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

Ενισχυτική Μάθηση και Αλγοριθμικές Συναλλαγές στο Χρηματιστήριο με την Τεχνική του Q-Learning

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΚΩΝΣΤΑΝΤΙΝΟΣ ΣΚΟΥΡΑΣ

Επιβλέπων : Γιώργος Στάμου
Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2019

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

Ενισχυτική Μάθηση και Αλγοριθμικές Συναλλαγές στο Χρηματιστήριο με την Τεχνική του Q-Learning

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΚΩΝΣΤΑΝΤΙΝΟΣ ΣΚΟΥΡΑΣ

Επιβλέπων : Γιώργος Στάμου
Αν. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 29η Οκτωβρίου 2019.

.....	.....	.....
Γιώργος Στάμου Αν. Καθηγητής Ε.Μ.Π.	Ανδρέας-Γεώργιος Σταφυλοπάτης Καθηγητής Ε.Μ.Π.	Νικόλαος Σ. Παπασπύρου Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2019

Κωνσταντίνος Σκούρας

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

Copyright © Κωνσταντίνος Σκούρας, 2019.
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Στη σημερινή εποχή, η Τεχνητή Νοημοσύνη χρησιμοποιείται ευρέως στον Χρηματοοικονομικό τομέα και ιδιαίτερα στις Συναλλαγές. Πολλές επενδυτικές Εταιρίες στην προσπάθειά τους να αξιοποιήσουν καλύτερα την πληθώρα δεδομένων της αγοράς, αλλά και να βελτιώσουν τα αποτελέσματά τους στο Χρηματιστήριο, κατασκευάζουν μοντέλα Τεχνητής Νοημοσύνης και τα εντάσσουν στην ροή εργασίας τους.

Η Ενισχυτική Μάθηση, είναι ένας τύπος μηχανικής μάθησης, στο οποίο ένα τεχνητό σύστημα δρα σαν υπεύθυνος λήψης αποφάσεων μέσα σε ένα Περιβάλλον. Το σύστημα αυτό, ονομάζεται Πράκτορας και αποφασίζει για τις δράσεις του, βλέποντας τα αποτελέσματα των προηγούμενων δράσεών του και αποκομίζοντας ανταμοιβές για αυτές. Αρκετή έρευνα γίνεται στον συγκεκριμένο τομέα και πολλά επιτυχή συστήματα έχουν δημιουργηθεί, όπως τα AlphaGo και AlphaZero της DeepMind που διαγωνίστηκαν κερδίζοντας πρωταθλητές στο παιχνίδι Go ή πιο πρόσφατα το ALphaStar που πέτυχε εξαιρετικά αποτελέσματα στο StarCraft 2.

Στην παρούσα εργασία, γίνεται μία μελέτη για την Εφαρμογή της Ενισχυτικής Μάθησης στις Συναλλαγές στο Χρηματιστήριο. Συγκεκριμένα, πρώτα υλοποιήθηκαν διαφορετικοί Πράκτορες, οι οποίοι κάνουν χρήση των αλγορίθμων Q Network, Deep Q Network, Double και Dueling Q Network, αλγόριθμοι Ενισχυτικής Μάθησης που βασίζονται στην τεχνική του Q-Learning. Στην συνέχεια, σχεδιάστηκε και υλοποιήθηκε το Περιβάλλον, το οποίο θα προσομοιώσει την αγορά του Χρηματιστηρίου. Ένα από τα πιο σημαντικά μέρη της εργασίας αποτελούν επίσης, οι συναρτήσεις επιβράβευσης, οι οποίες κατέχουν μείζονα ρόλο, στον τρόπο με τον οποίο συμπεριφέρεται ένα Πράκτορας. Στα πλαίσια της εργασίας δημιουργήθηκαν τρεις διαφορετικές συναρτήσεις επιβράβευσης με βάση τις οποίες οι παραπάνω Πράκτορες διαμόρφωσαν την στρατηγική τους, για την πραγματοποίηση επενδύσεων στο Περιβάλλον. Τέλος, οι Πράκτορες καθώς και οι στρατηγικές που δημιούργησαν, αξιολογήθηκαν ως προς τα κέρδη που κατάφεραν να αποκομίσουν.

Λέξεις κλειδιά

Τεχνητή Νοημοσύνη, Μηχανική Μάθηση, Ενισχυτική Μάθηση, Χρηματιστήριο, Συναλλαγές, Ευφυής Πράκτορες, Q-Learning

Abstract

In today's era, Artificial Intelligence is widely used in the Finance sector and especially in Trading. Many investment companies try to make the most of their market data and optimize their results on the stock market, by building artificial intelligence models and integrating them into their workflows.

Reinforcement Learning is a type of machine learning in which an artificial system acts as a decision maker in an environment. This system is called Agent and decides on its actions, by seeing the results of its previous actions and the rewards obdained. A lot of research is being done in this area and many successful systems have been developed, such as DeepMind's AlphaGo and AlphaZero which competed and won champions in the game Go or more recently ALphaStar which has achieved great results in StarCraft 2.

In the present thesis, a study is carried out on applying Reinforcement Learning in Stock Trading. In more specific, different Agents were implemented, using Q Network, Deep Q Network, Double and Dueling Q Network algorithms, Q-Learning-based Reinforcement Learning algorithms. Subsequently, the Environment was designed and implemented to simulate the stock market. One of the most important parts of the work preseneted is the reward functions, which play a major role in the way an Agent behaves. As part of the thesis, three different reward functions were created on the basis of which the above Agents formulated their strategy for investing in the Environment. Finally, the Agents as well as the strategies they created were evaluated for the profits they were able to make.

Key words

Artificial Intelligence, Machine Learning, Reinforcement Learning, Stock Market, Trading, Intelligent Agents, Q-Learning

Ευχαριστίες

Θα ήθελα να ευχαριστήσω, τον επιβλέπων καθηγητή μου κ. Στάμου για τις πολύτιμες γνώσεις που μου προσέφερε όλα αυτά τα χρόνια πάνω στον Προγραμματισμό, την Τεχνητή Νοημοσύνη και την Επιστήμη των Υπολογιστών γενικότερα. Ευχαριστώ θερμά, την κ. Νατάσα Σοφού για την καθοδήγησή της και τις στοχευμένες παρατηρήσεις της, που αποτέλεσαν σημαντικό παράγοντα για την διεκπεραίωση της παρούσας εργασίας. Ευχαριστώ επίσης, τον κ. Παπασπύρου και κ. Σταφυλοπάτη καθώς και τους υπόλοιπους καθηγητές του ΕΜΠ, οι οποίοι μέσα από την διδασκαλία τους συνέβαλλαν καθοριστικά στην εξέλιξή μου. Επιπρόσθετα, θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου για την στήριξή τους σε όλο το διάστημα της μέχρι τώρα ακαδημαϊκής μου πορείας.

Κωνσταντίνος Σκούρας,

Αθήνα, 29η Οκτωβρίου 2019

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

Κεφάλαιο 1

Εισαγωγή

1.1 Σκοπός της εργασίας

Σκοπός της συγκεκριμένης εργασίας, είναι η εκπαίδευση ενός τεχνητού συστήματος (Πράκτορας) με την τεχνική της Ενισχυτικής Μάθησης, ώστε να μάθει να πραγματοποιεί συναλλαγές στο Χρηματιστήριο. Με απλά λόγια, το σύστημα αυτό θα λαμβάνει αποφάσεις σχετικά με το πότε θα πρέπει να πουλήσει ή να αγοράσει μετοχές χωρίς να μεσολαβήσει ο ίδιος ο επενδυτής. Για την επίτευξη του σκοπού αυτού, γίνεται μία μελέτη διαφόρων συναρτήσεων επιβράβευσης, που καθορίζουν τον τρόπο με τον οποίο ο Πράκτορας θα λαμβάνει αποφάσεις για την πραγματοποίηση των συναλλαγών του. Κατόπιν του σχεδιασμού και υλοποίησης ενός Τεχνητού Περιβάλλοντος, το οποίο προσημειώνει την Αγορά του Χρηματιστηρίου καθώς και αλγορίθμων βασισμένους στην τεχνική του Q-Learning, γίνεται μία αξιολόγηση των Πρακτόρων και των διαφορετικών στρατηγικών που δημιούργησαν με βάση τα κέρδη που κατάφεραν να αποκομίσουν.

1.2 Αλγοριθμικές Συναλλαγές

Στη σημερινή εποχή, η Τεχνητή Νοημοσύνη χρησιμοποιείται ευρέως στον Χρηματοοικονομικό τομέα και ιδιαίτερα στις Συναλλαγές. Πολλές εταιρίες στην προσπάθειά τους να αξιοποιήσουν καλύτερα την πληθώρα δεδομένων της αγοράς, αλλά και να βελτιώσουν τα αποτελέσματά τους στο Χρηματιστήριο, κατασκευάζουν μοντέλα Τεχνητής Νοημοσύνης και τα εντάσσουν στην ροή εργασίας τους.

Μία ροή εργασίας, λοιπόν, για την δημιουργία μίας στρατηγικής για χρηματιστηριακές συναλλαγές, συνήθως περιλαμβάνει πολλά στάδια, στα οποία συμμετέχουν τόσο άνθρωποι, όσο και μοντέλα Τεχνητής Νοημοσύνης. Για παράδειγμα, μία τυπική ροή, θα ήταν αρχικά να μελετήσει ένας επενδυτής τα δεδομένα της αγοράς ώστε να βρει μετοχές για να επενδύσει, να αναπτύξει ένα μοντέλο Τεχνητής Νοημοσύνης, συνήθως ένα μοντέλο εποπτευόμενης μάθησης, ώστε να προβλέψει τις μελλοντικές τιμές τους ή άλλους δείκτες που μπορεί να τον ενδιαφέρουν και να δημιουργήσει την στρατηγική του σχετικά με το πότε θα πρέπει να πραγματοποιήσει τις αγοροπωλησίες του. Φυσικά, επειδή η αγορά αλλάζει ραγδαία, θα πρέπει να επαναλαμβάνει τα παραπάνω χρονοβόρα βήματα αρκετά συχνά, ώστε να προσαρμόζεται στις αλλαγές και να διατηρεί μία κερδοφόρα στρατηγική.

1.3 Ενισχυτική Μάθηση

Η Ενισχυτική Μάθηση, είναι ένας τύπος μηχανικής μάθησης, όπου ένας Πράκτορας (Agent), μαθαίνει πώς να συμπεριφέρεται μέσα σε ένα περιβάλλον (Enviroment), λαμβάνοντας δράσεις (Actions) σε αυτό, βλέποντας τα αποτελέσματα των δράσεών του (states) και αποκομίζοντας ανταμοιβές (Rewards) για τις δράσεις του. Αρκετή έρευνα γίνεται στον συγκεκριμένο τομέα και πολλά επιτυχή συστήματα έχουν δημιουργηθεί, όπως τα AlphaGo και AlphaZero της DeepMind που διαγωνίστηκαν κερδίζοντας πρωταθλητές στο παιχνίδι Go ή πιο πρόσφατα το ALphaStar που πέτυχε εξαιρετικά αποτελέσματα στο παιχνίδι StarCraft 2.

Χρησιμοποιώντας την Ενισχυτική μάθηση για Χρηματιστηριακές Συναλλαγές, μπορούν να επιλυθούν πολλά από τα προβλήματα της ροής εργασίας που αναφέρθηκαν παραπάνω. Συγκεκριμένα, ένας Πράκτορας θα δρα πλέον σε ένα Χρηματιστηριακό περιβάλλον, που θα περιλαμβάνει το ιστορικό των τιμών της μετοχής που επενδύει καθώς και άλλες μεταβλητές χρήσιμες για τον ίδιο, όπως το διαθέσιμο κεφάλαιο, τον αριθμό των μετοχών που έχει αγοράσει κ.α. Θα γνωρίζει επίσης τις δράσεις που μπορεί να εκτελέσει για παράδειγμα να αγοράσει, να πουλήσει, ή να μην κάνει τίποτα από τα δύο. Έτσι, ο Πράκτορας θα δρα κάθε φορά στο περιβάλλον και θα επιβραβεύεται για τις αποφάσεις του με βάση τη συνάρτηση επιβράβευσης που έχουμε δημιουργήσει. Με αυτό τον τρόπο, ο Πράκτορας θα προσπαθήσει να μεγιστοποιήσει την ανταμοιβή που λαμβάνει για τις δράσεις του, βελτιστοποιώντας έτσι την στρατηγική με την οποία πραγματοποιεί τις συναλλαγές του.

1.4 Δομή της εργασίας

Στο παρόν Κεφάλαιο γίνεται μία Εισαγωγή, σχετικά με το αντικείμενο, τον σκοπό και το κίνητρο της εργασίας. Στην συνέχεια, η εργασία έχει χωριστεί σε θεματικές ενότητες, με κατάλληλο τρόπο ώστε να διευκολύνεται η ανάγνωσή της. Συγκεκριμένα, στο Κεφάλαιο 2 ”Θεωρητικό Υπόβαθρο”, παρουσιάζονται γνώσεις που είναι απαραίτητες να έχει ο Αναγνώστης ώστε να μπορέσει να κατανοήσει τους αλγορίθμους Ενισχυτικής Μάθησης που παρουσιάζονται στο Κεφαλαίο 3. Στο Κεφάλαιο 3, παρουσιάζονται οι Αλγόριθμοι Ενισχυτικής Μάθησης που υλοποιήθηκαν και αξιολογήθηκαν στην παρούσα εργασία. Στην συνέχεια, το Κεφάλαιο 4, ”Αλγοριθμικές Συναλλαγές”, περιέχει τη βασική γνώση που χρειάζεται να έχει ο Αναγνώστης για το Χρηματιστήριο καθώς επίσης και για να κατανοήσει την εφαρμογή της Ενισχυτικής Μάθησης στις συναλλαγές. Στο Κεφάλαιο 5, ”Εφαρμογή Ενισχυτικής Μάθησης στις Συναλλαγές”, περιγράφεται ο τρόπος με τον οποίο μοντελοποιήθηκε το πρόβλημα των Συναλλαγών σαν πρόβλημα Ενισχυτικής Μάθησης, οι επιλογές που έγιναν για την μοντελοποίηση αυτή, καθώς επίσης και τα εργαλεία τα οποία χρησιμοποιήθηκαν για την υλοποίηση της. Τέλος, στα Κεφάλαια 6 και 7 παρουσιάζονται τα Πειράματα και τα Συμπεράσματα που λήφθηκαν, ενώ στο Κεφάλαιο 8, γίνεται αναφορά για τις κατευθύνσεις που θα μπορούσε να ακολουθήσει κάποιος μελλοντικά, ώστε να εμβαθύνει στην εφαρμογή της Ενισχυτικής Μάθησης στις συναλλαγές.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

2.1 Πράκτορες και Περιβάλλον

2.1.1 Ευφυής Πράκτορας

Με τον όρο ”Πράκτορας”, νοείται ένα υπολογιστικό σύστημα το οποίο δρα αυτόνομα σε ένα Περιβάλλον, με σκοπό την επίτευξη συγκεκριμένων στόχων. Βασικό χαρακτηριστικό του, λοιπόν, είναι η αυτονομία του. Σαν αυτόνομη οντότητα, ελέγχει τις καταστάσεις στις οποίες θα μεταβεί με τις δράσεις του, χωρίς να κατευθύνεται βήμα προς βήμα από κάποια άλλη εξωτερική οντότητα πχ τον άνθρωπο. Επιπλέον, για την επίτευξη στόχων , ένας Πράκτορας είναι πιθανό να συνεργαστεί με άλλους Πράκτορες. Έτσι, τελικά τα βασικά χαρακτηριστικά ενός Πράκτορα, είναι τα εξής [Wool95]:

- Αυτονομία (autonomy): Οι πράκτορες ενεργούν αυτόνομα, χωρίς παρεμβολή από χρήστες ή άλλους Πράκτορες, διατηρώντας πλήρη έλεγχο των πράξεων τους.
- Κοινωνικότητα (Social Ability): Οι Πράκτορες μπορεί να συνεργαστούν με άλλους Πράκτορες, για την επίτευξη των δικών τους ή κοινών στόχων.
- Αντιδραστικότητα (reactiveness): Οι Πράκτορες αλληλεπιδρούν με το Περιβάλλον, αντιλαμβάνονται τις αλλαγές σε αυτό και δρουν μέσα σε συγκεκριμένα χρονικά πλαίσια.
- Προνοητικότητα (pro-activeness): Οι Πράκτορες δεν αντιδρούν απλώς στις αλλαγές του Περιβάλλοντός, αλλά είναι ικανοί να επιδείξουν και συμπεριφορά που βασίζεται σε συγκεκριμένους στόχους, λαμβάνοντας πρωτοβουλία ανάλογα με τις συνθήκες που εμφανίζονται στο Περιβάλλον.

Οι ευφυείς πράκτορες διαθέτουν επιπλέον δευτερεύοντα χαρακτηριστικά που αφορούν το βαθμό νοημοσύνης που διαθέτουν, όπως:

- Προσαρμοστικότητα (adaptivity): Ο Πράκτορας προσαρμόζεται διαρκώς στο Περιβάλλον, έχει δηλαδή την ικανότητα μάθησης.
- Αγαθή Προαίρεση (benevolence): Προσπαθούν να πετύχουν πάντα τους στόχους που τους έχουν ανατεθεί.
- Ορθολογικότητα (Rationality): Αφορά την υπόθεση ότι ο Πράκτορας θα δρα πάντα βέλτιστα, προσπαθώντας να εκπληρώσει τον στόχο του.

2.1.2 Περιβάλλον

Όπως έχει αναφερθεί ένας Πράκτορας, αλληλεπιδρά και ανταποκρίνεται σε αλλαγές ενός Περιβάλλοντος. Ο όρος Περιβάλλον, ανταποκρίνεται συνήθως στην μοντελοποίηση ενός Προβλήματος, στο οποίο ο Πράκτορας καλείται να δώσει λύση. Έτσι, μπορούν να κατηγοριοποιηθούν ανάλογα με συγκεκριμένα χαρακτηριστικά ως εξής:

- Πλήρως Προσβάσιμα ή Μερικώς Προσβάσιμα (Fully Observable / Partially Observable): Ανάλογα με το αν υπάρχει διαθέσιμη ολόκληρη η πληροφορία. Σε ένα πλήρως προσβάσιμο περιβάλλον, ένας Πράκτορας δεν απαιτείται να έχει εσωτερική κατάσταση για την αναπαράσταση του Περιβάλλοντος. Αντιθέτως, σε ένα Μερικώς Προσβάσιμο Περιβάλλον, οι πράκτορες δε λαμβάνουν άμεση και λεπτομερειακή πληροφορία για αυτό.

