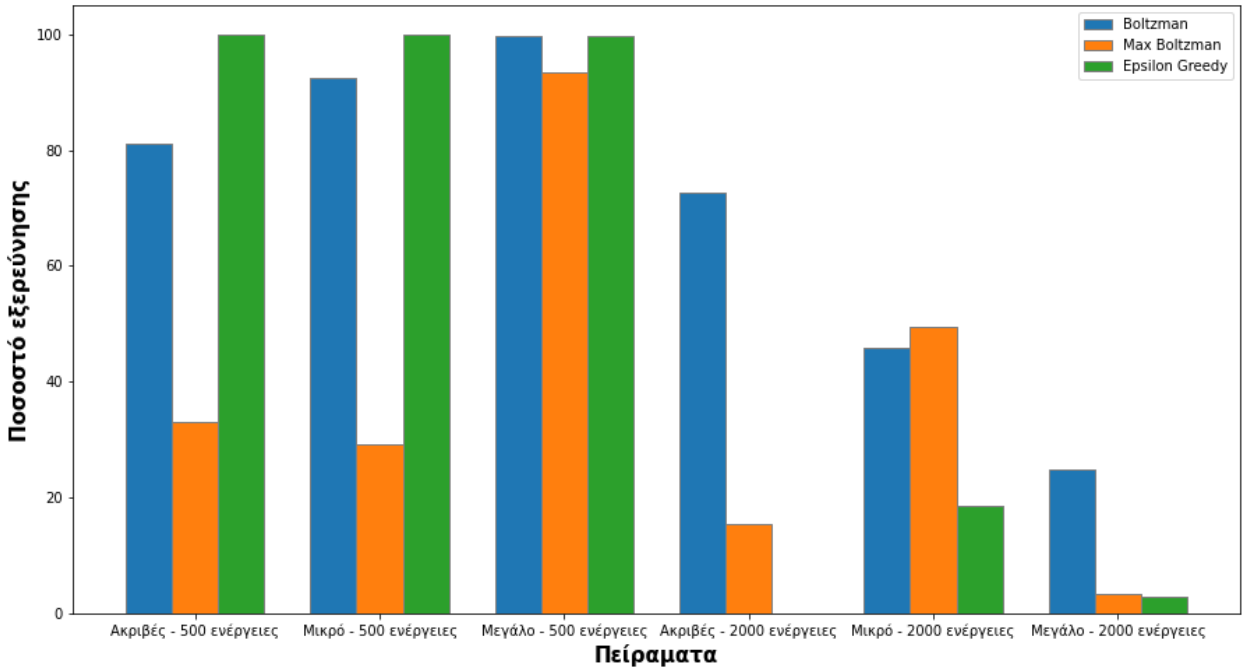
Εικόνα 5-21: : Γραφική εξερεύνησης - Epsilon Greedy