- Αιτιοκρατικά ή Μη Αιτιοκρατικά (Deterministic / Stochastic): Αν η επόμενη κατάσταση ενός περιβάλλοντος μπορεί να προσδιοριστεί πλήρως από την τρέχουσα κατάστασή του και τις τρέχουσες δράσεις του πράκτορα, τότε αυτό το περιβάλλον καλείται αιτιοκρατικό, ενώ στην αντίθετη περίπτωση χαρακτηρίζεται ως μη αιτιοκρατικό.

- Επεισοδιακά ή Μη Επεισοδιακά (Episodic / Sequential): Ένα περιβάλλον χαρακτηρίζεται ως επεισοδιακό, όταν χωρίζεται σε διακριτά επεισόδια. Κάθε επεισόδιο χαρακτηρίζεται από την αντίληψη του πράκτορα για το περιβάλλον στο οποίο βρίσκεται την τρέχουσα στιγμή και από τις δράσεις του μέσα σε αυτό.

- Στατικά ή Δυναμικά (Static / Dynamic): Όταν ένα περιβάλλον μπορεί να μεταβάλλεται από τη φύση του, ενώ παράλληλα δρα μέσα σε αυτό ένας πράκτορας, τότε το περιβάλλον χαρακτηρίζεται ως δυναμικό αντίθετα χαρακτηρίζεται ως στατικό.

- Διακριτά ή Συνεχή (Discrete / Continuous): ως προς την ύπαρξη ή οχι πεπερασμένου αριθμού ενεργειών και δεδομένων στον μηχανισμό αντίληψης του Πράκτορα.

- Μονοπρακτορικό ή Πολυπρακτορικό (Single Agent / Multiagent): Αν ένας πράκτορας δρα μόνος του σε ένα περιβάλλον, το περιβάλλον χαρακτηρίζεται μονοπρακτορικό, ενώ σε αντίθετη περίπτωση το περιβάλλον είναι πολυπρακτορικό.

Οι παραπάνω κατηγοριοποιήσεις περιγράφουν εντέλει την πολυπλοκότητα ενός περιβάλλοντος. Όσο πιο πολύπλοκο είναι το περιβάλλον, τόσο πιο δύσκολη είναι η ανάπτυξη ενός λειτουργικού πράκτορα που δρα μέσα σε αυτό. Έτσι, ευκολότερη καθίσταται η ανάπτυξη συστημάτων Πρακτόρων για Περιβάλλοντα που είναι προσβάσιμα, αιτιοκρατικά, επεισοδιακά, στατικά, διακριτά και μονοπρακτορικά. Αντίθετα είναι εξαιρετικά πολύπλοκη, για Περιβάλλοντα που είναι μη προσβάσιμα, μη αιτιοκρατικά, μη επεισοδιακά, δυναμικά, συνεχή ή πολυπρακτορικά.

2.2 Μαθηματικά Εργαλεία

2.2.1 Αλυσίδα Markov

Η αλυσίδα Markov, ή Μαρκοβιανή αλυσίδα, είναι ένα μαθηματικό σύστημα που μεταβάλλεται από μια κατάσταση σε μια άλλη, ανάμεσα σε ένα πεπερασμένο αριθμό καταστάσεων ικανοποιώντας την Μαρκοβιανή Ιδιότητα. Είναι μια τυχαία διαδικασία που δε διατηρεί πληροφορία για τις προηγούμενες μεταβολές: Η επόμενη κατάσταση εξαρτάται μόνο από την τωρινή κατάσταση και σε καμιά περίπτωση από αυτές που προηγήθηκαν.

Συνήθως μια Μαρκοβιανή αλυσίδα ορίζεται για ένα σύνολο διακριτών βημάτων στο χρόνο (Μαρκοβιανή αλυσίδα διακριτού χρόνου). Μια τυχαία διαδικασία διακριτού χρόνου περιλαμβάνει ένα σύστημα που βρίσκεται σε μια συγκεκριμένη κατάσταση σε κάθε βήμα, με την κατάσταση να μεταβάλλεται τυχαία μεταξύ των βημάτων. Τυπικά τα βήματα είναι ακέραιοι ή φυσικοί αριθμοί και η τυχαία διαδικασία είναι η χαρτογράφησή τους σε καταστάσεις. Η Μαρκοβιανή ιδιότητα δηλώνει ότι η δεσμευμένη πιθανότητα κατανομής του συστήματος στο επόμενο βήμα (και κατά βάση, σε όλα τα μελλοντικά βήματα) εξαρτάται μόνο από την παρούσα κατάσταση του συστήματος και όχι αθροιστικά από την κατάσταση του συστήματος σε προηγούμενα βήματα.

Καθώς το σύστημα μεταβάλλεται τυχαία, είναι γενικά αδύνατο να προβλεφθεί με βεβαιότητα η κατάσταση μιας Μαρκοβιανής αλυσίδας σε ένα δεδομένο μελλοντικό σημείο. Οι αλλαγές κατάστασης

του συστήματος ονομάζονται μεταβάσεις και οι πιθανότητες που σχετίζονται με τις διάφορες μεταβατικές καταστάσεις ονομάζονται πιθανότητες μετάβασης. Η διαδικασία χαρακτηρίζεται από ένα χώρο καταστάσεων, ένα πίνακα μετάβασης που περιγράφει τις πιθανότητες μιας συγκεκριμένης μετάβασης και μια αρχική κατάσταση ή αρχική κατανομή στο χώρο καταστάσεων.

Καθώς το σύστημα μεταβάλλεται τυχαία, είναι γενικά αδύνατο να προβλεφθεί με βεβαιότητα η κατάσταση μιας Μαρκοβιανής αλυσίδας σε ένα δεδομένο μελλοντικό σημείο. Παρ' όλα αυτά, οι στατιστικές ιδιότητες του μέλλοντος του συστήματος μπορούν να προβλεφθούν, κάτι που είναι ιδιαίτερα χρήσιμο για πολλές εφαρμογές.

Πιθανότητες Μετάβασης

Μια Μαρκοβιανή Αλυσίδα ορίζεται ως μια ακολουθία τυχαίων μεταβλητών X_1, X_2, X_3, ... ,X_n με τη Μαρκοβιανή Ιδιότητα, δηλαδή με δεδομένη την παρούσα κατάσταση, οι παλαιότερες και οι μελλοντικές καταστάσεις είναι ανεξάρτητες. Έτσι, για μία επόμενη κατάσταση X_{n+1} μπορούμε να ορίσουμε την πιθανότητα μετάβασης σε αυτή ως

$$P(X_{n+1} = x | X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = P(X_{n+1} = x | X_n = x_n) \tag{2.1}$$

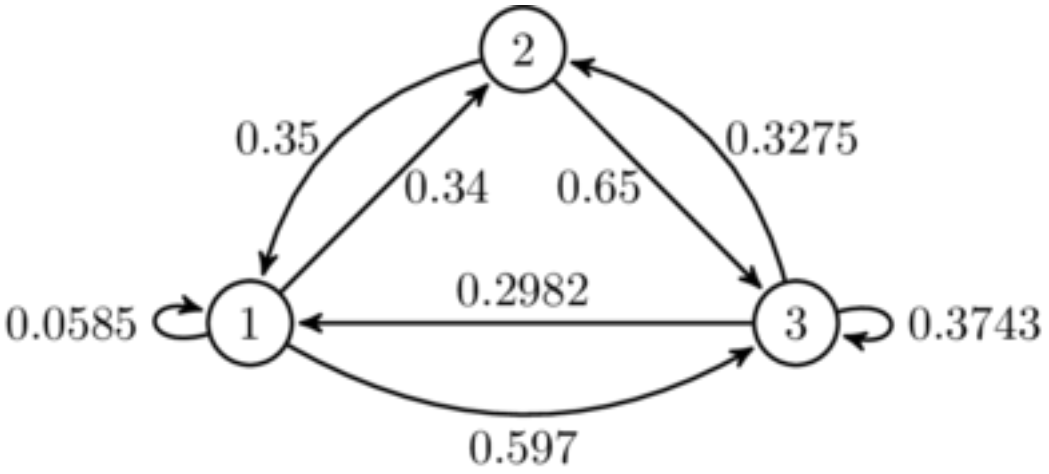
Όλες οι πιθανές τιμές των τυχαίων μεταβλητών X_i σχηματίζουν ένα αριθμήσιμο σύνολο S που ονομάζουμε χώρο-καταστάσεων της αλυσίδας. Έτσι, η μετάβαση από μία κατάσταση σε μία άλλη είναι μία πιθανοκρατική διαδικασία, όμως το σύστημα σίγουρα θα μεταβεί σε μία νέα κατάσταση κι έτσι η παραγωγή ενός συμβόλου εξόδου είναι ντετερμινιστική. Ορίζοντας λοιπόν, σύμφωνα με την παραπάνω σχέση, ως

$$p_{ij} = P(X_{n+1} = j | X_n = i) \tag{2.2}$$

, θα ισχύει

$$\sum_j p_{ij} = 1, \forall i \tag{2.3}$$

Οι Μαρκοβιανές Αλυσίδες συχνά περιγράφονται από ένα κατευθυνόμενο γράφημα που στις ακμές του αναγράφονται οι πιθανότητες μετάβασης από τη μια κατάσταση στις άλλες.



Σχήμα 2.1: Παράδειγμα αλυσίδας Markov.

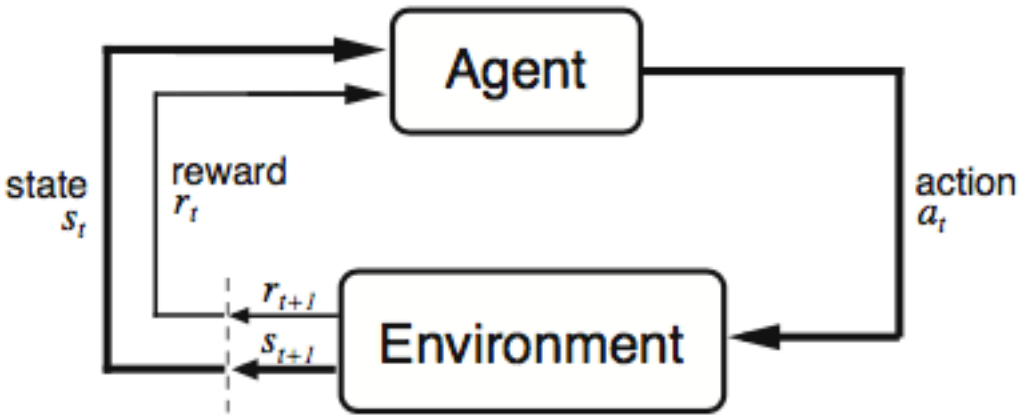
2.2.2 Διαδικασία Απόφασης Markov

Μία διαδικασία απόφασης Markov (MDP), παρέχει ένα μαθηματικό πλαίσιο για τη μοντελοποίηση λήψης αποφάσεων σε καταστάσεις όπου τα αποτελέσματα είναι εν μέρει τυχαία και εν μέρει υπό

τον έλεγχο ενός υπεύθυνου λήψης αποφάσεων. Συγκεκριμένα, αποτελεί μία γενίκευση της Αλυσίδας Markov, με την διαφορά ότι έχει προστεθεί η δυνατότητα για δράσεις και ανταμοιβές. Αν υπήρχε μόνο μία δυνατή δράση (πχ Στάση), και οι ανταμοιβές ήταν πάντα μηδέν, τότε θα είχαμε μία διαδικασία Markov ακριβώς όπως περιγράφηκε παραπάνω. Έτσι, σε κάθε βήμα, η MDP βρίσκεται σε κάποια κατάσταση s και ο υπεύθυνος λήψης αποφάσεων μπορεί να επιλέξει οποιαδήποτε δράση a που είναι δυνατή στην κατάσταση s. Η διαδικασία αποκρίνεται προχωρώντας σε μια νέα κατάσταση s' με πιθανότητα μετάβασης $P(s'|s,a)$ και δίνοντας στον υπεύθυνο λήψης αποφάσεων αντίστοιχη ανταμοιβή r_t.

Ετσι, μία MDP χαρακτηρίζεται, από τα παρακάτω:

- S ένα σύνολο καταστάσεων,
- A ένα σύνολο δράσεων,
- $P(s'|s,a)$ η πιθανότητα η επιλεγμένη δράση a στην κατάσταση s το χρόνο t να οδηγήσει στην κατάσταση s' την χρονική στιγμή $t+1$. Η πιθανότητα μετάβασης, ικανοποιεί την Μαρκοβιανή ιδιότητα.
- $R(s,a,s')$ η συνάρτηση ανταμοιβής, που καθορίζει την άμεση ανταμοιβή που θα λάβει ο υπεύθυνος λήψης αποφάσεων, μετά την μετάβαση από την κατάσταση s στην κατάσταση s', λόγω της δράσης a.



Σχήμα 2.2: Αλληλεπίδραση Πράκτορα - Περιβάλλοντος σε μία Διαδικασία Απόφασης Markov

Είναι σημαντικό να αναφερθεί, πώς παρόλο που η συνάρτηση $R(s,a,s')$ ονομάζεται συνάρτηση ανταμοιβής, μπορεί συχνά να παράγει αρνητικά σήματα ώστε να ενημερώσει τον Πράκτορα ότι η μετάβαση από την κατάσταση s στην κατάσταση s', λόγω της δράσης a, είναι ζημιογόνα.

2.2.3 Επίλυση Διαδικασίας Απόφασης Markov

Το βασικό πρόβλημα σε μία MDP, έγκειται στην εύρεση μίας πολιτικής π από τον υπεύθυνο αποφάσεων. Η συνάρτηση π, προσδιορίζει την δράση $\pi(s)$, που πρέπει να παρθεί από τον υπεύθυνο αποφάσεων, στην κατάσταση s. Στόχος είναι η επιλογή της πολιτικής π, η οποία θα μεγιστοποιήσει το αναμενόμενο άθροισμα των επιβραβεύσεων, σε άπειρο ή περιορισμένο χρονικά ορίζοντα H. Χρησιμοποιείται, επίσης, και ένας εκπτωτικός παράγοντας γ, ώστε να δείξει στον υπεύθυνο λήψης

αποφάσεων σε τι βαθμό τον ενδιαφέρει η επιβράβευση την τωρινή χρονική στιγμή, σε σχέση με μία μελλοντική. Έτσι, ο στόχος του Πράκτορα διαμορφώνεται ως εξής:

$$\max_{\pi} E[\sum_{t=0}^{H} \gamma^t R(s_t, a_t, s_{t+1})|\pi] \tag{2.4}$$

όπου,

$a_t = \pi(s_t)$: η δράση που επιλέγει ο Πράκτορας όταν βρίσκεται στην κατάσταση s_t,

γ : ο εκπτωτικός παράγοντας με $0 \leq \gamma \leq 1$. Τυπική τιμή του γ είναι κοντά στο 1,

ώστε να παροτρύνει τον Πράκτορα, να λάβει δράση στο άμεσο μέλλον.

H : Ο χρονικός ορίζοντας, πεπερασμένος ή άπειρος.

Για την εύρεση της βέλτιστης πολιτικής η οποία περιγράφει την καλύτερη δυνατή δράση σε μία κατάσταση της MDP, χρησιμοποιούνται συχνά τεχνικές δυναμικού προγραμματισμού. Αυτές οι τεχνικές απαιτούν συνήθως να είναι γνωστή η συνάρτηση μετάβασης P, καθώς και η συνάρτηση ανταμοιβής R.

Θεωρώντας γνωστές τις συγκεκριμένες συναρτήσεις, αναζητείται η ποσότητα π^*, η οποία θα μεγιστοποιήσει το αναμενόμενο άθροισμα των μειωμένων κατά τον παράγοντα γ, ανταμοιβών όπως περιγράφεται παραπάνω στην (2.4)

Για την εύρεση αυτού του Στόχου 2.4 ορίζουμε ως βέλτιστη Συνάρτηση Αξίας

$$V^*(s) = \max_{\pi} E[\sum_{t=0}^{H} \gamma^t R(s_t, a_t, s_{t+1})|\pi, s_0 = s] \tag{2.5}$$

, όπου s_0 η αρχική κατάσταση.

Η βέλτιστη συνάρτηση αξίας $V^*(s)$ δείχνει το αναμενόμενο άθροισμα μειωμένης ανταμοιβής, που μπορεί να λάβει ο Πράκτορας ξεκινώντας από την κατάσταση s, αν χρησιμοποιούσε την βέλτιστη πολιτική π .

Value Iteration

Για την εύρεση της βέλτιστης πολιτικής και κατ' επέκταση της $V^*(s)$ χρησιμοποιούμε την τεχνική του Value Iteration ως εξής:

Σε κάθε χρονική στιγμή k του ορίζοντα H, η $V_k^*(s)$ υπολογίζεται αναδρομικά, από την επιλογή εκείνης της δράσης a που μεγιστοποιεί το άθροισμα της επιβράβευσης $R(s, a, s')$ και της βέλτιστης αξίας $V_{k-1}^*(s')$ της επόμενης κατάστασης s' μειωμένης κατά τον εκπτωτικό παράγοντα γ και έχοντας πλέον $k-1$ εναπομείναντες επαναλήψεις. Έτσι,

$$V_0^*(s) = 0, \forall s, H = 0 \tag{2.6}$$

$$V_k^*(s) = \max_{a} \sum_{s'} P(s'|s, a)(R(s_t, a_t, s_{t+1}) + \gamma V_{k-1}^*(s')) \tag{2.7}$$

Αντίστοιχα, υπολογίζεται και η βέλτιστη πολιτική, ως

$$\pi_k^*(s) = \arg\max_{a} \sum_{s'} P(s'|s, a)(R(s_t, a_t, s_{t+1}) + \gamma V_{k-1}^*(s')) \tag{2.8}$$

Έτσι, τελικά διαμορφώνεται ο παρακάτω Αλγόριθμος, γνωστός και ως Value Update ή Bellman / Update back up:

Start with $V_0^*(s) = 0$ for all s.

For k = 1, … , H:

For all states s in S:

$$V_k^*(s) \leftarrow \max_a \sum_{s'} P(s'|s,a) \left(R(s,a,s') + \gamma V_{k-1}^*(s')\right)$$

$$\pi_k^*(s) \leftarrow \arg\max_a \sum_{s'} P(s'|s,a) \left(R(s,a,s') + \gamma V_{k-1}^*(s')\right)$$

Σχήμα 2.3: Αλγόριθμος Value Update ή Bellman / Update back up

2.2.4 Μερικώς Προσβάσιμη Διαδικασία Απόφασης Markov

Σε μία διαδικασία απόφασης Markov, όπως έχει ήδη διατυπωθεί, η δράση a του Πράκτορα, στην κατάσταση s, καθορίζει την επόμενη κατάσταση s'. Παρόλα αυτά, σε πολλές περιπτώσεις στον πραγματικό κόσμο, όπως αναφέρθηκε στην ενότητα 2.1.2, η νέα κατάσταση του Περιβάλλοντος δεν μπορεί να παρατηρηθεί ολόκληρη από τον Πράκτορα, για παράδειγμα επειδή η μεταβολή κάποιων ποσοτήτων της εξαρτάται από παράγοντες που είναι άγνωστοι για τον ίδιο τον Πράκτορα. Αυτού του είδους τα προβλήματα μπορούν να αναπαρασταθούν ως μία Μερικώς Παρατηρήσιμη Διαδικασία Απόφασης Markov.

2.3 Μηχανική Μάθηση

Η Μηχανική Μάθηση είναι ένα πεδίο της Επιστήμης Υπολογιστών κατά το οποίο μελετώνται αλγόριθμοι, οι οποίοι επιτρέπουν σε ένα υπολογιστικό σύστημα να πραγματοποιεί προβλέψεις ή να λαμβάνει αποφάσεις χωρίς να έχει προγραμματιστεί η ακριβής συλλογιστική πορεία που πρέπει να ακολουθήσει για να τις λάβει. Συγκεκριμένα, ο αλγόριθμος μηχανικής μάθησης δημιουργεί ένα μαθηματικό μοντέλο που βασίζεται σε δεδομένα - δείγματα, ή αλλιώς δεδομένα εκπαίδευσης. Μοντέλα μηχανικής μάθησης, χρησιμοποιούνται σε διάφορους τομείς, όπως η πρόβλεψη των τιμών μίας μετοχής, η αυτόματη αναγνώριση ηλεκτρονικών μηνυμάτων απάτης, η επεξεργασία φυσικής γλώσσας κ.α. . Ανάλογα με το είδος του προβλήματος που χρειάζεται κάθε φορά να αντιμετωπιστεί, οι αλγόριθμοι Μηχανικής Μάθησης χωρίζονται σε τρεις βασικές κατηγορίες

- Εποπτευόμενη Μάθηση (Supervided Learning)
- Μη Εποπτευόμενη Μάθηση (Unsupervised Learning)
- Ενισχυτική Μάθηση (Reinforcement Learning)

2.3.1 Είδη Μηχανικής Μάθησης

Εποπτευόμενη Μάθηση

Ένα σύστημα εποπτευόμενης μάθησης, προσπαθεί να προσεγγίσει επαγωγικά μία Συνάρτηση Στόχο (Target Function), ώστε να μοντελοποιήσει τα δεδομένα εισόδου .Το σύστημα, λαμβάνει σαν είσοδο δείγματα, δηλαδή τιμές ενός συνόλου μεταβλητών (χαρακτηριστικά) καθώς επίσης και το αποτέλεσμα που θα έπρεπε να παράξει η Συνάρτηση Στόχος εάν ήταν γνωστή (labels). Έτσι, το σύστημα εξετάζει διαφορετικές συναρτήσεις οι οποίες θα μπορούσαν να μοντελοποιούν τα δεδομένα εισόδου, θεωρώντας επαγωγικά ότι η συνάρτηση που έχει διαμορφώσει, μοντελοποιεί σωστά και περιπτώσεις που δεν έχουν εξετασθεί. Τα προβλήματα εποπτευόμενης μάθησης, διακρίνονται σε επιπλέον κατηγορίες, εκ των οποίων οι κυριότερες είναι:

- Ταξινόμηση (Classification)
- Παρεμβολή ή Παλινδρόμηση (Regression)

Οι αλγόριθμοι ταξινόμησης, προσπαθούν να δημιουργήσουν μοντέλα ώστε να κάνουν πρόβλεψη διακριτών κλάσεων / κατηγοριών. Αντίθετα, οι αλγόριθμοι Παρεμβολής προσπαθούν να προβλέψουν μία συγκεκριμένη τιμή, μέσα σε ένα εύρος τιμών.

Αρκετές τεχνικές έχουν δημιουργηθεί για την αντιμετώπιση προβλημάτων των παραπάνω κατηγοριών, κυριότερες εκ των οποίων είναι:

- Μάθηση εννοιών (Concept Learning)
- Δένδρα ταξινόμησης ή απόφασης (Classification or Decision Trees)
- Μάθηση Κανόνων (Rule Learning)
- Μάθηση κατά Περίπτωση (Instance Based Learning)
- Μάθηση κατά Bayes
- Γραμμική Παρεμβολή (Linear Regression)
- Νευρωνικά Δίκτυα (Neural Networks)
- Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines, SVMs)

Μη Εποπτευόμενη Μάθηση

Οι αλγόριθμοι μη εποπτευόμενης μάθησης λαμβάνουν ένα σύνολο δεδομένων στα οποία δεν προσδιορίζεται η επιθυμητή έξοδος. Συνεπώς, οι αλγόριθμοι μαθαίνουν από δεδομένα που δεν έχουν επισημανθεί ή κατηγοριοποιηθεί. Έτσι ένα σύστημα μη εποπτευόμενης μάθησης έχει σκοπό να ανακαλύψει συσχετίσεις και πρότυπα συμπεριφοράς με βάση τα χαρακτηριστικά των δεδομένων εισόδου.

Ενισχυτική Μάθηση

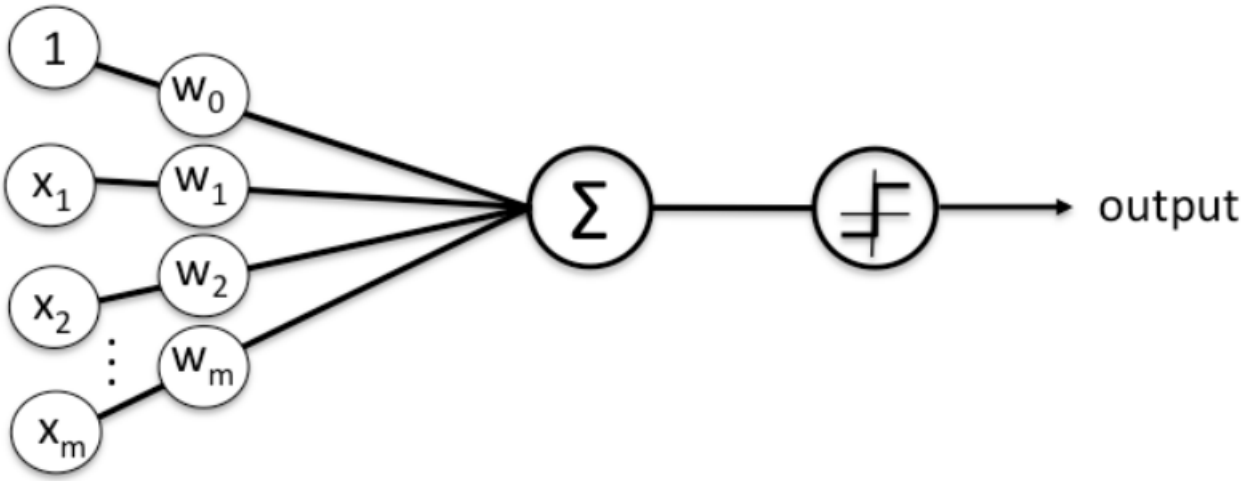
Η Ενισχυτική Μάθηση, μελετά τεχνικές στις οποίες ένα σύστημα προσπαθεί να μάθει από την άμεση αλληλεπίδραση με το Περιβάλλον. Σκοπός του συστήματος μάθησης, είναι να μεγιστοποιήσει μία συνάρτηση σήματος ενίσχυσης (αμοιβή). Το σύστημα, είναι αποκλειστικά υπεύθυνο για την επιλογή των δράσεων που θα ακολουθήσει ώστε να μεγιστοποιήσει την αμοιβή του. Λόγω της γενικότητας της μεθόδου, το πεδίο έχει μελετηθεί και εφαρμοστεί σε πολλούς κλάδους, όπως η θεωρία παιγνίων, η θεωρία ελέγχου, η επιχειρησιακή έρευνα κλπ.

2.3.2 Τεχνητά Νευρωνικά Δίκτυα

Τα Νευρωνικά Δίκτυα, εμπνευσμένα από τον τρόπο λειτουργίας του ανθρώπινου εγκεφάλου, είναι υπολογιστικά συστήματα, τα οποία αποτελούνται από ένα σύνολο συνδεδεμένων κόμβων, γνωστοί και ως Τεχνητοί Νευρώνες. Ο τρόπος με τον οποίο αυτοί οι κόμβοι είναι σχεδιασμένοι, καθιστά τα Νευρωνικά Δίκτυα ικανά να αναγνωρίζουν πρότυπα συμπεριφοράς, να βρίσκουν συσχετίσεις μεταξύ των δεδομένων που τους δίνονται σαν είσοδο. Είναι, έτσι, εξαιρετικά αποτελεσματικά στην προσέγγιση άγνωστων συναρτήσεων $f(x) = y$, βρίσκοντας το σωστό τρόπο μετάβασης ή μετατροπής του διανύσματος εισόδου x στο επιθυμητό αποτέλεσμα y.

Τεχνητός Νευρώνας

Ο Τεχνητός Νευρώνας, αποτελεί δομικό συστατικό ενός Νευρωνικού δικτύου, και προσπαθεί να προσομοιώσει την λειτουργία ενός βιολογικού νευρώνα του ανθρωπίνου εγκεφάλου. Συγκεκριμένα, λαμβάνει σαν είσοδο ένα σύνολο σημάτων $x_1, x_2, ..x_m$ (μεταβλητών), σε αντίθεση με τους νευρώνες του εγκεφάλου που λαμβάνουν ηλεκτρικούς παλμούς. Κάθε τέτοιο σήμα εισόδου, πολλαπλασιάζεται με ένα συντελεστή, ο οποίος ονομάζεται βάρος και στόχο έχει είτε να ενισχύσει είτε να αποδυναμώσει το συγκεκριμένο σήμα. Το παραγόμενο σήμα, προχωράει στην συνέχεια από δύο νέα στάδια, έναν αθροιστή Σ ο οποίος αθροίζει τα επηρεασμένα από τα βάρη σήματα και μία συνάρτηση συνάρτηση ενεργοποίησης (Activation Function), που δρα σαν φίλτρο, ορίζοντας ένα κατώφλι, το οποίο θα διαμορφώσει την τελική τιμή της εξόδου u. Επιπλέον των σημάτων εισόδου και των βαρών, ο νευρώνας έχει και ένα βάρος w_0, που ονομάζεται πόλωση (bias) και στο οποίο εφαρμόζεται συνεχώς μία σταθερή τιμή εισόδου $x_0 = 1$. Ο όρος αυτός, πρόκειται για ένα εξωτερικό σήμα εισόδου, το οποίο χρησιμοποιείται συχνά για διαμορφώσει κατάλληλα το κατώφλι της συνάρτησης ενεργοποίησης.



Σχήμα 2.4: Αναπαράσταση ενός Νευρώνα.

Έτσι, για τον κάθε Νευρώνα k θα ισχύει

$$u_k = \phi(\sum_{j=0}^{m} w_{kj} x_j) \tag{2.9}$$

, όπου ϕ η συνάρτηση ενεργοποίησης. Εάν το σήμα του Νευρώνα, δηλαδή το σήμα της εισόδου πολλαπλασιαμένο με τα βάρη, περάσει από την συνάρτηση ενεργοποίησης, τότε ο Νευρώνας θεωρείται ενεργοποιημένος.

2.3.3 Συναρτήσεις ενεργοποίησης

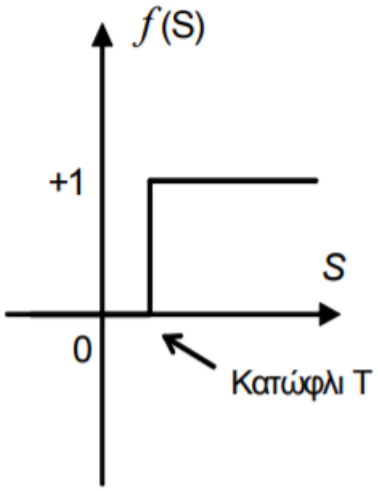
Κάποιες από τις βασικότερες συναρτήσεις ενεργοποίησης είναι οι:

1. Βηματική Συνάρτηση (Step Function)
2. Συνάρτηση Προσήμου (Sign Function)
3. Λογιστική Συνάρτηση (Logistic Function)
4. Συνάρτηση διορθωμένης γραμμικής μονάδας (ReLu)

Βηματική Συνάρτηση

Η βηματική συνάρτηση (step function), δίνει στην έξοδο την τιμή 1, μόνο αν η τιμή που υπολογίζει ο αθροιστής είναι μεγαλύτερη από μία τιμή κατωφλίου T.

$$\phi(S) = \begin{cases} 1, & \text{εάν S > T} \\ 0, & \text{σε οποιαδήποτε άλλη περίπτωση} \end{cases}$$

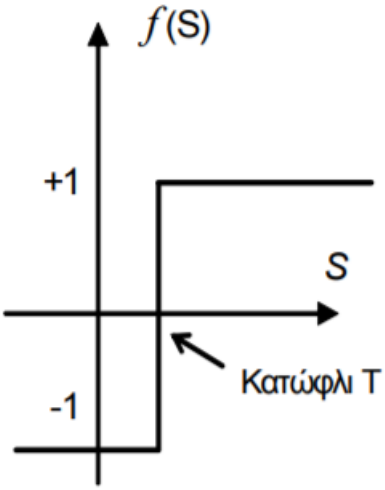


Σχήμα 2.5: Βηματική Συνάρτηση Ενεργοποίησης [IB06].

Συνάρτηση Προσήμου (Sign Function)

Η συγκεκριμένη συνάρτηση δίνει στην έξοδο αρνητικό ή θετικό σήμα, εάν η τιμή του αθροιστή είναι μικρότερη ή μεγαλύτερη από μία τιμή κατωφλίου T.

$$\phi(S) = \begin{cases} 1, & \text{εάν S > T} \\ -1, & \text{σε οποιαδήποτε άλλη περίπτωση} \end{cases}$$



Σχήμα 2.6: Συνάρτηση Ενεργοποίησης Προσήμου [IB06] .

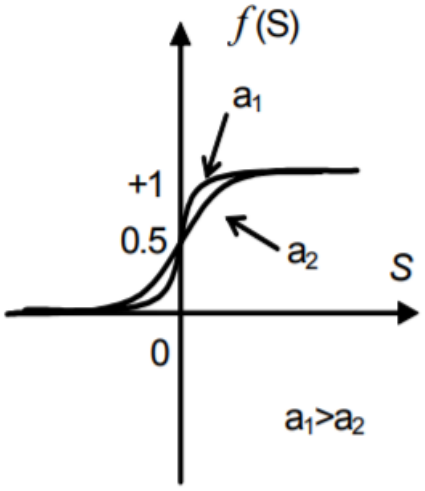
Λογιστική Συνάρτηση (Logistic Function)

Η λογιστική συνάρτηση περιγράφεται από την σχέση:

$$\phi(S) = \frac{1}{1 + e^{-aS}}$$

, όπου ο συντελεστής a, ρυθμίζει την ταχύτητα μετάβασης μεταξύ δύο ασυμπτωτικών τιμών.

Η λογιστική συνάρτηση, ανήκει σε μία οικογένεια συναρτήσεων που ονομάζεται σιγμοειδής (sigmoid). Άλλες τέτοιες συναρτήσεις είναι οι αντίστροφη εφαπτομένη (arctan) και η υπερβολική εφαπτομένη (tanh).



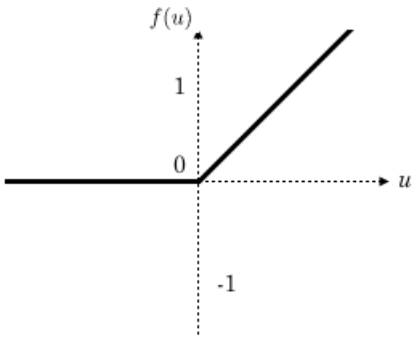
Σχήμα 2.7: Συνάρτηση Ενεργοποίησης Προσήμου [IB06].

Συνάρτηση διορθωμένης γραμμικής μονάδας (ReLu)

Μία επίσης συνηθισμένη συνάρτηση ενεργοποίησης, είναι η ReLU, η οποία χαρακτηρίζεται από την εξίσωση

$$f(u) = \max(0, u) \tag{2.10}$$

, όπου ο Νευρώνας ενεργοποιείται για τις θετικές τιμές, ενώ για τις αρνητικές δίνει σαν έξοδο μηδενικό σήμα.



Σχήμα 2.8: Συνάρτηση Ενεργοποίησης ReLU.

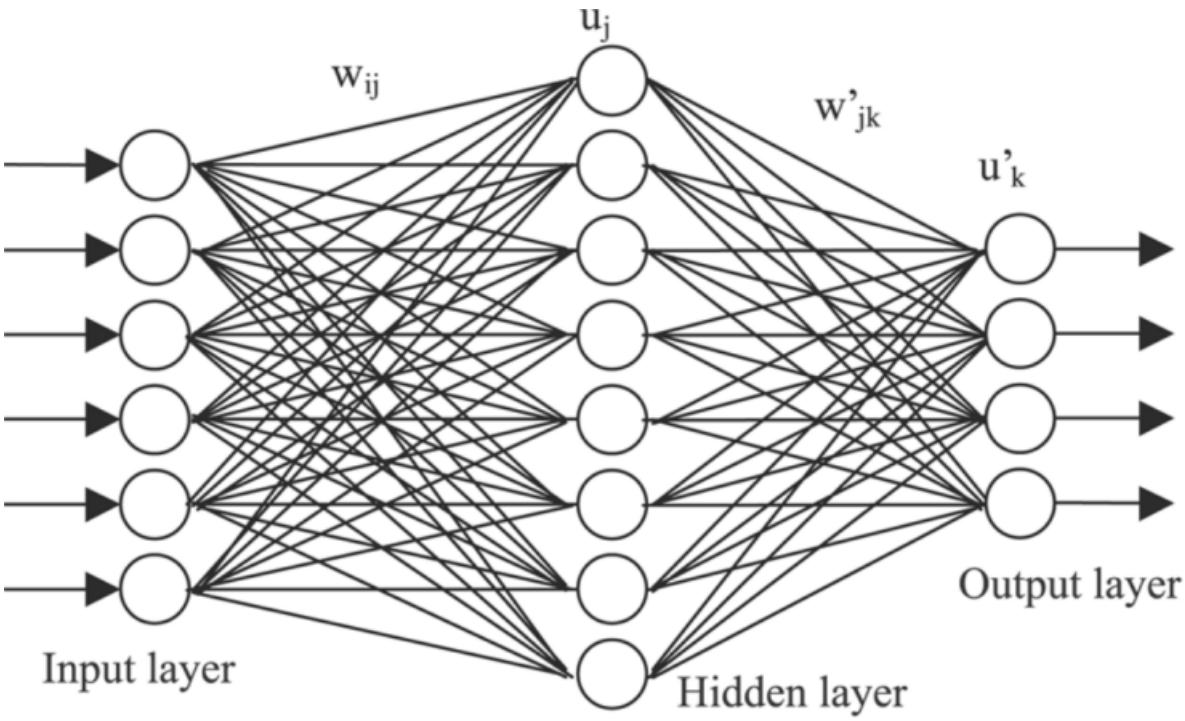
Αυτό που χαρακτηρίζει τις παραπάνω συναρτήσεις και τις καθιστά κατάλληλες για χρήση σε έναν Νευρώνα, είναι η μη-γραμμικότητα. Αν είχε επιλεχθεί μία γραμμική συνάρτηση, τότε θα παρήγαγε πά-

ντα έξοδο ανάλογης της εισόδου, κάτι που θα καθιστούσε αδύνατη τη μοντελοποίηση μη-γραμμικών φαινομένων.