Εικόνα 5-22: Γραφική λάθος ενεργειών – Epsilon Greedy

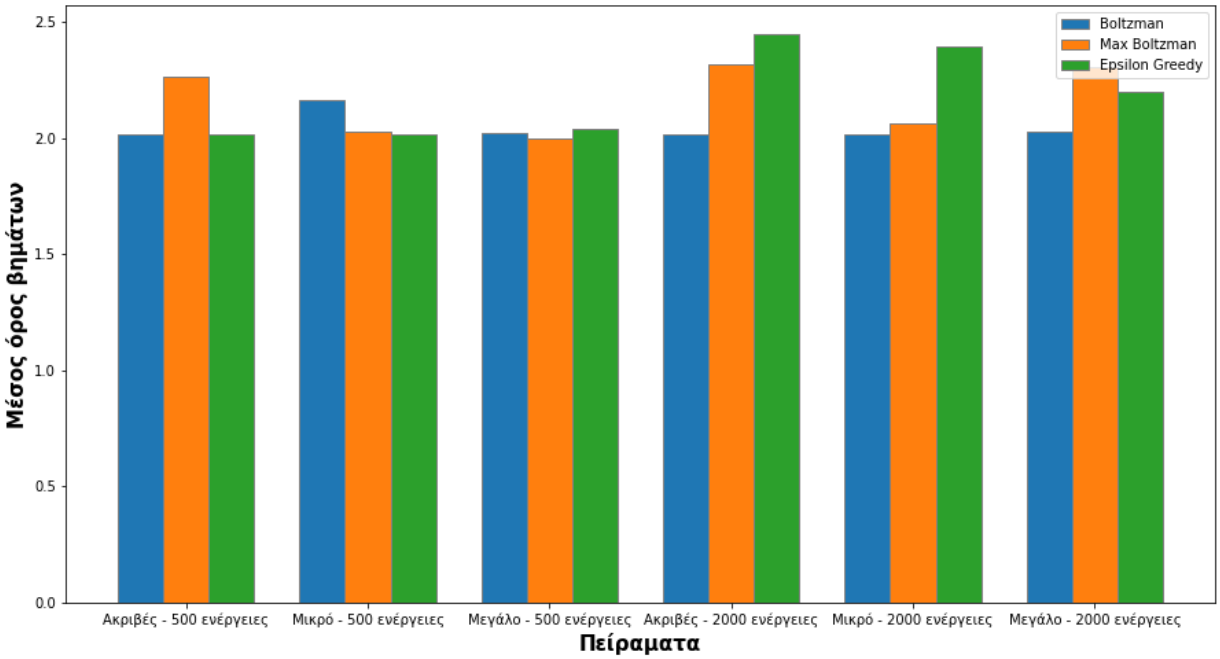
Από τις γραφικές μπορούμε να διακρίνουμε όσα παρατηρήσαμε και από τις τιμές του Πίνακας 5-10 : Αποτελέσματα Πειράματος 2. Με την πολιτική Boltzman παρατηρούμε η εξερεύνηση να γίνεται πιο γρήγορα άλλα και να σταθεροποιείται όταν οι τιμές των κοστών είναι κοντά μεταξύ τους και ταυτόχρονα μειώνει την εκτέλεση μη επιτρεπτών ενεργειών. Στις άλλες δύο πολιτικές παρατηρούμε δυσκολία στην εξερεύνηση και συνεχή επιλογή μη επιτρεπτών κινήσεων. Συγκεκριμένα και στις 2 πολιτικές Max Boltzman και Epsilon Greedy το εύρος κόστους δεν φαίνεται να επηρεάζει την εκτέλεση μη επιτρεπτών κινήσεων ενώ το μεγάλο και μικρό εύρος κόστους μειώνουν σημαντικά την εξερεύνηση. Επίσης για την πολιτική Max Boltzman παρατηρούμε ότι για το μικρό εύρος κόστους η εξερεύνηση αργεί ενώ από ένα σημείο και μετά αυξάνεται με μεγάλο ρυθμό.

Τα ποιο πάνω συμπεράσματα μπορούμε να τα διακρίνουμε πιο ξεκάθαρα στις πιο κάτω εικόνες όπου παρουσιάζουμε κάποιες από τις τιμές των πειραμάτων σε μορφή ραβδοδιαγράμματος.



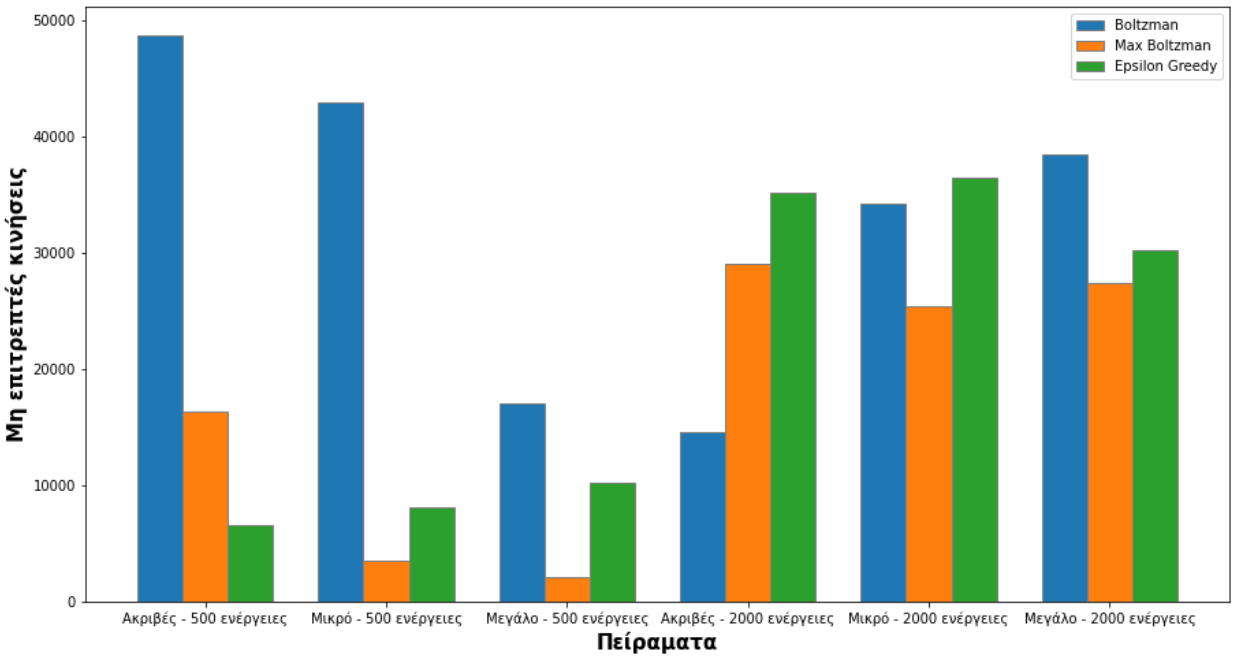
Εικόνα 5-23: Ποσοστά εξερεύνησης πειραμάτων

Στην Εικόνα 5-23 μπορούμε να διακρίνουμε ξεκάθαρα την δυσκολία της πολιτικής Max Boltzman για την εξερεύνηση του περιβάλλοντος. Εξαίρεση αποτελούν το μεγάλο εύρος τιμών για 500 ενέργειες και το μικρό εύρος τιμών για 2000 ενέργειες όπου εξερευνήθηκε ικανοποιητικά το περιβάλλον σε σχέση και με τις άλλες δύο πολιτικές. Η πολιτική Boltzman φαίνεται να έχει τα καλύτερα αποτελέσματα εξερεύνησης. Στα πειράματα με τις 500 ενέργειες έχει την ίδια περίπου εξερεύνηση με την πολιτική Epsilon Greedy, ενώ για τις 2000 ενέργειες έχει συνολικά την καλύτερη εξερεύνηση και από τις 2 άλλες πολιτικές. Ακόμη η δυσκολία της πολιτικής Epsilon Greedy για την εξερεύνηση στις 2000 ενέργειες πιθανών να οφείλεται στην μεγάλη τυχαιότητα επιλογής ενεργειών και την συχνή επιλογή μη επιτρεπτών κινήσεων λόγο του μεγάλου αριθμού ενεργειών. Όσων αφορά τα εύρη κοστών για τις 500 ενέργειες όσο μεγαλύτερη είναι η διαφορά των κότσων τόσο καλύτερη είναι και η εξερεύνηση. Αντίθετα στις 2000 ενέργειες η αύξηση του εύρους κόστους επηρεάζει αρνητικά την εξερεύνηση. Η μείωση στην εξερεύνηση στις 2000 ενέργειες πολύ πιθανών να οφείλεται στο ότι τα βήματα εκπαίδευσης που εκτελέστηκαν ήταν τα ίδια με τις 500 ενέργειες. Αυτό σημαίνει ότι για μεγαλύτερο αριθμό ενεργειών χρειάζεται περισσότερη εκπαίδευση κάτι το οποίο είναι αρκετά λογικό.