2.3.4 Δομή Νευρωνικού Δικτύου

Για την δημιουργία ενός Νευρωνικού Δικτύου, οι παραπάνω Τεχνητοί Νευρώνες, οργανώνονται σε επίπεδα, όπου το κάθε επίπεδο επεξεργάζεται ένα σύνολο σημάτων. Το πρώτο επίπεδο, ονομάζεται επίπεδο εισόδου (input layer) και χρησιμοποιείται για την εισαγωγή των δεδομένων. Είναι σημαντικό να αναφερθεί ότι τα στοιχεία του επιπέδου αυτού, δεν είναι νευρώνες σαν αυτούς που περιγράφηκαν παραπάνω, αφού δεν εκτελούν κάποιον υπολογισμό. Στην συνέχεια, προστίθενται κρυφά επίπεδα (hidden layers), ένα ή περισσότερα, ενώ στο τέλος υπάρχει το επίπεδο εξόδου (output layer).

Επιπλέον, οι Νευρώνες σε ένα Νευρωνικό Δίκτυο, μπορούν να χαρακτηριστούν ως μερικώς ή πλήρως συνδεδεμένοι. Έτσι, εάν όλοι οι Νευρώνες ενός επιπέδου συνδέονται με όλους τους υπόλοιπους, τότε χαρακτηρίζονται ως πλήρως συνδεδεμένοι (fully connected). Σε κάθε άλλη περίπτωση χαρακτηρίζονται ως μερικώς συνδεδεμένοι (partially connected). Μία τυπική περίπτωση μερικής σύνδεσης, είναι τα Δίκτυα με πρόσθια προώθηση (feedforward). Στα συγκεκριμένα, οι Νευρώνες ενός επιπέδου συνδέονται πλήρως με τους Νευρώνες στο επόμενο επίπεδο, ενώ δεν υπάρχουν συνδέσεις μεταξύ των Νευρώνων ενός επιπέδου και του προηγουμένου στο Νευρωνικό Δίκτυο. Σε αντίθετη περίπτωση το δίκτυο χαρακτηρίζεται ως δίκτυο ανατροφοδότητας (feedback network ή recurrent network) , για παράδειγμα εάν υπάρχουν συνδέσεις μεταξύ νευρώνων σε προηγούμενο ή στο ίδιο επίπεδο.



Σχήμα 2.9: Αναπαράσταση ενός Νευρωνικού Δικτύου.

Βασικές Ιδιότητες των Νευρωνικών Δικτύων

Είναι σημαντικό επίσης να αναφερθούν, κάποιες από τις πιο βασικές ιδιότητες των Νευρωνικών Δικτύων [IB06]. Συγκεκριμένα,

- Μπορούν να μαθαίνουν μέσω παραδειγμάτων (learn by exsmple)
- Μπορούν να θεωρηθούν ως μνήμες συσχέτισης (associative memory)

- Έχουν μεγάλη ανοχή στο σφάλμα (fault-tolerant).
- Είναι ικανά να αναγνωρίσουν πρότυπα.

Ειδικότερα, το γεγονός ότι μαθαίνουν μόνο μέσω παραδειγμάτων, τα καθιστά ικανά να οργανώνουν κατάλληλα την πληροφορία εισόδου και εξόδου, δημιουργώντας ένα μοντέλο που αναπαριστά τη σχέση μεταξύ τους. Λειτουργούν επίσης, ως μνήμη συσχέτισης, αφού δεν αποθηκεύουν πληροφορία όπως ένα απλό υπολογιστικό πρόγραμμα σε θέσεις μνήμης, αλλά μέσω κατάλληλων συσχετίσεων που έχουν εντοπίσει μεταξύ των δεδομένων εισόδου και εξόδου. Αντίστοιχα και η ανάκτηση της πληροφορίας αυτής γίνεται με βάση το περιεχόμενό της. Αυτό τα καθιστά ανεκτικά, σε μικρές αλλαγές ή θόρυβο στα σήματα εισόδου.

Επιπλέον, είναι πολύ σημαντικό το γεγονός ότι η αναπαράσταση του μοντέλου, είναι κατανεμημένη σε όλα τα βάρη του δικτύου. Έτσι, ακόμη κι αν καταστραφεί κάποιος νευρώνας ή συνδέσεις, η απόδοση του δικτύου δεν θα επηρεαστεί σημαντικά. Πιο συγκεκριμένα, το μέγεθος του σφάλματος λόγω αστοχιών στην δομή του δικτύου, είναι ανάλογο του ποσοστού των κατεστραμμένων συνδέσεων. Τέλος, έχοντας πλέον εκπαιδευτεί, το Δίκτυο μπορεί να αναγνωρίσει κατευθείαν καταστάσεις στις οποίες δεν έχει ξαναβρεθεί.

Νευρωνικά Δίκτυα Πρόσθιας Τροφοδότησης

Τα Νευρωνικά Δίκτυα Πρόσθιας Τροφοδότησης, αποτελούν μία απλή μορφή νευρωνικών δικτύων και όπως αναφέρθηκε αυτό που τα χαρακτηρίζει είναι πως η πληροφορία κινείται μόνο από την είσοδο προς την έξοδο. Έτσι, σε ένα τέτοιο δίκτυο, υπάρχει ένα επίπεδο εισόδου, ένα ή προαιρετικά περισσότερα κρυφά επίπεδα καθώς και το επίπεδο εξόδου. Για την επίλυση προβλημάτων με την συγκεκριμένη τοπολογία, πρέπει να προσδιοριστούν, δύο πολύ βασικά χαρακτηριστικά. Το πρώτο σχετίζεται με τον τρόπο με τον οποίο θα εκπαιδευτεί το δίκτυο, ώστε να μοντελοποιήσει σωστά τα δεδομένα εισόδου. Το δεύτερο, σχετίζεται με την ακριβή δομή του δικτύου, δηλαδή, πόσα κρυφά επίπεδα θα έχει το δίκτυο πόσους νευρώνες θα έχει το καθένα και πώς θα συνδέονται μεταξύ τους. Ο προσδιορισμός των νευρώνων στο επίπεδο της εισόδου καθώς και στης εξόδου, είναι σχετικά εύκολος και μπορεί να καθοριστεί από το ίδιο το πρόβλημα που χρήζει επίλυσης. Αντίθετα, για τον αριθμό των νευρώνων που θα απαρτίζουν τα κρυφά επίπεδα, δεν υπάρχει συγκεκριμένος κανόνας. Δεν υπάρχει επίσης, κανόνας για την συνδεσμολογία που θα πρέπει να έχουν οι Νευρώνες. Παρόλα αυτά, στην πράξη αρκετά συνηθισμένη είναι η περίπτωση, ο κάθε Νευρώνας, να συνδέεται με όλους τους Νευρώνες του επόμενου επιπέδου [IB06] .

2.3.5 Percepton

Το Percepton, μία από τις πρώτες προσεγγίσεις τεχνητών νευρωνικών δικτύων, είναι μία τοπολογία πρόσθιας τροφοδότησης, χωρίς κρυφά επίπεδα. Η πιο απλή μορφή του, στοιχειώδες percepton, περιλαμβάνει έναν τεχνητό Νευρώνα, σαν αυτόν που απεικονίζεται στο σχήμα (2.4) και χρησιμοποιεί ως συνάρτηση ενεργοποίησης τη βηματική συνάρτηση. Στην συνέχεια, λαμβάνει σαν είσοδο μια σειρά δεδομένων και προσπαθεί με βάση το σφάλμα (error) μεταξύ της δικής του και της επιθυμητής εξόδου, να διαμορφώσει κατάλληλα τις τιμές των βαρών w_j. Έτσι, ο αλγόριθμος μάθησης ενός Percepton ακολουθεί τα παρακάτω βήματα.

Μέχρι να μην μεταβάλλονται σημαντικά τα βάρη ή να συμπληρωθεί δεδομένος αριθμός εποχών και για κάθε ζευγάρι εισόδου x και επιθυμητής εξόδου t

1. Υπολογίζεται η έξοδος y
2. Εάν η έξοδος είναι ίση με την επιθυμητή έξοδο t δεν γίνεται μεταβολή στα βάρη.
3. Εάν η έξοδος διαφέρει από την επιθυμητή τα βάρη μεταβάλλονται κατά ποσότητα $\Delta w = \alpha(t - y)x$, ώστε το y να πλησιάσει το t.

Η ποσότητα α που αναφέρεται παραπάνω, καθορίζει τον ρυθμό μεταβολής των βαρών. Έτσι, ονομάζεται ρυθμός μάθησης (learning rate) και έχει συνήθως τιμή μεταξύ 0 και 1.

Με βάση το στοιχειώδες percepton, μπορούν να δημιουργηθούν πιο προηγμένες αρχιτεκτονικές με περισσοτέρους Νευρώνες. Παρόλα αυτά, ο τρόπος λειτουργίας τους, είναι παρόμοιος.

2.3.6 Κανόνας Δέλτα (Delta rule learning)

Ο κανόνας Δέλτα (Delta Rule), αναπτύχθηκε από τους Widrow και Hoff, και αποτελεί γενίκευση του αλγορίθμου εκπαίδευσης Percepton. Συγκεκριμένα, είναι και αυτός καθοδηγούμενος από το σφάλμα, αφού προκύπτει από την ελαχιστοποίηση του μέσου τετραγωνικού σφάλματος (mean square error) των διανυσμάτων εκπαίδευσης. Έτσι, δείχνει τελικά πόσο αποκλίνει ένα δίκτυο από την επιθυμητή συνάρτηση. Το μέσο τετραγωνικό σφάλμα στο στοιχειώδες Percepton, για k διανύσματα εκπαίδευσης, μπορεί να υπολογιστεί από τη σχέση:

$$MSE = \frac{1}{k}\sum_{j=0}^{k}(t_j - input_j) \tag{2.11}$$

, όπου σαν σήμα εισόδου θεωρείται το $\sum_{i=0}^{n} w_{ki}x_i$, με n ο αριθμός των επιμέρους σημάτων εισόδου του νευρώνα.

Θεωρώντας έτσι, το διάνυσμα των βαρών $(w_1, w_2, ..., w_n)$, μπορεί να οριστεί ο κανόνας Δέλτα που ονομάζεται και κανόνας επικλινούς μεθόδου (gradient decent), ως η αρνητική παράγωγος του μέσου τετραγωνικού σφάλματος, έτσι ώστε το διάνυσμα βαρών να προσεγγίσει σταδιακά το ιδανικό διάνυσμα. Θεωρώντας ότι

$$\Delta w = -\frac{\partial(MSE)}{\partial w_i} \tag{2.12}$$

και την παράγωγο του MSE ώς προς όλα τα w

$$\nabla(MSE) = (\frac{\partial(MSE)}{\partial w_1}, ..., \frac{\partial(MSE)}{\partial w_n}) \tag{2.13}$$

προκύπτει ότι η μεταβολή στην τιμή του βάρους w_i, για ένα από τα διανύσματα εκπαίδευσης x_i, ως

$$\Delta w = w_{i(new)} - w_{i(old)} = \alpha(t - input)x_i \tag{2.14}$$

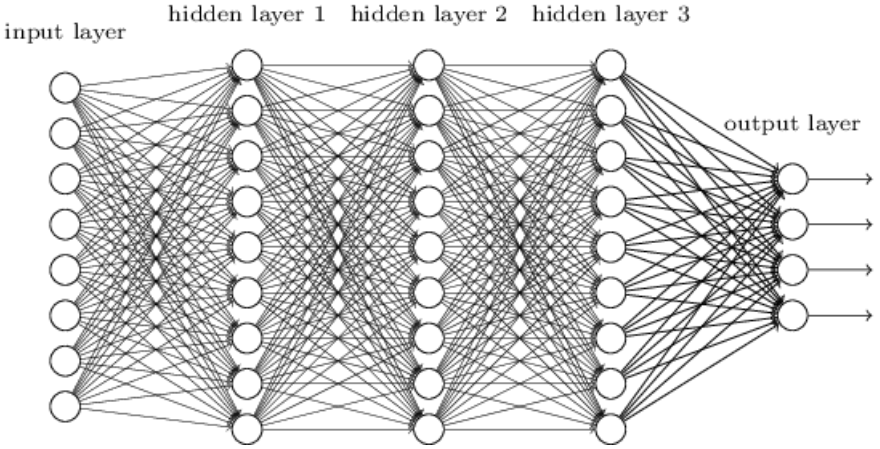
, όπου $input$ είναι το συνολικό σήμα του νευρώνα, t η επιθυμητή έξοδος, $w_{i(new)}$ και $w_{i(old)}$ η νέα και η παλιά τιμή του βάρους w_i, x_i η είσοδος της οποίας το βάρος αναπροσαρμόζεται και α ο ρυθμός μάθησης (learning rate), που ρυθμίζει το ρυθμό μεταβολής των βαρών. Η σταθερά α επηρεάζει την ταχύτητα σύγκλισης και καθορίζει την απόδοση του κανόνα Δέλτα. Συγκεκριμένα, πολύ μεγάλες τιμές του α επιταχύνουν τη σύγκλιση στο ελάχιστο σφάλμα, όμως αυξάνουν τον κίνδυνο να προσπεραστεί το ελάχιστο MSE. Αντίθετα, οι μικρές τιμές του α μπορεί να αυξήσουν σημαντικά τον χρόνο εκπαίδευσης.

Επιπλέον, στην παραπάνω σχέση μπορεί αντί του σήματος εισόδου $input$ να χρησιμοποιηθεί η πραγματική έξοδος του Νευρωνικού Δικτύου, όπως περιγράφεται στην σχέση (2.9), λαμβάνοντας έτσι υπόψιν και τη συνάρτηση ενεργοποίησης. Η εκπαίδευση του Δικτύου, σταματά όταν το μέσο τετραγωνικό σφάλμα, γίνει μικρότερο από κάποια επιθυμητή τιμή.

Παρόλο που ο Αλγόριθμος Δέλτα, αποτελεί βελτίωση του αλγορίθμου μάθησης του στοιχειωδούς percepton, δεν μπορεί να εφαρμοστεί στα δίκτυα με κρυφά επίπεδα, επειδή δεν είναι γνωστή η επιθυμητή έξοδος t σε κάθε Νευρώνα. Για την επίλθυση του συγκεκριμένου προβλήματος χρησιμοποιείται ο Αλγόριθμος ανάστροφης μετάδοσης λάθους.

2.3.7 Νευρωνικό δίκτυο πολλών επιπέδων (Multi Layer Percepton)

Αρχικά, κατασκευάζεται ένα Δίκτυο με πολλά κρυμμένα επίπεδα, με παρόμοιο τρόπο με αυτόν που κατασκευάστηκε το απλό Νευρωνικό Δίκτυο.



Σχήμα 2.10: Νευρωνικό δίκτυο πολλών επιπέδων. (Multi Layer Percepton)

Είναι εμφανές πως η πολυπλοκότητα ενός τέτοιου δικτύου εξαρτάται τόσο από τον βαθμό των επιπέδων όσο και από τον αριθμό των Νευρώνων σε κάθε επίπεδο. Έχει αποδειχθεί πως η συγκεκριμένη τοπολογία, με τον κατάλληλο αριθμό επιπέδων και νευρώνων μπορεί να προσεγγίσει οποιαδήποτε συνάρτηση, σε οποιονδήποτε βαθμό ακριβείας [IB06].

Αλγόριθμος ανάστροφης μετάδοσης λάθους (back propagation)

Για την παραπάνω τοπολογία είναι απαραίτητος ο υπολογισμός διορθώσεων στα βάρη του κάθε Νευρώνα ξεχωριστά. Έτσι, γίνεται χρήση της ανάστροφης μετάδοσης λάθους (back propagation), η οποία βασίζεται στον γενικευμένο κανόνα Δέλτα (generalized Delta rule). Ο γενικευμένος κανόνας Δέλτα επιτρέπει τον καθορισμό του ποσοστού του συνολικού σφάλματος που αντιστοιχεί στα βάρη του κάθε Νευρώνα ακόμη και αν αυτά ανήκουν σε κρυφά επίπεδα των οποίων η επιθυμητή έξοδος δεν είναι γνωστή.

Για την εκπαίδευση του Multi Layer Percepton (MLP), γίνεται αρχικά ένα πρόσθιο πέρασμα (forward pass). Συγκεκριμένα εισάγονται στην είσοδο δεδομένα από ένα διάνυσμα εκπαίδευσης και οι νευρώνες στο επίπεδο εισόδου παράγουν ένα αποτέλεσμα, το οποίο στην συνέχεια αποτελεί είσοδο στο επόμενο επίπεδο. Η συγκεκριμένη διαδικασία επαναλαμβάνεται διαδοχικά για τα επόμενα κρυφά επίπεδα, μέχρι το επίπεδο εξόδου. Έτσι, θεωρώντας ως n τον αριθμό των Νευρώνων του επιπέδου εισόδου, η είσοδος ενός κρυφού Νευρώνα j, δίνεται από την σχέση:

$$input_j = \sum_{i=0}^{n} v_{ij} x_i \tag{2.15}$$

όπου v_{ij} το βάρος της σύνδεσης μεταξύ των νευρώνων i, j και x_i το σήμα εισόδου του νευρώνα i. Αντίστοιχα, η έξοδος του συγκεκριμένου Νευρώνα, θα είναι

$$out_j = \phi(\sum_{i=0}^{n} v_{ij} x_i) \tag{2.16}$$

,η οποία θα προωθηθεί στους νευρώνες του επόμενου επιπέδου.

Η παραπάνω σχέση, χρησιμοποιείται για τους Νευρώνες όλων των επιπέδων, εκτός από το επίπεδο εισόδου, μιας και οι Νευρώνες αυτοί χρησιμοποιούνται μόνο για την μεταφορά των δεδομένων

εκπαίδευσης στα επόμενα επίπεδα. Αντίστοιχα, θεωρώντας ως q, τον αριθμό των Νευρώνων του κάθε κρυφού επιπέδου j οι παραπάνω σχέσεις για το επίπεδο εξόδου k με m Νευρώνες γίνονται

$$input_k = \sum_{j=1}^{q} w_{jk} x_i \tag{2.17}$$

$$out_k = \phi(\sum_{j=1}^{m} w_{jk} x_j) \tag{2.18}$$

Είναι σημαντικό επίσης να αναφερθεί, πως η συνάρτηση ενεργοποίησης για τα δίκτυα που εκπαιδεύονται με ανάστροφη μετάδοση λάθους, πρέπει να είναι μη γραμμική αλλά παράλληλα μονότονα αύξουσα και παραγωγίσιμη για όλες τις τιμές εισόδου. [IB06]

Έτσι, το δίκτυο ξεκινά τους υπολογισμούς όπως και ένα απλό percepton, με τυχαίες τιμές στα βάρη των νευρώνων. Αντίστοιχα, θα υπολογιστεί και το σφάλμα εξόδου για τους Νευρώνες του επιπέδου εξόδου. Αφού υπολογιστεί το ακριβές σφάλμα στο επίπεδο εξόδου, για το οποίο είναι γνωστό το επιθυμητό αποτέλεσμα, είναι δυνατό να χρησιμοποιηθεί ο γενικευμένος κανόνας Δέλτα, για να προσαρμοστούν κατάλληλα οι τιμές των βαρών του προηγούμενου επιπέδου.