Εικόνα 5-24: Μέσος όρος βημάτων στα πειράματα

Παρατηρώντας την Εικόνα 5-24 μπορούμε να συμπεράνουμε ότι το εύρος κόστους δεν επηρεάζει πολύ τον μέσο όρο βημάτων. Αντιθέτως η αύξηση του αριθμού των ενεργειών αυξάνει τον μέσο όρο βημάτων για τις πολιτικές Max Boltzman και Epsilon Greedy. Αυτή η παρατήρηση είναι ένα καλό σημάδι ότι οι δύο αυτές πολιτικές αδυνατούν να κατανοήσουν και να μάθουν ορθά την λειτουργία του περιβάλλοντος σε σχέση με την πολιτική Boltzman. Σύμφωνα με την κατασκευή του περιβάλλοντος μας η βέλτιστη και γενικά οι επιτρεπτές κινήσεις έχουν ως αποτέλεσμα την εκτέλεση δυο βημάτων για την μετάβαση στην τελική κατάσταση. Άρα όσο πιο κοντά είναι ο μέσος όρος βημάτων στον αριθμό 2 τόσο καλύτερα αντιλαμβάνεται ορθά το περιβάλλον μας και ο πράκτορας. Η μεγαλύτερη τιμή στον μέσο όρο βημάτων για τις πολιτικές Max Boltzman και Epsilon Greedy πιθανών να είναι αποτέλεσμα την ανεπαρκής εξερεύνησης που αναφέραμε πιο πριν. Η πρώτη αδυνατεί να εξερεύνηση λόγο της επιμονής της στις καλύτερες ενέργειες που βρήκε οι οποίες πιθανών να εκτελούσαν περισσότερα από 2 βήματα. Η δεύτερη λόγο της τυχαιότητας στην επιλογή πολλές φορές επιλέγηκαν μη επιτρεπτές κινήσεις και κινήσεις με περισσότερα από 2 βήματα.



Εικόνα 5-25: Αριθμός μη επιτρεπτών κινήσεων στα πειράματα

Στην Εικόνα 5-25 διακρίνουμε τον αριθμό των μη επιτρεπτών κινήσεων που εκτελέστηκαν στα πειράματα. Καλό είναι να αναφέρουμε ότι οι μη επιτρεπτές κινήσεις είναι ένα χαρακτηριστικό της δικής μας υλοποίησης το οποίο θεωρητικά δεν θα έπρεπε να υπάρχει. Παρόλα αυτά μπορεί να χρησιμοποιηθεί για να συμπεράνουμε εάν οι αλγόριθμοι που χρησιμοποιούμε μπορούν να κατανοήσουν και να μάθουν την λειτουργεία του περιβάλλοντος μας. Αυτό επιτυγχάνεται με το μεγάλο κόστος που δίνουμε στις μη επιτρεπτές κινήσεις το οποίο οι πράκτορες μας προσπαθούν να αποφύγουν. Σύμφωνα με την Εικόνα 5-25 για τις 500 ενέργειες διακρίνουμε ένα μεγάλο αριθμό μη επιτρεπτών κινήσεων για την πολιτική Boltzman και μεγάλη διαφορά σε σχέση με τις υπόλοιπες πολιτικές. Όμως η διαφορά αυτή μειώνεται καθώς αυξάνεται το εύρος κόστους. Για τις 2000 ενέργειες η διαφορά μεταξύ του αριθμού μη επιτρεπτών κινήσεων στις τρείς πολίτικές δεν είναι πολύ μεγάλη, με εξαίρεση το ακριβές εύρος κόστους όπου η πολιτική Boltzman είχε αισθητά μικρότερο αριθμό. Για τις άλλες δύο πολιτικές παρατηρήθηκε σημαντική αύξηση στον αριθμό μη επιτρεπτών κινήσεων για τις 2000 ενέργειες. Όσων αφορά το εύρος κόστους για τις 500 ενέργειες η αύξηση του επηρεάζει θετικά τις πολιτικές Boltzman και Max Boltzman καθώς μειώνει τον αριθμό μη επιτρεπτών κινήσεων. Η αύξηση του εύρους κόστους αυξάνει και τον αριθμό των μη επιτρεπτών κινήσεων για την πολιτική Epsilon Greedy. Αντίθετα όμως στις 2000 ενέργειες η αύξηση του εύρους κόστους επηρεάζει αρνητικά την πολιτική Boltzman αφού αυξάνεται και ο αριθμός των μη επιτρεπτών κινήσεων, ενώ δεν φαίνεται να επηρεάζει ιδιαίτερα τις άλλες δυο πολίτικές.

6 Συμπεράσματα και Μελλοντική Εργασία

6.1 Σύνοψη και Συμπεράσματα

Στην παρούσα διπλωματική εργασία προτείνεται η υλοποίηση ενός μοντέλου για την λήψη επιχειρηματικών αποφάσεων με χρήση βαθιάς ενισχυτικής μάθησης. Αρχικά, μοντελοποιούμε κατάλληλα το πρόβλημα έτσι ώστε να εφαρμόσουμε ενισχυτική μάθηση σε αυτό. Η μοντελοποίηση αυτή έγινε χρησιμοποιώντας MDPs, οι οποίες είναι ένας τυποποιημένος τρόπος έκφρασης προβλημάτων απόφασης. Η χρήση της ενισχυτικής μάθησης προϋποθέτει την ύπαρξη περιβάλλοντος και πράκτορα τα οποία κατασκευάσαμε βασισμένα στην μοντελοποίηση που πραγματοποιήσαμε. Η υλοποίηση του περιβάλλοντος και του πράκτορα έγινε με την βοήθεια των OpenAI gym και Keras, τα οποία μας παρέχουν αλγορίθμους για την εφαρμογή ενισχυτικής μάθησης . Σημαντική ιδιότητα του περιβάλλοντος που υλοποιήσαμε είναι ότι δεν είναι σταθερό αλλά κατασκευάζεται δυναμικά αναλόγως με το πρόβλημα που καλείτε το μοντέλο μας να επιλύσει. Συγκεκριμένα, το περιβάλλον κατασκευάζεται σύμφωνα με το πλήθος των αποφάσεων που υπάρχουν στο πρόβλημα. Ακόμη η επίλυση του προβλήματος βασιζόταν στην εκπαίδευση του πράκτορα για την επίλυση του προβλήματος, ο οποίος αλληλοεπιδρούσε με το περιβάλλον που κατασκευάσαμε και σκοπός του ήταν να βρεθεί η βέλτιστη απόφαση. Η εκπαίδευση του πράκτορα έγινε με χρήση βαθιάς ενισχυτικής μάθησης με την χρήση του αλγόριθμου DQN και ενός νευρωνικού δικτύου.

Για την αξιολόγηση του μοντέλου μας χρησιμοποιήσαμε μετρικές του νευρωνικού, του πράκτορα και τιμές που αφορούν την συμπεριφορά του πράκτορα κατά την διάρκεια της εκπαίδευσης. Αρχικά, εκτελέσαμε διάφορες εκπαιδεύσεις του πράκτορα αλλάζοντας παραμέτρους στο νευρωνικό δίκτυο. Σκοπός ήταν να παρατηρήσουμε ποιες παράμετροι του νευρωνικού είχαν τα καλύτερα αποτελέσματα για το πρόβλημα μας. Αφού παρατηρήσαμε ποιες παράμετροι ταιριάζουν καλύτερα στο πρόβλημα μας, εκτελέσαμε περαιτέρω εκπαιδεύσεις για να αξιολογήσουμε συνολικά το μοντέλο μας. Στις εκπαιδεύσεις αυτές κάναμε αλλαγές σε διάφορες παραμέτρους του προβλήματος, του περιβάλλοντος μας δηλαδή, και του πράκτορα. Αναλυτικά οι παράμετροι που ελέγξαμε ήταν ο αριθμός των αποφάσεων του

προβλήματος, τα κόστη των αποφάσεων και η πολιτική που χρησιμοποιούσε ο πράκτορας μας. Στο Κεφάλαιο 5 παρουσιάζονται συνοπτικά κάποιες από τις εκπαιδεύσεις αυτές σε μορφή δύο πειραμάτων.