Συγκεκριμένα, για k ως το επίπεδο εξόδου, και j το αμέσως προηγούμενο, μπορούμε αρχικά να ορίσουμε την ποσότητα , σύμφωνα με τον γενικευμένο κανόνα Δέλτα

$$\delta_k = (t_k - out_k)\phi\,'(input_k) \tag{2.19}$$

, ώστε να υπολογιστεί η μεταβολή στα βάρη

$$\Delta w_{jk} = \alpha \cdot \delta_k \cdot out_j \tag{2.20}$$

Αντίστοιχα, για το κρυφό επίπεδο j

$$\delta_j = \phi\,'(input_j) \sum_{k=1}^{m} w_{jk}\delta_k \tag{2.21}$$

$$\Delta w_{ij} = \alpha \cdot \delta_j \cdot x_i \tag{2.22}$$

Με τον παραπάνω τρόπο, διαμορφώνονται τα βάρη όλων των κρυφών επιπέδων μέχρι το επίπεδο εισόδου, όπως φαίνεται στην εξίσωση 2.22. Αυτή η διαδικασία προσαρμογής των βαρών ονομάζεται ανάστροφο πέρασμα (backward pass) ή ανάστροφη μετάδοση (back propagation)

Έτσι, ο αλγόριθμος της ανάστροφης λάθους, ελαχιστοποιεί το μέσο τετραγωνικό σφάλμα μεταξύ της εξόδου του δικτύου και της επιθυμητής εξόδου, για διανύσματα p του συνόλου εκπαίδευσης

$$MSE = \frac{1}{p} \sum_{p} \sum_{k=1}^{m} t_k^{(p)} - out_k^{(p)} \tag{2.23}$$

Στην ουσία, με την παραπάνω διαδικασία, αναζητείται το ολικό ελάχιστο της συνάρτησης σφάλματος. Η διόρθωση που γίνεται κάθε φορά προσπαθεί να κάνει εκείνες τις αλλαγές που θα μειώσουν το σφάλμα τοπικά. Αυτό είναι πιθανό να δημιουργήσει πρόβλημα, αφού υπάρχει ο κίνδυνος να εγκλωβιστεί το δίκτυο σε τοπικά ελάχιστα. Για να αντιμετωπιστεί το συγκεκριμένο πρόβλημα μπορεί να γίνει αρχικοποίηση των βαρών με διαφορετικό τρόπο, όμως πάλι υπάρχει ο κίνδυνος να βρεθεί το δίκτυο σε άλλα τοπικά ελάχιστα.

Ένα ακόμη σημαντικό ζήτημα που είναι πιθανό να προκύψει, είναι το δίκτυο να παραλύσει (network paralysis). Στην συγκεκριμένη περίπτωση ένα ή περισσότερα βάρη μπορεί να αποκτήσει πολύ υψηλή τιμή, με αποτέλεσμα να μην μεταβάλλεται σημαντικά στις επόμενες επαναλήψεις [Hayk09] Το πρόβλημα αυτό μπορεί να επιλυθεί αυξάνοντας τον ρυθμό μάθησης a

Υποπροσαρμογή και Υπερπροσαρμογή

Η εκπαίδευση του δικτύου με τα δεδομένα εισόδου, γίνεται ανά κύκλους. Έτσι, κάθε φορά το δίκτυο λαμβάνει ένα διάνυσμα εισόδου και αναπροσαρμόζει τα βάρη του με την τεχνική του αλγορίθμου της ανάστροφης μετάδοσης λάθους. Σε ένα Νευρωνικό Δίκτυο, είναι αρκετά συχνό το φαινόμενο της υποπροσαρμογής (underfitting) ή της υπερπροσαρμογής (overfitting). Συγκεκριμένα, ένα Νευρωνικό Δίκτυο που δεν είναι αρκετά περίπλοκο μπορεί να μην καταφέρει να μοντελοποιήσει σωστά τα δεδομένα εκπαίδευσης. Αντίθετα, ένα αρκετά πολύπλοκο Νευρωνικό δίκτυο, μπορεί να μοντελοποιήσει τα δεδομένα εκπαίδευσης, σε τέτοιον βαθμό, που να είναι αδύνατη η γενίκευση του μοντέλου σε δεδομένα που δεν έχει ξανασυναντήσει. Για να περιοριστεί η εμφάνιση αυτών των προβλημάτων, είναι σημαντική η ύπαρξη μεγάλου αριθμού δεδομένων εκπαίδευσης.

Κεφάλαιο 3

Αλγόριθμοι Ενισχυτικής Μάθησης

Όπως αναφέρθηκε εν συντομία προηγουμένως, η ενισχυτική μάθηση είναι μία περιοχή της Μηχανικής Μάθησης, στην οποία ένας πράκτορας λαμβάνει δράση σε ένα Περιβάλλον, προσπαθώντας να μεγιστοποιήσει το άθροισμα των ανταμοιβών που λαμβάνει για τις δράσεις του. Το Περιβάλλον, αναπαρίσταται σαν μία Διαδικασία Απόφασης Markov (Markov Decision Process) για την επίλυση της οποίας χρησιμοποιούνται συχνά τεχνικές δυναμικού προγραμματισμού [Wier12]

Η βασική διαφορά της Ενισχυτική Μάθηση με τον Δυναμικό Προγραμματισμό, είναι ότι η Ενισχυτική Μάθηση στοχεύει σε αρκετά μεγάλες MDPs που η εφαρμογή κλασσικών αλγορίθμων Δυναμικού Προγραμματισμού θα ήταν ανέφικτη. Χρησιμοποιείται, επίσης, σε περιπτώσεις όπου δεν είναι γνωστή η συνάρτηση μετάβασης P ή και συνάρτηση ανταμοιβής R.

3.1 Q-Learning

Μία από τις βασικότερες τεχνικές της Ενισχυτικής Μάθησης, αποτελεί η τεχνική του Q-Learning [Li18] η οποία μπορεί να χρησιμοποιηθεί για την εύρεση της βέλτιστης πολιτικής π^*, όταν δεν είναι εφικτή η εφαρμογή αλγορίθμων Δυναμικού Προγραμματισμού.

Συγκεκριμένα, είναι χρήσιμο να οριστεί πρώτα μία νέα συνάρτηση $Q(s, a)$,γνωστή και ως Q-Value, στην οποία ο Πράκτορας ξεκινάει από την κατάσταση s, επιλέγει την δράση a και στην συνέχεια θεωρείται ότι δρα βέλτιστα. Έτσι, η συνάρτηση $Q(s, a)$ υπολογίζει την ποιότητα ενός συνδυασμού κατάστασης και δράσης (s, a) ως εξής:

$$Q^*(s, a) = \sum_{s'} P(s'|s, a)(R(s, a, s') + \gamma \max_{a'} Q^*(s', a')) \tag{3.1}$$

Στην παραπάνω εξίσωση, λοιπόν, ο Πράκτορας όταν βρίσκεται στην κατάσταση s, επιλέγει την δράση a, αυτή τον οδηγεί στην κατάσταση s' και εκεί επιλέγει την βέλτιστη δράση a' που θα μεγιστοποιήσει την συνάρτηση Q-Value.

Αντίστοιχα, με την μέθοδο του Value Iteration, μπορούμε πλέον να εφαρμόσουμε Q-Value Iteration, για να υπολογίζουμε τα Q-Values.

$$Q^*_{k+1}(s, a) = \sum_{s'} P(s'|s, a)(R(s, a, s') + \gamma \max_{a'} Q_k(s', a')) \tag{3.2}$$

Για την επίλυση της συγκεκριμένης εξίσωσης και των υπολογισμό των Q-Values, είναι απαραίτητο να είναι εξαρχής γνωστές όλες οι δυνατές μεταβάσεις από κάθε κατάσταση s σε μία άλλη κατάσταση s', καθώς και οι πιθανότητες ώστε αυτές να συμβούν. Παρόλα αυτά, σε πολλά προβλήματα αυτές οι ποσότητες δεν είναι γνωστές. Ισοδύναμα, η παραπάνω εξίσωση μπορεί να γραφθεί χρησιμοποιώντας την αναμενόμενη τιμή:

$$Q^*_{k+1}(s, a) = E_{s' \sim P(s'|s,a)}[R(s, a, s') + \gamma \max_{a'} Q_k(s', a')] \tag{3.3}$$

Επειδή, λοιπόν ο Πράκτορας δεν γνωρίζει την πιθανότητα μετάβασης στην κατάσταση s', θα βασιστεί στην εμπειρία από αποκτά από την συνεχή αλληλεπίδρασή με το Περιβάλλον, ώστε να υπολογίσει τα Q-Values. Ορίζοντας σαν στόχο, μετά από μία αλληλεπίδραση, την εκτίμηση,

$$target(s') = R(s, a, s') + \gamma \max_{a'} Q_k(s', a') \tag{3.4}$$

στην οποία, λαμβάνει την ανταμοιβή $R(s, a, s')$ θεωρεί πως στην συνέχεια δρα βέλτιστα και υπολογίζει σταδιακά τα Q-Values

$$Q^*_{k+1}(s, a) = (1 - \alpha)Q_k(s, a) + a[target(s')] \tag{3.5}$$

Έτσι, τελικά ο Αλγόριθμος του Q-Learning διαμορφώνεται ως εξής:

Start with $Q_0(s, a)$ for all s, a.
Get initial state s
For k = 1, 2, ... till convergence
    Sample action a, get next state s'
    If s' is terminal:
        $\text{target} = R(s, a, s')$
        Sample new initial state s'
    else:
        $\text{target} = R(s, a, s') + \gamma \max_{a'} Q_k(s', a')$
    $Q_{k+1}(s, a) \leftarrow (1 - \alpha)Q_k(s, a) + \alpha\,[\text{target}]$
    $s \leftarrow s'$

Σχήμα 3.1: Αλγόριθμος Q-Learning

3.1.1 Επίδραση Μεταβλητών

Ρυθμός εκμάθησης

Ο ρυθμός εκμάθησης, α, καθορίζει σε ποιο βαθμό η νέα γνώση για την ποσότητα $Q(s, a)$, θα υπερισχύσει των παλιών πληροφοριών. Μία τιμή 0, οδηγεί τον Πράκτορα στην χρήση μόνο της ήδη υπάρχουσας γνώσης, χωρίς να μαθαίνει περαιτέρω στις επόμενες επαναλήψεις. Αντίθετα, αν δοθεί τιμή 1, τότε ο Πράκτορας δεν θα λάβει καθόλου υπόψιν του την προϋπάρχουσα γνώση. Σε ένα ντετερμινιστικό περιβάλλον, η βέλτιστη τιμή για τον ρυθμό εκμάθησης, θα ήταν $\alpha = 1$. Όταν όμως το πρόβλημα είναι στοχαστικό, απαιτείται για την σύγκλιση του αλγορίθμου μία τιμή που να χρησιμοποιεί σε μεγάλο βαθμό την προϋπάρχουσα γνώση, ώστε να επιτευχθεί τελικά η σύγκλιση. Έτσι, στην πράξη χρησιμοποιείται πολύ συχνά τιμή κοντά στο 0, της τάξεως του 0.01.

Εκπτωτικός Παράγοντας

Ο εκπτωτικός παράγοντας γ, έχει ακριβώς τον ίδιο ρόλο, με αυτό στην MDP. Συγκεκριμένα, καθορίζει την αξία των μελλοντικών ανταμοιβών. Εάν χρησιμοποιηθεί η τιμή 0, τότε ο Πράκτορας, δεν λαμβάνει καθόλου υπόψιν του τις μελλοντικές αμοιβές. Αντίθετα, μία τιμή ίση με 1, θα οδηγήσει τον Πράκτορα να προσπαθεί συνεχώς να πετύχει μία καλύτερη μελλοντική ανταμοιβή, γεγονός που καθιστά αδύνατη την σύγκλιση του αλγορίθμου. Στην πράξη χρησιμοποιούνται συχνά τιμές μικρότερες αλλά κοντά στην μονάδα, ώστε σταδιακά ο Αλγόριθμος να συγκλίνει [Fran18].

3.1.2 Προβλήματα Q-Learning

Όπως φαίνεται, από τον Αλγόριθμο (3.1), ο Πράκτορας είναι απαραίτητο να αποθηκεύει τις τιμές των Q-Values, ώστε να μπορέσει να τις χρησιμοποιήσει μετέπειτα. Έτσι, σε προβλήματα που ο χώρος καταστάσεων είναι πολύ μεγάλος, απαιτείται τεράστια ποσότητα μνήμης για να μπορέσει ο Αλγόριθμος να λειτουργήσει σωστά. Αυτό, καθιστά πολλές φορές αδύνατη την χρήση του.

Q-Table		Actions		
		Buy	Hold	Sell
States	S1	0	0	0
	.	.	.	.
	.	.	.	.
	.	.	.	.
	S17	0	0	0
	.	.	.	.
	.	.	.	.
	.	.	.	.
	S42	0	0	0
	.	.	.	.
	.	.	.	.
	.	.	.	.

Σχήμα 3.2: Πίνακας Q-Table, πριν την εκμάθηση.

Q-Table		Actions		
		Buy	Hold	Sell
States	S1	0	0	0
	.	.	.	.
	.	.	.	.
	.	.	.	.
	S17	4.30182	1.98274	3.92011
	.	.	.	.
	.	.	.	.
	.	.	.	.
	S42	3.902819	2.56147	4.965412
	.	.	.	.
	.	.	.	.
	.	.	.	.

Σχήμα 3.3: Πίνακας Q-Table, μετά την εκμάθηση.

Επίσης, αποθηκεύοντας απλώς τις τιμές των Q-Values, είναι αδύνατο για τον Πράκτορα, να γνωρίζει πώς θα συμπεριφερθεί σε καταστάσεις που δεν συνάντησε στην εκπαίδευσή του.

3.1.3 Επίλυση Προβλημάτων

Τα προβλήματα που αναφέρθηκαν, είναι σημαντικό να επιλυθούν ώστε να χρησιμοποιηθεί ο παραπάνω αλγόριθμος σε καταστάσεις που συναντώνται στον πραγματικό κόσμο. Η επίλυσή σου καθίσταται εφικτή χρησιμοποιώντας τις παρακάτω τεχνικές.

Ποσοτικοποίηση (Quantization)

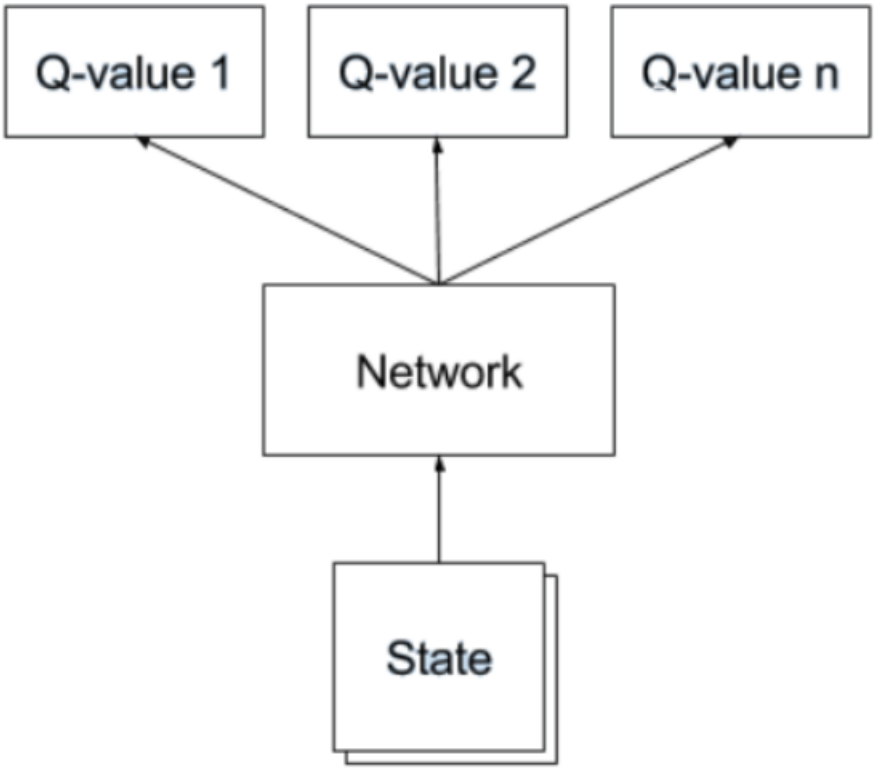
Αρχικά, είναι σημαντικό, να μειωθούν οι χώροι των δράσεων και των καταστάσεων όσο το δυνατών περισσότερο. Με αυτό τον τρόπο, ίσως γίνει εφικτή η δημιουργία πίνακα μνήμης για την αποθήκευση των Q-Values. Επιπλέον, η σωστή Ποσοτικοποίηση θα βοηθήσει σημαντικά στην μείωση του χρόνου σύγκλισης του Αλγορίθμου, όταν γίνεται χρήση συνάρτησης προσέγγισης αφού πλέον ο Πράκτορας θα μπορεί πιο εύκολα να επιλέξει την κατάλληλη δράση.

Προσέγγιση συνάρτησης (Function approximation)

Το δεύτερο πρόβλημα, δηλαδή η αντιμετώπιση άγνωστων καταστάσεων, μπορεί να επιλυθεί με την χρήση ενός συστήματος, το οποίο αντί να αποθηκεύει τις τιμές Q-Values, θα προσπαθήσει να προσεγγίσει την συνάρτηση $Q(s, a)$, για κάθε πιθανό ζεύγος (s, a), με βάση την εμπειρία που έχει αποκτήσει από καταστάσεις που έχει επισκεφθεί. Αυτή, ακριβώς είναι η θεμελιώδης ιδέα των Νευρωνικών Δικτύων.

3.2 Q Network

Αξιοποιώντας την γνώση για τα Νευρωνικά δίκτυα, είναι εφικτή η τροποποίηση του Αλγορίθμου Q-Learning, ώστε να επιλυθούν τα προβλήματα που παρουσίαζε η απλή αποθήκευση των Q-Values.