Από τα αποτελέσματα των πειραμάτων μπορέσαμε να παρατηρήσουμε πως οι παράμετροι του μοντέλου μας επηρεάζουν την απόδοση του. Οι παράμετροι που έδειξαν την μεγαλύτερη επιρροή στην απόδοση του μοντέλου μας ήταν ο αριθμός των αποφάσεων (ενεργειών) και η πολιτική που χρησιμοποιούσε ο πράκτορας. Για μεγαλύτερο αριθμό αποφάσεων παρατηρήθηκε και μεγαλύτερη δυσκολία επίλυσης του προβλήματος, καθώς τα ποσοστά εξερεύνησης ήταν πιο χαμηλά και σε σχεδόν καμία εκπαίδευση δεν βρέθηκε η βέλτιστη λύση. Αυτό μας οδηγεί στο συμπέρασμα ότι καθώς αυξάνονται οι αποφάσεις χρειάζεται και περισσότερη εκπαίδευση του πράκτορα. Οι πολιτικές με τη σειρά τους έδειξαν διαφορετικές συμπεριφορές και αποτελέσματα. Η πολιτική Epsilon Greedy έδειξε καλή εξερεύνηση για τις 500 αποφάσεις αλλά δυσκολία στην εξερεύνηση στις 2000 αποφάσεις. Η πολιτική Max Boltzman αν και έδειξε επιμονή στην επιλογή τον καλύτερων αποφάσεων που βρήκε αυτό είχε σαν αποτέλεσμα να μην εξερεύνα ικανοποιητικά τις υπόλοιπες αποφάσεις. Η πολιτική Boltzman είχε συνολικά τα καλύτερα αποτελέσματα με εμφανή διαφορά από τις υπόλοιπες πολιτικές για τις εκπαιδεύσεις των 2000 αποφάσεων. Επιπρόσθετα, το εύρος κόστους των αποφάσεων επηρέαζε διαφορετικά κάθε πολιτική και αριθμό αποφάσεων. Γενικά η αύξηση του εύρους κόστους επηρέαζε θετικά τις εκπαιδεύσεις των 500 αποφάσεων. Αντιθέτως η αύξηση του στις 2000 αποφάσεις επηρέαζε αρνητικά τις εκπαιδεύσεις.

6.2 Μελλοντική Έρευνα

Όπως έχουμε αναφέρει κύριος σκοπός αυτής της διπλωματικής είναι η μοντελοποίηση επιχειρηματικών αποφάσεων με χρήση βαθιάς ενισχυτικής μάθησης. Το σημαντικό στην υλοποίηση που κάναμε για το μοντέλο μας είναι η δυναμική δημιουργία του περιβάλλοντος που αναπαριστά το συγκεκριμένο πρόβλημα απόφασης που καλείτε η βαθιά ενισχυτική μάθηση να λύσει. Από τα αποτελέσματα των πειράματα μας μπορούμε να συμπεράνουμε ότι το μοντέλο μας λειτουργά ικανοποιητικά και καταφέρνει να επιλύσει το πρόβλημα βρίσκοντας την βέλτιστη απόφαση.

Μελλοντικά, για την βελτίωση της επίδοσης του μοντέλου της συγκεκριμένης διπλωματικής εργασίας θα μπορούσε να γίνει περισσότερη παραμετροποίηση. Αυτό οφείλεται στο γεγονός ότι οι παράμετροι που ελέγξαμε για τα πειράματα μας αποτελούν ένα μικρό αριθμό από το σύνολο των παραμέτρων που μπορούν να αλλαχτούν και να παραμετροποιηθούν κατάλληλα. Οι παράμετροι αυτοί μπορεί να αφορούν τα χαρακτηριστικά του πράκτορα και της

ενισχυτικής μάθησης που χρησιμοποιούμε αλλά και του περιβάλλοντος που κατασκευάζεται. Επίσης περισσότερα βήματα στις εκπαιδεύσεις θα βοηθήσουν την απόδοση του μοντέλου μας σε μεγαλύτερο αριθμό αποφάσεων.