Σχήμα 3.4: Αρχιτεκτονική με χρήση Νευρωνικού Δικτύου

Συγκεκριμένα, αρκεί πλέον να εκφράσουμε την συνάρτηση $Q(s, a)$ ως προς κάποιες παραμέτρους θ. Έτσι, μπορούμε να ορίσουμε ξανά τον στόχο ως:

$$target(s') = R(s, a, s') + \gamma \max_{a'} Q_{\theta k}(s', a') \tag{3.6}$$

Έτσι, πλέον σε κάθε επανάληψη, αντί να γίνεται απευθείας ανανέωση των τιμών Q-Value, γίνεται ανανέωση των παραμέτρων $theta$ χρησιμοποιώντας Gradient Decent.

$$\theta_{k+1} = \theta_k - \alpha \nabla_\theta E_{s' \sim P(s'|s,a)}[(Q_{\theta(s,a)} - target(s'))^2]|_{\theta=\theta k} \tag{3.7}$$

Ο τελικός αλγόριθμος, λοιπόν, για το Q-Network, παρουσιάζεται παρακάτω.

Start with $Q_0(s, a)$ for all s, a.

Get initial state s

For k = 1, 2, ... till convergence

  Sample action a, get next state s'

  If s' is terminal:

    $\text{target} = R(s, a, s')$

    Sample new initial state s'

  else:

    $\text{target} = R(s, a, s') + \gamma \max_{a'} Q_k(s', a')$

$\theta_{k+1} \leftarrow \theta_k - \alpha \nabla_\theta \mathbb{E}_{s' \sim P(s'|s,a)} \left[(Q_\theta(s, a) - \text{target}(s'))^2 \right] \Big|_{\theta=\theta_k}$

$s \leftarrow s'$

Σχήμα 3.5: Αλγόριθμος Q-Network

Με αυτό τον τρόπο, το Νευρωνικό δίκτυο, λαμβάνει κάθε φορά την κατάσταση s και υπολογίζει τόσες τιμές Q-values, όσες και οι διαθέσιμες δράσεις.

Ο Πράκτορας, θεωρώντας ότι σε κάθε κατάσταση s δρα βέλτιστα, θα επιλέξει εκείνη την δράση a με το μεγαλύτερο $Q(s, a)$.

3.3 Deep Q Network (DQN)

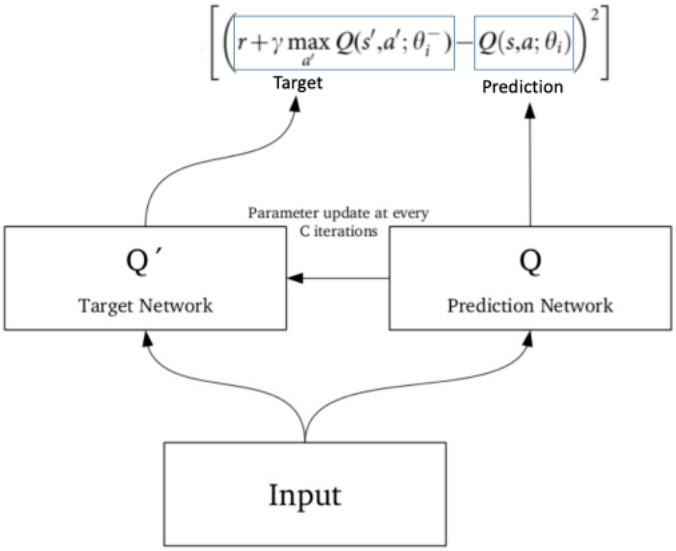
Όπως έχει ήδη αναφερθεί, η χρήση ενός Νευρωνικού Δικτύου με περισσότερα επίπεδα, μπορεί να ενισχύσει σημαντικά την εξαγωγή χαρακτηριστικών , προσεγγίζοντας πιο πολύπλοκες συναρτήσεις.

Παρόλα αυτά, με την χρήση της παραπάνω Αρχιτεκτονικής, για την προσέγγιση του στόχου (3.6) προκύπτουν δύο σημαντικά προβλήματα στην λειτουργία του Νευρωνικού Δικτύου.

- Ο Στόχος δεν είναι σταθερός,
- Υπάρχει ισχυρή συσχέτιση των δεδομένων

Αρχικά, είναι εμφανές από τις εξισώσεις (3.6) και (3.7) πως καθώς γίνεται ανανέωση των παραμέτρων θ για την προσέγγιση του στόχου, μετακινείται και ο ίδιος ο στόχος. Ανανεώνοντας, δηλαδή, τις παραμέτρους για την προσαρμογή του Q-Value σε μία κατάσταση s στην οποία λήφθηκε η δράση a, ανανεώνονται τα Q-Values και των υπολοίπων δράσεων στην ίδια κατάσταση. Κάτι τέτοιο δεν συμβαίνει στην μηχανική μάθηση, στην οποία ο στόχος θεωρείται σταθερός και εξασφαλίζει αντίστοιχα σταθερότητα κατά την εκπαίδευση του Νευρωνικού Δικτύου. Έτσι, η συνεχής μετακίνηση του στόχου μπορεί να δημιουργήσει βρόγχους ανατροφοδότησης, μην επιτρέποντας στο Νευρωνικό Δίκτυο να συγκλίνει.

Επιπλέον, καθώς οι δράσεις του Πράκτορα καθορίζουν την επόμενη κατάσταση του Περιβάλλοντος, οι είσοδοι στο Νευρωνικό έχουν άμεση συσχέτιση μεταξύ τους. Κάτι τέτοιο, μπορεί να προκαλέσει τον εγκλωβισμό του Νευρωνικού Δικτύου σε τοπικά βέλτιστα.



Σχήμα 3.6: Χρήση Target Network

Για την επίλυση των παραπάνω προβλημάτων, είναι σημαντική η εισαγωγή δύο επιπλέον συστατικών [Mnih13] . Συγκεκριμένα, θα γίνει χρήση:

- Ενός δεύτερου Νευρωνικού Δικτύου, Target Nerwork
- Μίας αποθήκης εμπειριών (Experience Replay), οι οποίες θα αποτελέσουν τις νέες εισόδους στο Νευρωνικό Δίκτυο.

Target Network

Το Target Network, θα βοηθήσει στο πρόβλημα του μετακινούμενου στόχου. Συγκεκριμένα, γίνεται χρήση ενός Νευρωνικού Δικτύου $\widehat{Q}$ με παραμέτρους θ^-. Το Target Network, έχει ακριβώς την ίδια αρχιτεκτονική με το βασικό Νευρωνικό Δίκτυο Q, Prediction Network. Έτσι, για τον υπολογισμό του στόχου (3.6) γίνεται πλέον η χρήση του Target Network και εφαρμόζεται Gradient Descent ως προς τις παραμέτρους θ ανάμεσα στην πρόβλεψη που λήφθηκε από το Prediction Network Q με παραμέτρους θ και τον Στόχο που υπολογίσαμε από το Target Network με παραμέτρους θ^-.

$$target(s') = R(s, a, s') + \gamma \max_{a'} \widehat{Q}(s', a', \theta^-) \tag{3.8}$$

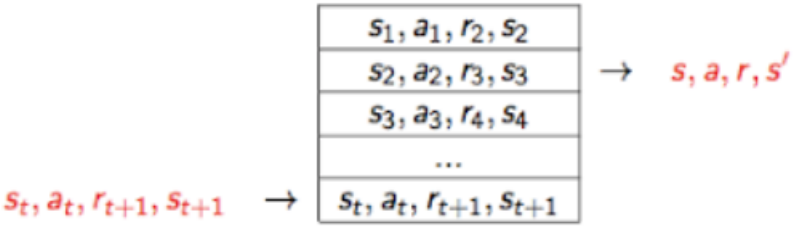
$$\theta_{k+1} = \theta_k - \alpha \nabla_\theta [(Q_{\theta(s,a)} - target(s'))^2]|_{\theta=\theta k} \tag{3.9}$$

Ανά τακτά χρονικά διαστήματα C, θα ανανεώνονται οι παράμετροι θ^- του Target Network, με βάση τις νέες παραμέτρους θ του Prediction Network.

Experience Replay

Για επίλυση του προβλήματος της συσχέτισης των δεδομένων εισόδου στο Νευρωνικό Δίκτυο, είναι απαραίτητη η δημιουργία νέων εμπειριών για τον Πράκτορα, στις οποίες θα συμπεριλαμβάνονται τα αποτελέσματα από την εκάστοτε δράση του, όμως οι ίδιες οι εμπειρίες θα είναι ανεξάρτητες μεταξύ τους. Κάθε εμπειρία, λοιπόν, θα αποτελείται από μία κατάσταση s τη δράση a που λήφθηκε σε αυτή την κατάσταση, την νέα κατάσταση s' που οδήγησε η επιλεγμένη δράση, καθώς και την επιβράβευση που λήφθηκε για την μετάβαση στη νέα κατάσταση. Έτσι, οι εμπειρίες, θα είναι της μορφής $< s, a, r, s' >$.

Αυτές οι εμπειρίες, θα αποθηκεύονται σε έναν πίνακα και σε κάθε επανάληψη, θα επιλέγονται τυχαία και ομοιόμορφα, ομάδες εμπειριών, με τις οποίες θα εκπαιδεύεται το Νευρωνικό Δίκτυο. Παράλληλα, καθώς προστίθενται νέες εμπειρίες στον πίνακα σε κάθε επανάληψη t, οι παλιές θα αφαιρούνται.



Σχήμα 3.7: Experience Replay Buffer

Εφαρμόζοντας τις παραπάνω τροποποιήσεις, ο Αλγόριθμος για το Deep Q-Network, διαμορφώνεται όπως φαίνεται παρακάτω.

Initialize replay memory D to capacity N
Initialize action-value function Q with random weights θ
Initialize target action-value function $\hat{Q}$ with weights $\theta^{-} = \theta$
For episode = 1, M **do**
  Initialize sequence $s_1 = \{x_1\}$ and preprocessed sequence $\phi_1 = \phi(s_1)$
  For t = 1,T **do**
    With probability ε select a random action a_t
    otherwise select $a_t = \text{argmax}_a Q(\phi(s_t), a; \theta)$
    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 D
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from D
    Set $y_j = \begin{cases} r_j & \text{if episode terminates at step j+1} \\ r_j + \gamma \max_{a'} \hat{Q}(\phi_{j+1}, a'; \theta^{-}) & \text{otherwise} \end{cases}$
    Perform a gradient descent step on $\left(y_j - Q(\phi_j, a_j; \theta)\right)^2$ with respect to the network parameters θ
    Every C steps reset $\hat{Q} = Q$
  End For
End For

Σχήμα 3.8: Αλγόριθμος DQN, με Experience Replay

Όπως θα φανεί στο κεφάλαιο των Πειραμάτων, η προσθήκη των Target Network και Experience Replay, βοήθησε σημαντικά στην επίλυση των προβλημάτων που δημιουργήθηκαν από τη χρήση του Νευρωνικού Δικτύου. Παρόλα αυτά υπάρχουν ακόμη μερικές βελτιώσεις που θα μπορούσαν να γίνουν στον Αλγόριθμο, ώστε να επιτευχθεί καλύτερη απόδοση.

Εξερεύνηση ή Αξιοποίηση (Exploration vs Exploitaion)

Για την καλύτερη προσέγγιση της επιθυμητής συνάρτησης, χρησιμοποιείται επιπλέον μία ποσότητα ϵ. Αυτή η ποσότητα ορίζει την πιθανότητα να ληφθεί μία τυχαία δράση έναντι της δράσης του Πράκτορα. Έτσι, ο Πράκτορας, θα μπορέσει να επισκεφθεί, "εξερευνήσει", καταστάσεις στις οποίες είναι πιθανό να μην μπορούσε να βρεθεί χρησιμοποιώντας απλά την υπάρχουσα γνώση του. Η ποσότητα αυτή, επιλέγεται συνήθως με τρόπο τέτοιο ώστε στα αρχικά στάδια εκμάθησης να επιτρέψει

στον Πράκτορα να εξερευνήσει σημαντικά το Περιβάλλον, ενώ σταδιακά μειώνεται μέχρι τελικά να φτάσει τιμή κοντά στο 0.1, όπου και μένει σταθερή. [Arul17]

3.4 Double DQN

Η χρήση του Target Network, βοήθησε ώστε να αντιμετωπιστεί το πρόβλημα του συνεχούς μετακινούμενου στόχου. Ο πράκτορας, όμως, βασίζεται στην εμπειρία που έχει αποκτήσει στην εκάστοτε επανάληψη για την επιλογή του βέλτιστου Q-Value. Έτσι, είναι πολύ πιθανό, ειδικά στην αρχή της εκπαίδευσης, που δεν έχει ακόμα αρκετή πληροφορία, να εκτιμήσει λανθασμένα, γεγονός που μπορεί στην πορεία της εκπαίδευσης να οδηγήσει σε υπερεκτίμηση των τιμών του Q-Value. Η λύση, στο συγκεκριμένο πρόβλημα, είναι η επιλογή της δράσης a' να γίνεται από το βασικό Prediction Network, όμως το Q-Value για την δράση αυτή να λαμβάνεται ένα δεύτερο Νευρωνικό δίκτυο ίδιας αρχιτεκτονικής [Hass10] και συγκεκριμένα από το Target Network [Hass16]. Έτσι, η εξίσωση (3.8), μετατρέπεται ως εξής:

$$a^{max} = \arg\max_{a'} Q(s', a', \theta) \tag{3.10}$$

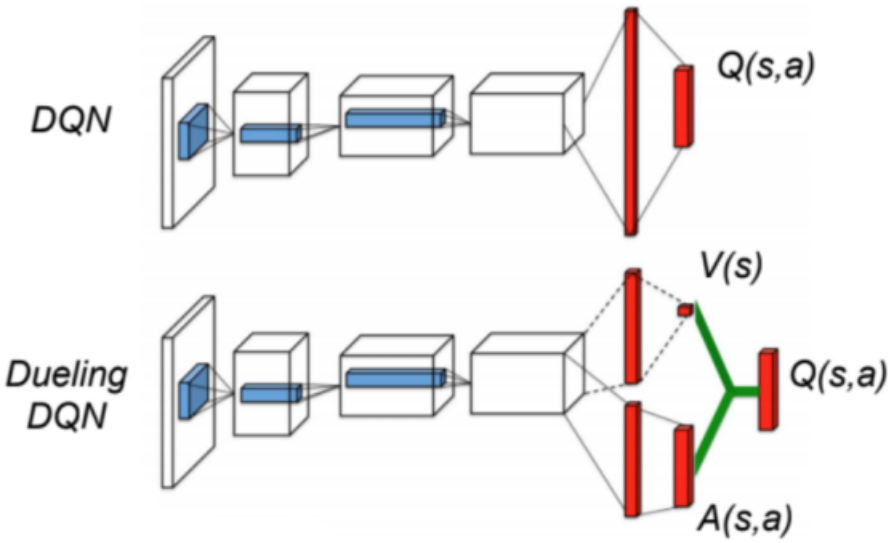
$$target(s') = R(s, a, s') + \gamma \widehat{Q}(s', a^{max}, \theta^{-}) \tag{3.11}$$

3.5 Dueling DQN

Τέλος, μία επιπλέον βελτίωση θα μπορούσε να ενισχύσει τον παραπάνω Αλγόριθμο. Αρχικά χρησιμοποιώντας τις εξισώσεις (2.5) και (3.1) στις οποίες έχουν οριστεί οι ποσότητες $Q(s, a)$ και $V(s, a)$ μπορεί να οριστεί μία νέα ποσότητα (s, a), η οποία ονομάζεται Πλεονέκτημα (Advantage).

$$A^{\pi}(s, a) = Q^{\pi}(s, a) - V^{\pi}(s) \tag{3.12}$$

Η συγκεκριμένη ποσότητα, εκφράζει πόσο χειρότερη είναι η κάθε δράση a, από την καλύτερη δράση στην κατάσταση s, δεδομένης μίας πολιτικής π.



Σχήμα 3.9: Αρχιτεκνονική Dueling DQN [Wang15]

Κεφάλαιο 4

Χρηματιστήριο

Ως Χρηματιστήριο νοείται μία οργανωμένη αγορά στην οποία οι ενδιαφερόμενοι, μπορούν να πραγματοποιούν αγοροπωλησίες κινητών αξιών, όπως μερίδια κεφαλαίου ανωνύμων εταιριών (μετοχές), τραπεζικά, κρατικά ή άλλα ομόλογα. Το Χρηματιστήριο αποτελεί έναν σύγχρονο τρόπο άντλησης κεφαλαίου για τις Εταιρίες, έτσι ώστε να επεκταθούν ή να πραγματοποιήσουν περαιτέρω επενδύσεις.

4.1 Μετοχή

Ο όρος μετοχή (stock), αναφέρεται στα κομμάτια στα οποία αποφασίζει μία Εταιρία, να διαιρέσει την Ιδιοκτησία της. Έτσι, όταν κάποιος αγοράζει κομμάτι (share) μίας μετοχής, του ανήκει και ένα αντίστοιχο κομμάτι της συγκεκριμένης Εταιρίας.

Η αγορά μετοχών μίας Εταιρίας αποτελεί επένδυση για τον αγοραστή που προσδοκεί στη διανομή μερίσματος, αλλά και στην αύξηση της τιμής της μετοχής. Έτσι, μπορεί να αποκομίσει κέρδος, πουλώντας τις μετοχές που κατέχει, όταν η τιμή τους φτάσει σε ικανοποιητικό επίπεδο, ανώτερο της τιμής αγοράς τους.

4.2 Τιμή

Οι τιμές των μετοχών διαμορφώνονται από τον κανόνα της προσφοράς και της ζήτησης. Με τον όρο ”προσφορά”, περιγράφεται η ποσότητα των μετοχών που βρίσκονται σε μία συγκεκριμένη χρονική στιγμή προς πώληση. Αντίστοιχα, ο όρος ”ζήτηση”, εκφράζει την ποσότητα των μετοχών της Εταιρίας, οι οποίες είναι επιθυμητό να αγοραστούν εκείνη την χρονική στιγμή. Έτσι, όταν γίνεται αναφορά στην όρο ”Συναλλαγή” νοείται η μεταφορά ή η ανταλλαγή για χρήματα, μετοχών από έναν πωλητή (seller), σε έναν αγοραστή (buyer). Αυτό απαιτεί ο αγοραστής και ο πωλητής να συμφωνήσουν στην τιμή της αγοραπωλησίας. Η τιμή στην οποία έγινε η τελευταία αγοραπωλησία αποτελεί και την τιμή της μετοχής.

4.3 Όγκος

Ο όγκος των συναλλαγών της μετοχής μίας Εταιρίας, υποδηλώνει τον αριθμό των συναλλαγών που έγιναν μία συγκεκριμένη χρονική στιγμή. Πρόκειται για ένα χαρακτηριστικό, το οποίο εκφράζει την ικανότητα ”ρευστοποίησης” μίας μετοχής, στην αγορά. Έτσι, ένας υψηλός όγκος συναλλαγών σε μία ημέρα, υποδηλώνει ότι θα είναι εφικτή η πώληση των μετοχών που κατέχει ο Επενδυτής. Οι μετοχές που έχουν μεγάλο όγκο συναλλαγών είναι πιο αξιόπιστες και προτιμούνται από τους επενδυτές ακόμη κι αν παρέχουν μικρότερα κέρδη από άλλες.

4.4 Αλγοριθμικές Συναλλαγές

Οι αλγοριθμικές συναλλαγές είναι μια διαδικασία εκτέλεσης εντολών αγοράς ή πώλησης με χρήση αυτοματοποιημένων και προ-προγραμματισμένων στρατηγικών για τη διαχείριση μεταβλητών όπως η τιμή, το χρονοδιάγραμμα και ο όγκος. [Chen19] Ένας αλγόριθμος αποτελείται από ένα σύνολο οδηγιών για την επίλυση ενός προβλήματος. Ένα υπολογιστικό σύστημα στέλνει μικρές μερίδες εντολών για αγορά ή πώληση μετοχών. Για τις αλγοριθμικές συναλλαγές χρησιμοποιούνται συνήθως σύνθετοι τύποι, σε συνδυασμό με μαθηματικά μοντέλα και ανθρώπινη εποπτεία, για να λάβει το σύστημα τις κατάλληλες αποφάσεις.

4.5 Στρατηγική Συναλλαγών

Μια στρατηγική συναλλαγών στο Χρηματιστήριο είναι η μέθοδος αγοράς και πώλησης μετοχών που βασίζεται σε προκαθορισμένους κανόνες που χρησιμοποιούνται για τη λήψη αποφάσεων.

Μια στρατηγική συναλλαγών περιλαμβάνει ένα καλά εξεταζόμενο επενδυτικό και εμπορικό σχέδιο που καθορίζει επενδυτικούς στόχους, ανοχή κινδύνου, χρονικό ορίζοντα και φορολογικές επιπτώσεις. Ο σχεδιασμός για συναλλαγές περιλαμβάνει την ανάπτυξη μεθόδων που αφορούν την αγορά ή πώληση μετοχών , ομολόγων ή άλλων κατηγοριών αξιών. Οι βασικές στρατηγικές διαπραγμάτευσης λαμβάνουν υπόψη τους θεμελιώδεις παράγοντες. Για παράδειγμα, ένας επενδυτής μπορεί να έχει ένα σύνολο κριτηρίων επιλογής για να δημιουργήσει μια λίστα ευκαιριών. Τα κριτήρια αυτά αναπτύσσονται αναλύοντας παράγοντες όπως η αύξηση των εσόδων και η κερδοφορία μίας Εταιρίας.

Ποσοτικές Συναλλαγές (Quantitive Trading)

Το Quantitive Trading βασίζεται στην ποσοτική ανάλυση χαρακτηριστικών της μετοχής, σε μαθηματικούς υπολογισμούς και στην αριθμητική κρίση για τον εντοπισμό των εμπορικών ευκαιριών. Η τιμή και ο όγκος είναι δύο από τις πιο κοινές εισροές δεδομένων που χρησιμοποιούνται στην ποσοτική ανάλυση.

Δεδομένου ότι οι χρηματιστηριακές συναλλαγές πραγματοποιούνται γενικά από χρηματοπιστωτικά ιδρύματα και αμοιβαία κεφάλαια αντιστάθμισης κινδύνου (Hegde Funds), οι συναλλαγές είναι συνήθως μεγάλες και μπορεί να περιλαμβάνουν την αγορά και πώληση εκατοντάδων χιλιάδων μετοχών και άλλων τίτλων. Ωστόσο, το Quantitive Trading χρησιμοποιείται συχνότερα από μεμονωμένους επενδυτές.

Οι ”ποσοτικοί επενδυτές” (Quantitive Traders) επωφελούνται από τη σύγχρονη τεχνολογία, τα μαθηματικά και τη διαθεσιμότητα ολοκληρωμένων δεδομένων για τη λήψη αποφάσεων ορθολογικών συναλλαγών. Συνήθως ακολουθούν μία συγκεκριμένα ροή εργασίας για την δημιουργία κατάλληλης στρατηγικής.

Ροή Εργασίας για Δημιουργία Στρατηγικής Συναλλαγών

Η αγοραπωλησία μετοχών αποτελεί μία χρονοβόρα διαδικασία, κατά την οποία ο Επενδυτής, πρέπει να αποφασίσει σε ποιες χρονικές στιγμές θα αγοράσει ή θα πουλήσει τις μετοχές που διαθέτει, ώστε τελικά να αποκομίσει κέρδος. Η συγκεκριμένη διαδικασία περιλαμβάνει συνήθως αρκετά στάδια, ώστε να μπορέσει να πραγματοποιηθεί σωστά.

Αρχικά, φτιάχνεται ένα μοντέλο επιβλεπόμενης μάθησης ώστε χρησιμοποιώντας την ιστορική πορεία της μετοχής μιας Εταιρίας, το μοντέλο να προβλέψει τις μελλοντικές τιμές της. Φυσικά αντίστοιχα μοντέλα, μπορούν να δημιουργηθούν ώστε να γίνει πρόβλεψη και άλλων ποιοτικών ή ποσοτικών δεικτών που είναι πιθανό να ενδιαφέρουν τον επενδυτή. Στη συνέχεια, ο επενδυτής με βάση την τωρινή κατάσταση της αγοράς και των προβλέψεων του μοντέλου που δημιούργησε, καλείται να κατασκευάσει μία στρατηγική, αποτελούμενη από κανόνες σχετικά με το πότε θα πρέπει να πραγματοποιηθούν οι αγοροπωλησίες. Φυσικά, για τους κανόνες αυτούς, συχνά χρησιμοποιούνται κατώφλια, για την λήψη των αποφάσεων, τα οποία θα πρέπει και αυτά να βελτιστοποιηθούν.

Έπειτα, η στρατηγική δοκιμάζεται ξανά σε ιστορικές τιμές, ώστε να διαπιστωθεί εάν στις ιστορικές τιμές αποδίδει (backtesting). Εάν αποδίδει γίνεται προσπάθεια, για επιπλέον βελτιστοποίηση των τιμών του μοντέλου και των κατωφλίων. Φυσικά σε όλη αυτή την διαδικασία, υπάρχει ο κίνδυνος overfitting του δικτύου επιβλεπόμενης μάθησης κάτι που μπορεί να οδηγήσει σε καταστροφικά αποτελέσματα σε πραγματικές συναλλαγές.

Πλεονεκτήματα και Μειονεκτήματα του Quantitive Trading

Το βασικό πλεονέκτημα του Quantitive Trading είναι ότι επιτρέπει τη βέλτιστη χρήση δεδομένων που έχουν υποβληθεί σε δοκιμή και εξαλείφει τη συναισθηματική λήψη αποφάσεων κατά τη διάρκεια των συναλλαγών. [Shar19]

Παρόλα αυτά, οι κερδοφόρες στρατηγικές συναλλαγών είναι δύσκολο να αναπτυχθούν, και υπάρχει ο κίνδυνος να καταστούν υπερβολικά εξαρτημένες από τους κανόνες που έχουν δημιουργηθεί κατά τη παραπάνω ροή εργασίας. Η στρατηγική μπορεί να έχει λειτουργήσει καλά θεωρητικά βάσει παλαιών δεδομένων της αγοράς, αλλά οι προηγούμενες επιδόσεις δεν εγγυώνται τη μελλοντική επιτυχία σε πραγματικές συνθήκες της αγοράς, οι οποίες μπορεί να διαφέρουν σημαντικά από την περίοδο δοκιμών.Ένα ακόμη μειονέκτημα της ποσοτικής στρατηγικής είναι ότι έχει περιορισμένη χρήση. Μια ποσοτική στρατηγική συναλλαγών χάνει την αποτελεσματικότητά της όταν αλλάξουν οι συνθήκες της αγοράς. Έτσι, καθώς η αγορά μεταβάλλεται συνεχώς όλη η παραπάνω διαδικασία είναι απαραίτητο να επαναλαμβάνεται σε τακτά χρονικά διαστήματα, ώστε να προσαρμόζεται εκ νέου το μοντέλο επιβλεπόνεης μάθησης για την πρόβλεψη των μελλοντικών ποσοτήτων αλλά και η στρατηγική του επενδυτή ώστε να παραμείνει κερδοφόρα.

Κεφάλαιο 5

Εφαρμογή Ενισχυτικής Μάθησης στις Συναλλαγές

5.1 Κίνητρο

Όπως αναφέρθηκε στο προηγούμενο Κεφάλαιο, η δημιουργία μίας στρατηγικής για αγοραπωλησία μετοχών, απαιτεί πολλά χρονοβόρα στάδια, τα οποία όχι μόνο χρειάζεται να επαναλαμβάνονται ανά τακτά χρονικά διαστήματα για την διατήρηση μίας κερδοφόρας στρατηγικής, αλλά παράλληλα εξαρτώνται από την δημιουργία κανόνων, που θα πρέπει συνεχώς να προσαρμόζονται και να εξελίσσονται.

Η Ενισχυτική Μάθηση, προσφέρει ένα μοντέλο, η χρήση του οποίου καθιστά ικανή την επίλυση αυτών των προβλημάτων. Συγκεκριμένα, ένας Πράκτορας μπορεί να εκπαιδευτεί χρησιμοποιώντας τις ιστορικές τιμές των μετοχών της Εταιρίας που ενδιαφέρει τον επενδυτή. Κατά τη διάρκεια της εκπαίδευσής του, ο Πράκτορας, δοκιμάζοντας να επενδύσει σε ένα Περιβάλλον που προσομοιώνει τη λειτουργία του Χρηματιστηρίου, διαμορφώνει μία στρατηγική για τον τρόπο με τον οποίο θα πρέπει να πραγματοποιήσει τις συναλλαγές του. Ο τρόπος αυτός, είναι άμεσα συνδεδεμένος με τον δείκτη που θέλει να μεγιστοποιήσει ο επενδυτής. Για παράδειγμα, εάν ο επενδυτής επιθυμεί μέσα από τις αγοροπωλησίες του να μεγιστοποιήσει τα κέρδη του, αρκεί να δημιουργήσει μία κατάλληλη συνάρτηση ανταμοιβής, η οποία θα υποδείξει στον Πράκτορα, πότε οι συναλλαγές που πραγματοποιεί, οδηγούν στο επιθυμητό αποτέλεσμα.

Στην παραπάνω διαδικασία, δεν θα υπάρχει πλέον η ανάγκη για την κατασκευή συγκεκριμένων κανόνων και κατωφλίων. Αυτά θα τα δημιουργήσει ο ίδιος ο Πράκτορας μέσα από την εκπαίδευσή του. Παράλληλα, ο τρόπος με τον οποίο εκπαιδεύεται ο Πράκτορας, του δίνει την δυνατότητα να προσαρμόζει την στρατηγική στις αλλαγές που συμβαίνουν στην αγορά.

5.2 Σχεδιαστικές Επιλογές

5.2.1 Περιβάλλον

Αρχικά, είναι απαραίτητο να διαμορφωθεί το Περιβάλλον κατάλληλα, ώστε να προσομοιώσει τη λειτουργία του Χρηματιστηρίου. Παράλληλα, είναι αναγκαίο το Περιβάλλον να προσφέρει στον Πράκτορα την πληροφορία που θα ήθελε να διαθέτει ένας επενδυτής ώστε να πραγματοποιήσει αγοροπωλησίες μετοχών μίας Εταιρίας.

Σύμφωνα, λοιπόν, με την ανάλυση που έγινε στο Κεφάλαιο 2, για να προσημειωθεί η Χρηματιστηριακή αγορά, απαιτείται γνώση καταστάσεων, για παράδειγμα η εξέλιξη των τιμών της μετοχής, που δεν εξαρτώνται από τις πράξεις του Πράκτορα. Το Περιβάλλον, λοιπόν, μπορεί να χαρακτηριστεί ως Μερικώς Προσβάσιμο.

Έτσι, το Περιβάλλον σχεδιάστηκε με τέτοιο τρόπο, ώστε να γνωρίζει εξαρχής όλα τα δεδομένα της εκπαίδευσης, όμως να αποκρύπτει οποιαδήποτε μελλοντική πληροφορία. Με αυτό το σκεπτικό, γνωστοποιεί στον Πράκτορα μόνο πληροφορίες σχετικά με την τωρινή χρονική στιγμή t καθώς και χρήσιμες ιστορικές πληροφορίες οι οποίες διαμορφώνουν τις Καταστάσεις του Περιβάλλοντος.

5.2.2 Καταστάσεις

Μία κατάσταση του Περιβάλλοντος s την χρονική στιγμή t , επιλέχθηκε να αποτελείται από τις παρακάτω πληροφορίες

- Το κεφάλαιο (Capital), που έχει στην διάθεσή του για επένδυση ο Πράκτορας.
- Ο αριθμός των μετοχών (Shares) που έχει αγοράσει ο Πράκτορας.
- Η καθαρή αξία (Net Worth), που υπολογίζεται παρακάτω από το άθροισμα του κεφαλαίου και την τωρινή αξία των μετοχών που διαθέτει ο Πράκτορας.
- Την τωρινή τιμή της μετοχής καθώς και ιστορικές τιμές των τελευταίων 90 ημερών.

Λαμβάνοντας υπόψιν την δυσκολία εκμάθησης, που χαρακτηρίζει ένα Μερικώς Προσβάσιμο Περιβάλλον, έγινε προσπάθεια καλής ποσοτικοποίησης του χώρου καταστάσεων. Έτσι, ο Πράκτορας θα λαμβάνει ημερήσια τιμή της μετοχής, ώστε να πραγματοποιήσει τις αγοροπωλησίες του.

Τα παραπάνω χαρακτηριστικά επιλέχθηκαν, με τρόπο τέτοιο ώστε ο Πράκτορας να έχει πλήρη εικόνα της κατάστασης στην οποία βρίσκεται, δηλαδή εικόνα αντίστοιχη με αυτή που θα ήθελε να έχει ένα επενδυτής. Επίσης, καθώς δεν είναι εφικτή η χρήση πίνακα, για την διατήρηση των καταστάσεων και κατ' επέκταση των ιστορικών τιμών της μετοχής, είναι απαραίτητο το περιβάλλον να γνωστοποιεί στον Πράκτορα ιστορικές τιμές, ώστε να μπορέσει να διακρίνει πρότυπα συμπεριφοράς (patterns) κατά την εκπαίδευσή του.

Στην αρχική κατάσταση, επιλέχθηκε ο Πράκτορας να έχει στη διάθεσή του 10.000 $ και μηδενικό αριθμό μετοχών,

Η καθαρή του αξία, υπολογίζεται επίσης σε κάθε χρονική στιγμή ως εξής:

$$NetWorth(t) = Capital(t) + p(t) * \sharp shares(t) \tag{5.1}$$

,όπου p_t η τιμή της μετοχής τη χρονική στιγμή t και $\sharp shares(t)$ ο αριθμός των μετοχών που διαθέτει ο Πράκτορας εκείνη τη στιγμή.

5.2.3 Δράσεις

Με αντίστοιχη λογική, επιλέχθηκε και ο χώρος των δράσεων. Έτσι, το Περιβάλλον, επιτρέπει στον Πράκτορα, τις παρακάτω δράσεις:

- Αγορά (Buy), κατά την οποία ο Πράκτορας προσπαθεί να αγοράσει όσο το δυνατόν περισσότερες μετοχές, με το κεφάλαιο που διαθέτει.
- Πώληση (Sell), κατά την οποία ο Πράκτορας προσπαθεί να πουλήσει όσο το δυνατόν περισσότερες μετοχές, από αυτές που διαθέτει.
- Στάση (Hold), κατά την οποία ο Πράκτορας έχει την δυνατότητα να μην πραγματοποιήσει αγοραπωλησία.

Σύμφωνα, λοιπόν, με τον χώρο καταστάσεων, ο Πράκτορας θα μπορεί να εκτελεί μία αγοραπωλησία την ημέρα, για παράδειγμα να αγοράσει ή να πουλήσει μετοχές στην ημερήσια τιμή.

5.2.4 Συναρτήσεις Ανταμοιβής

Όπως έχει αναφερθεί, η συνάρτηση ανταμοιβής είναι από τα πιο σημαντικά συστατικά ενός προβλήματος Ενισχυτικής Μάθησης. Έχει καθοριστικό ρόλο στην συμπεριφορά του Πράκτορα καθ' όλη τη διάρκεια της εκπαίδευσης, αφού είναι το μόνο σήμα που ενημερώνει τον Πράκτορα σχετικά με το πόσο καλές είναι οι δράσεις που επιλέγει.

Παρόλα αυτά, αρκετά συχνά χρήζει ιδιαίτερης δυσκολίας η εύρεση της κατάλληλης συνάρτησης, για την επίτευξη του επιθυμητού αποτελέσματος.

Στην συγκεκριμένη μελέτη, θεωρήσαμε πώς ο επενδυτής επιθυμεί να δημιουργήσει μία στρατηγική με βάση τα κέρδη που αποκομίζει. Έτσι, κατόπιν μελέτης και κατασκευής διαφόρων συναρτήσεων, τρεις παρουσίασαν μεγαλύτερο ενδιαφέρον.

Κέρδος (Profit)

Η πρώτη συνάρτηση, υπολογίστηκε με βάση τα κέρδη του Πράκτορα, σε κάθε αγοραπωλησία. Έτσι, όταν ο Πράκτορας πουλήσει μετοχές και παράξει θετικό κέρδος, λαμβάνει σαν ανταμοιβή ένα θετικό σήμα, ενώ όταν παράξει αρνητικό κέρδος λαμβάνει ένα αρνητικό σήμα. Σε οποιαδήποτε άλλη περίπτωση, ο πράκτορας λαμβάνει μηδενικό σήμα.

Το Κέρδος μπορεί να υπολογιστεί σε σχέση με την τιμή που αγόρασε την κάθε μετοχή και την τιμή που την πούλησε. Έτσι, εάν αγόρασε μία μετοχή στην τιμή p_a και την πούλησε στην τωρινή τιμή p_b, τότε το κέρδος που θα αποκομίσει από την αγοραπωλησία της θα προκύψει από την διαφορά των τιμών της αγοράς και της πώλησης

$$\text{Κέρδος Μετοχής} = p_b - p_a \tag{5.2}$$

Άρα, τελικά αν είχε στην κατοχή του k μετοχές που την κάθε μία την αγόρασε στην τιμή p_i και τις πούλησε όλες στην τιμή p_b, τότε το συνολικό κέρδος που αποκόμισε, είναι

$$\text{Κέρδος} = \sum_{i=1}^{k} (p_b - p_i) \tag{5.3}$$

Έτσι, η συγκεκριμένη συνάρτηση έχει την παρακάτω απλή μορφή:

$$R(s, a, s') = \begin{cases} 1, & \text{εάν παράχθηκε θετικό Κέρδος} \\ -1, & \text{εάν παράχθηκε αρνητικό Κέρδος} \\ 0, & \text{σε οποιαδήποτε άλλη περίπτωση} \end{cases}$$

Είναι εμφανές ότι η συγκεκριμένη συνάρτηση, παράγει σήμα θετικό ή αρνητικό μόνο όταν γίνεται κάποια πώληση. Για τις δράσεις της Αγοράς και της Στάσης, ο Πράκτορας, δεν λαμβάνει ανταμοιβή. Πρέπει, λοιπόν, κατά τη διάρκεια της εκπαίδευσης να κατανοήσει αρχικά ότι είναι απαραίτητο να αγοράσει μετοχές για να μπορέσει να τις πουλήσει και να βγάλει κέρδος, καθώς επίσης και πότε είναι χρήσιμο να αξιοποιεί την ενέργεια της Στάσης, ώστε να μεγιστοποιήσει μελλοντικά την ανταμοιβή του.