Η παρούσα διπλωματική εργασία μπορεί να επεκταθεί σε πολλές κατευθύνσεις. Συγκεκριμένα στη χρήση της βαθιάς ενισχυτικής μάθησης θα μπορούσαν να εφαρμοστούν διαφορετικές μέθοδοι και αλγόριθμοι επίλυσης. Παραδείγματα μεθόδων θα μπορούσαν να είναι αυτές του Δράστη- Αξιολογιτή (Actor – Critic) [25] ενώ για αλγορίθμους θα μπορούσαν να χρησιμοποιηθούν παραλλαγές του DQN όπως οι αλγόριθμοι DDQN (Double DQN) [26] και Dueling DQN [27] αλλά και άλλοι πολλοί αλγόριθμοι βαθιάς ενισχυτικής μάθησης. Επίσης, ακόμη μια επέκταση είναι η τροποποίηση του μοντέλου μας έτσι ώστε να μην επιλύει μόνο ένα δυναμικά ορισμένο περιβάλλον αλλά και ένα δυναμικά ορισμένο κόστος αποφάσεων. Αρκετές προσπάθειες έχουν γίνει για την επίλυση τέτοιων προβλημάτων ειδικά για την σύσταση σε χρήστες και την πλοήγηση σε περιβάλλοντα [28] [29].

Βιβλιογραφία

[1] Y. Li, "Deep Reinforcement Learning: An Overview," *ArXiv170107274 Cs*, Nov. 2018, Accessed: Jul. 04, 2021. [Online]. Available: http://arxiv.org/abs/1701.07274

[2] B. Mirchevska, C. Pek, M. Werling, M. Althoff, and J. Boedecker, "High-level Decision Making for Safe and Reasonable Autonomous Lane Changing using Reinforcement Learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2018, pp. 2156–2162. doi: 10.1109/ITSC.2018.8569448.

[3] S. M. Shortreed, E. Laber, D. J. Lizotte, T. S. Stroup, J. Pineau, and S. A. Murphy, "Informing sequential clinical decision-making through reinforcement learning: an empirical study," *Mach. Learn.*, vol. 84, no. 1–2, pp. 109–136, Jul. 2011, doi: 10.1007/s10994-010-5229-0.

[4] T. J. Loftus *et al.*, "Decision analysis and reinforcement learning in surgical decision-making," *Surgery*, vol. 168, no. 2, pp. 253–266, Aug. 2020, doi: 10.1016/j.surg.2020.04.049.

[5] R. S. Sutton and A. G. Barto, *Reinforcement Learning, second edition: An Introduction*. MIT Press, 2018.

[6] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A Survey," *arXiv:cs/9605103*, Apr. 1996, Accessed: Jun. 25, 2021. [Online]. Available: http://arxiv.org/abs/cs/9605103

[7] "Part 2: Kinds of RL Algorithms — Spinning Up documentation." https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html#citations-below (accessed Jul. 06, 2021).

[8] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of Markov Chain Monte Carlo*. CRC Press, 2011.

[9] C. Szepesvári, "Algorithms for Reinforcement Learning," *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 4, no. 1, pp. 1–103, Jan. 2010, doi: 10.2200/S00268ED1V01Y201005AIM009.

[10] T. Jaakkola, S. P. Singh, and M. I. Jordan, "Reinforcement Learning Algorithm for Partially Observable Markov Decision Problems," p. 8.

[11] "Bellman equation - Wikipedia." https://en.wikipedia.org/wiki/Bellman_equation (accessed Jun. 30, 2021).

[12] C. Dann, G. Neumann, and J. Peters, "Policy Evaluation with Temporal Differences: A Survey and Comparison," *J. Mach. Learn. Res.*, vol. 15, pp. 809–883, Mar. 2014.

[13] J. Groot Kormelink, M. M. Drugan, and M. A. Wiering, "Exploration Methods for Connectionist Q-learning in Bomberman:," in *Proceedings of the 10th International Conference on Agents and Artificial Intelligence*, Funchal, Madeira, Portugal, 2018, pp. 355–362. doi: 10.5220/0006556403550362.

[14] L. Pan, Q. Cai, Q. Meng, W. Chen, L. Huang, and T.-Y. Liu, "Reinforcement Learning with Dynamic Boltzmann Softmax Updates," *ArXiv190305926 Cs Stat*, Sep. 2019, Accessed: Jun. 30, 2021. [Online]. Available: http://arxiv.org/abs/1903.05926

[15] B. O'Donoghue, R. Munos, K. Kavukcuoglu, and V. Mnih, "Combining policy gradient and Q-learning," *ArXiv161101626 Cs Math Stat*, Apr. 2017, Accessed: Jun. 30, 2021. [Online]. Available: http://arxiv.org/abs/1611.01626

[16] C. J. C. H. Watkins and P. Dayan, "Technical Note: Q-Learning," *Mach. Learn.*, vol. 8, no. 3, pp. 279–292, May 1992, doi: 10.1023/A:1022676722315.

[17] "AlphaGo: The story so far," *Deepmind*. https://deepmind.com/research/case-studies/alphago-the-story-so-far (accessed Jul. 02, 2021).

[18] V. Mnih *et al.*, "Playing Atari with Deep Reinforcement Learning," *ArXiv13125602 Cs*, Dec. 2013, Accessed: Jul. 02, 2021. [Online]. Available: http://arxiv.org/abs/1312.5602

[19] S. S. Haykin and S. S. Haykin, *Neural networks and learning machines*, 3rd ed. New York: Prentice Hall, 2009.

[20] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, “Resource Management with Deep Reinforcement Learning,” in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, Atlanta GA USA, Nov. 2016, pp. 50–56. doi: 10.1145/3005745.3005750.
[21] OpenAI, “Gym: A toolkit for developing and comparing reinforcement learning algorithms.” https://gym.openai.com (accessed Jun. 26, 2021).
[22] “Keras: the Python deep learning API.” https://keras.io/ (accessed Jul. 09, 2021).
[23] L. Metz, J. Ibarz, N. Jaitly, and J. Davidson, “Discrete Sequential Prediction of Continuous Actions for Deep RL,” *ArXiv170505035 Cs Stat*, Jun. 2019, Accessed: Jul. 07, 2021. [Online]. Available: http://arxiv.org/abs/1705.05035
[24] Z. Zhang and M. Sabuncu, “Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels,” p. 11.
[25] “TensorBoard | TensorFlow.” https://www.tensorflow.org/tensorboard (accessed Jul. 08, 2021).
[26] L. Xi, L. Yu, Y. Xu, S. Wang, and X. Chen, “A Novel Multi-Agent DDQN-AD Method-Based Distributed Strategy for Automatic Generation Control of Integrated Energy Systems,” *IEEE Trans. Sustain. Energy*, vol. 11, no. 4, pp. 2417–2426, Oct. 2020, doi: 10.1109/TSTE.2019.2958361.
[27] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, “Dueling Network Architectures for Deep Reinforcement Learning,” *ArXiv151106581 Cs*, Apr. 2016, Accessed: Jul. 08, 2021. [Online]. Available: http://arxiv.org/abs/1511.06581
[28] S.-Y. Chen, Y. Yu, Q. Da, J. Tan, H.-K. Huang, and H.-H. Tang, “Stabilizing Reinforcement Learning in Dynamic Environment with Application to Online Recommendation,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA, Jul. 2018, pp. 1187–1196. doi: 10.1145/3219819.3220122.
[29] M. A. Kareem Jaradat, M. Al-Rousan, and L. Quadan, “Reinforcement based mobile robot navigation in dynamic environment,” *Robot. Comput.-Integr. Manuf.*, vol. 27, no. 1, pp. 135–149, Feb. 2011, doi: 10.1016/j.rcim.2010.06.019.