Επιστροφή στην Επένδυση (Return on Investment)

Στην συγκεκριμένη συνάρτηση, ο Πράκτορας δεν επιβραβεύεται απευθείας για τα κέρδη του. Αντίθετα, η ανταμοιβή υπολογίζεται με βάση τα κέρδη του ως προς τα χρήματα που επένδυσε. Επίσης, έγινε προσπάθεια, ώστε να δοθεί στον Πράκτορα μία εικόνα σχετικά με το πόσο καλές ήταν η πωλήσεις του, ώστε να μπορέσει να εξάγει συμπεράσματα και για τις αγορές του.

Αρχικά, κατ' αναλογία με την προηγούμενη συνάρτηση, υπολογίζουμε το Return on Investment (RoI), σαν κλάσμα του Κέρδους και των χρημάτων που χρειάστηκε να επενδυθούν για να αγοραστούν οι μετοχές που απέφεραν το κέρδος αυτό.

$$RoI = \frac{\text{Κέρδος}}{\sum_{i=1}^{k} p_i}$$

Επιπλέον, για να ενισχύσουμε την αποκόμιση καλής αναλογίας Κέρδους προς Χρήματα που επενδύθηκαν, θα πάρουμε τον λογάριθμο της ποσότητας $1 + RoI$. Έτσι, παράγουμε μία θετική τιμή η οποία μπορεί να προσφέρει στον Πράκτορα πληροφορία για την ποιότητα των επιλογών του. Είναι

επίσης, προφανές ότι το RoI, θα έχει σχετικά χαμηλή τιμή συνήθως κάτω της μονάδας. Ενδεικτικά, για να έχει κάποιος RoI ίσο με τη μονάδα, θα πρέπει να έχει 100% κέρδος, δηλαδή όσα χρήματα επένδυσε, άλλα τόσα να κερδίσει, κάτι που είναι ιδιαίτερα δύσκολο σε τέτοιες αγορές. Για να αντισταθμίσουμε, λοιπόν, τα αρνητικά σήματα -1, που προκύπτουν εάν ο Πράκτορας παράξει ζημία, θα δοθεί τελικά σαν θετικό σήμα η ποσότητα $1 + log(1 + RoI)$. Έτσι, τελικά η συνάρτηση διαμορφώνεται ως εξής:

$$R(s, a, s') = \begin{cases} 1 + log(1 + RoI), & \text{εάν παράχθηκε θετικό RoI} \\ -1, & \text{εάν παράχθηκε αρνητικό RoI} \\ 0, & \text{σε οποιαδήποτε άλλη περίπτωση} \end{cases}$$

Ποσοστό Καθαρής Αξίας (Net Worth Percentage)

Τέλος, γίνεται προσπάθεια για περαιτέρω ομαλοποίηση της συνάρτησης ανταμοιβής, ώστε ο Πράκτορας να λαμβάνει κατάλληλες ανταμοιβές για όλες του τις δράσεις. Για να επιτευχθεί αυτό, χρησιμοποιείται το ποσοστό καθαρού ημερήσιου κέρδους, σαν σήμα ανταμοιβής. Έτσι, χρησιμοποιώντας την εξίσωση (5.1), η συνάρτηση επιβράβευσης, διαμορφώνεται ως εξής:

$$R(s, a, s') = 100 * \frac{NetWorth(t) - NetWorth(t-1)}{NetWorth(t-1)}$$

Η συνάρτηση αυτή, δίνει μία καλή εικόνα στο Πράκτορα για την πορεία της αξίας του σε σχέση με τις μετοχές που έχει αγοράσει. Έτσι, εάν δεν έχει αγοράσει μετοχές, λαμβάνει μηδενικό σήμα. Αντίθετα, εάν έχει αγοράσει, λαμβάνει σήμα ανάλογο με την πορεία της τιμής της μετοχής, δηλαδή θετικό σήμα εάν η μετοχή έχει ανοδική πορεία, ενώ αρνητικό εάν έχει καθοδική.

5.3 Υλοποίηση

Περιβάλλον

Αρχικά, το Περιβάλλον υλοποιήθηκε σύμφωνα με τις οδηγίες και την διεπαφή gym που έχει ορίσει η OpenAI [Broc16]. Έτσι, το Περιβάλλον έχει την παρακάτω μορφή

```
class TradeEnv:
  """Custom Environment that follows gym guidelines"""

  def __init__(self, data, history_t, initial_capital, reward_type):
    # Initialize the neccessary variables

  def step(self, action):
    # Execute one time step within the environment
    ...
    return state, reward, done, info

  def reset(self):
    # Reset the state of the environment to an initial state
    ...
  def render(self):
    # (Optional) Render the environment to the screen
    ...
```

Σχήμα 5.1: Διεπαφή Περιβάλλοντος σύμφωνα με τα πρότυπα της OpenAI.

Σύμφωνα με την παραπάνω διεπαφή, έγινε η υλοποίηση των συναρτήσεων ώστε να προσομοιωθεί το Περιβάλλον όπως ορίστηκε στην προηγούμενη ενότητα. Το Περιβάλλον, θα λάβει ως ορίσματα:

- Τα δεδομένα ($data$) της μετοχής, που είναι απαραίτητα για όλο το διάστημα της εκπαίδευσης ή του testing
- Τον αριθμό των ιστορικών ημερών ($histoy\_t$) για τις οποίες είναι απαραίτητο να γνωρίζει ο Πράκτορας την τιμή της μετοχής.
- Το αρχικό κεφάλαιο ($initial\_capital$) που μπορεί να διαθέσει ο Πράκτορας, για τις αγορές του.
- Τον τύπο ($reward\_typel$) της συνάρτησης επιβράβευσης που είναι επιθυμητό να χρησιμοποιηθεί.

Η παράμετρος ($reward\_typel$), μπορεί να πάρει μόνο μία τιμή από τις "profit", "roi" ή "net_worthl", η κάθε μία εκ των οποίων, παραπέμπει στην χρήση της αντίστοιχης συνάρτησης αυτές που περιγράφηκαν παραπάνω.

Η λειτουργία των συναρτήσεων init και reset μπορεί να γίνει αμέσως αντιληπτή, αφού θα χρησιμοποιηθούν για να αρχικοποιήσουν το περιβάλλον ή να το επαναφέρουν στην αρχική του κατάσταση, αντίστοιχα. Η συνάρτηση step, είναι υπεύθυνη για την δημιουργία της ανταμοιβής, ανάλογα με την εκάστοτε κατάσταση και δράση του Πράκτορα καθώς και για την δημιουργία της νέας κατάστασης στην οποία ο Πράκτορας θα μεταβεί.

Πράκτορες

Για την εκτέλεση επαρκών πειραμάτων και την εξαγωγή συμπερασμάτων, υλοποιήθηκαν αρχικά οι Αλγόριθμοι, που αναφέρονται στο Κεφάλαιο 3. Συγκεκριμένα, έγινε η υλοποίηση των

- Q-Network
- Deep Q Network (DQN)
- Double και Dueling Deep Q Network (DDDQN)

Ο καθένας από αυτούς, έδρασε σαν Πράκτορας στο Περιβάλλον, πραγματοποιώντας Συναλλαγές και διαμορφώνοντας την στρατηγική του για να αποκτά κέρδος.

```
class Agent:
  def __init__(self,
               session,
               scope_name,
               input_size,
               hidden_layer_sizes,
               output_size,
               learning_rate):

  with tf.variable_scope(self.scope_name):
  # Build the Agent with the desired layers
  # Keep the Agent's Tensors into a scope

  def predict(self, state):
  # Predict the next action based on states

  def update(self, state, y):
  # Calculate the loss update, based on the desired target y
```

Σχήμα 5.2: Διεπαφή Πράκτορα.

Κάθε ένας από τους πράκτορες έχει την δική του δομή και χαρακτηριστικά, όπως ορίστηκαν στο Κεφάλαιο 3. Έτσι, για το Q-Network, χρησιμοποιήθηκε ένα απλό Δίκτυο Percepton, με 100 Νευρώνες.

DQN

Για τον Πράκτορα που έκανε χρήση DQN, επεκτάθηκε η αρχιτεκτονική του Q Network με δύο Multi Layer Perceptons, 3 κρυφών επιπέδων το καθένα και 100 Νευρώνες στο κάθε επίπεδο. Για τον DQN, η διεπαφή του Πράκτορα επεκτάθηκε αντίστοιχα με δύο επιπλέον συναρτήσεις, ώστε να υποστηριχθεί η εισαγωγή των Target Networks και Experience Replay.

```
@staticmethod
  def create_copy_operations(source_scope, dest_scope):
    # Copy the parameters theta of the Main Network, to the Target Network.

    return result
```

Σχήμα 5.3: Υποστήριξη Target Network.

Double and Dueling DQN

Για τους Double και Dueling DQN, έγιναν οι αλλαγές που αναφέρονται στο Κεφάλαιο 3, στην υπάρχουσα αρχιτεκτονική του DQN. Έτσι, κατά την δημιουργία του Πράκτορα, το τελευταίο επίπεδο διαχωρίστηκε σε δύο επιμέρους επίπεδα, με 100 νευρώνες το καθένα. Το πρώτο υπολογίζει την ποσότητα $V(s)$ για το κάθε state και συνεπώς έχει μία έξοδο. Το δεύτερο υπολογίζει την ποσότητα του Advantage $A(a, s)$ και συνεπώς έχει τρεις εξόδους. Κάθε μία από τις εξόδους αυτές, αθροίζεται με την ποσότητα $V(s)$, ώστε τελικά να παραχθούν τα τρία Q-values, που δίνει σαν έξοδο, το δίκτυο.

```
def train_dqn(main_dqn, target_dqn, mini_batch):
  # mini_batch = a list of experiences in the form of
  #(state, action, reward, next_state, done, info)

  # Retrieve from the mini batch the states, actions, rewards, next_states, done

  # Predict the target actions using the target Network and the next states

  # Calculate the Target using the max Q Value from the Target Network

  # Predict the current actions using the Main DQN Network

  # Update the Parametes of the Main DQN Network

  return loss
```

Σχήμα 5.4: Υποστήριξη Target DQN.

Μεταβλητές Εκπαίδευσης

Τέλος, κάθε ένας από τους παραπάνω πράκτορες εκπαιδεύτηκε ανά εποχές, μέχρι τελικά να επιτευχθεί η σύγκλιση.

```
for epoch in epochs:
    state = env.reset()
    while not done:
        # Choose a random action or predict with Probability e
        # Receive th next_state, reward, done, info from the env.step(action)
        # Update the Q-Values based on the chosen Algorithm
        # Consider as state the next_state
```

Σχήμα 5.5: Εκπαίδευση Πράκτορα ανά εποχές.

Μία από τις πιο σημαντικές παραμέτρους κατά την εκπαίδευση είναι η Πιθανότητα e, η οποία δίνει την δυνατότητα στον Πράκτορα, να εξερευνήσει νέες καταστάσεις, λαμβάνοντας τυχαίες δράσεις. Η ποσότητα αυτή αρχικοποιήθηκε στην μονάδα και ορίστηκε ένας παράγοντας epsilon decrease ώστε το $epsilon$ σταδιακά να μειώνεται και ο Πράκτορας να έχει μεγαλύτερο έλεγχο των δράσεων που επιλέγονται.

Έτσι, η ποσότητα $epsilon$, αρχικοποιήθηκε στην τιμή 1 ,ενώ η ποσότητα epsilon decrease επιλέχθηκε ως $10^(-3)$. Το $epsilon$, μειώνεται σταδιακά μέχρι να φτάσει τιμή μικρότερη του 0.1, στην οποία επιτρέπει την πραγματοποίηση τυχαίων δράσεων που αποτελούν περίπου το 10% των συνολικών δυνατών δράσεων σε κάθε εποχή. Η ποσότητα αυτή είναι κρίσιμη, ώστε ο Πράκτορας να μπορέσει να εξερευνήσει νέες καταστάσεις πέραν αυτών που που έχει εξερευνήσει λόγω των δράσεών του.

Κεφάλαιο 6

Πειράματα και Αποτελέσματα

6.1 Δεδομένα

Για την εκπαίδευση του Πράκτορα, θα χρησιμοποιηθούν η τιμή της Google, για μία διετή περίοδο από την περίοδο 2014 έως και το 2016.

Σχήμα 6.1: Μετοχή της Google.

Όπως φαίνεται από την πορεία της μετοχής, η περίοδος αυτή παρουσιάζει διαστήματα με διαφορετικές διακυμάνσεις στην τιμή της μετοχής. Συγκεκριμένα, η μετοχή έχει καθοδική πορεία, από τον Ιούλιο του 2014 έως τον Ιανουάριο του 2015, ανοδική πορεία από τον Ιούλιο του 2015 έως τον Ιανουάριο του 2016. Αντίστοιχα, στο διάστημα από τον Ιανουάριο του 2015 εώς τον Ιούλιο του 2015, η τιμή της μετοχής παραμένει σχετικά σταθερή. Έτσι, ο Πράκτορας, θα μπορέσει να εξερευνήσει αυτές τις περιπτώσεις και να μάθει να διαμορφώνει την στρατηγική του ανάλογα.

Η επόμενη περίοδος, θα χρησιμοποιηθεί ώστε να αξιολογηθεί ο βαθμός στον οποίο ο Πράκτορας κατάφερε τελικά να δημιουργήσει μία αποδοτική στρατηγική για να επενδύει. Από το διάγραμμα, είναι εμφανές ότι η συγκεκριμένη δοκιμαστική περίοδος έχει καθαρά ανοδική πορεία. Έτσι, για την καλύτερη αξιολόγηση και εξαγωγή συμπερασμάτων, ο ήδη εκπαιδευμένος Πράκτορας θα δοκιμαστεί και στην μετοχή της Acuity, μία μετοχή η οποία την ίδια περίοδο έχει καθοδική πορεία.

Σχήμα 6.2: Μετοχή της Acuity Brands Inc.

Είναι σημαντικό σε αυτό το σημείο να αναφερθεί, πως δεν έχει σημασία ποια μετοχή χρησιμοποιείται για εκμάθηση και ποια για τις πραγματικές επενδύσεις. Στόχος του Πράκτορα κατά τη διάρκεια της εκπαίδευσής του, δεν είναι να προβλέψει την μελλοντική τιμή της μετοχής, αλλά να ανακαλύψει πρότυπα συμπεριφοράς (patterns) και με ποιο τρόπο θα πρέπει να προσαρμόζει την στρατηγική του ανάλογα με την διαφορετική πορεία της μετοχής ώστε να μεγιστοποιεί κάθε φορά την επιβράβευσή του.

6.2 Εκπαίδευση και Αξιολόγηση

Αρχικά, κάθε ένας από τους Πράκτορες θα εκπαιδευτεί στο Περιβάλλον που υλοποιήσαμε. Θα εκπαιδεύσουμε τους Πράκτορες σε εποχές, όπου η κάθε μία αποτελείται από την πορεία της μετοχής για δύο πραγματικά χρόνια. Έτσι, κάθε Πράκτορας, αφού ολοκληρώσει την εκπαίδευσή του, θα έχει διαμορφώσει μία στρατηγική με την οποία θα πραγματοποιεί τις επενδύσεις του.

Για να αξιολογήσουμε την εκπαίδευση των Πρακτόρων, θα εξετάσουμε δύο σημαντικές γραφικές παραστάσεις. Αρχικά, θα ελέγξουμε τον τρόπο μεταβολής του loss, ώστε να δούμε εάν ο Πράκτορας συγκλίνει και με ποιον τρόπο. Στην συνέχεια, θα ελέγξουμε τον τρόπο με τον οποίο ο Πράκτορας λάμβανε τις ανταμοιβές ανάλογα με τις δράσεις του κατά τη διάρκεια της εκπαίδευσης.

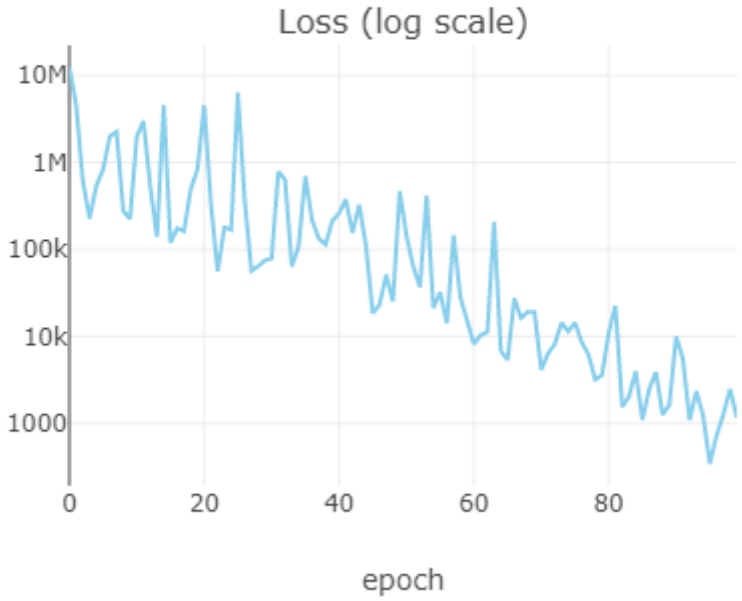
6.2.1 Q-Network

Παρακάτω μπορούμε να δούμε τα αποτελέσματα για τον Q-Netowrk. Ο συγκεκριμένος αλγόριθμος, φαίνεται να συγκλίνει, όμως μπορούν να παρατηρηθούν αρκετά απότομες μεταβολές.

Είναι, όμως, σημαντικό να εξετάσουμε και τις αθροιστικές ανταμοιβές που λαμβάνει ο συγκεκριμένος Πράκτορας σε αυτό το διάστημα.

Αρχικά, είναι, σημαντικό να σημειωθεί πως το μεγάλο πλήθος των αρνητικών σημάτων είναι αναμενόμενο. Συγκεκριμένα, κατά την διάρκεια της εκπαίδευσης, όπως αναφέρθηκε, χρησιμοποιείται ο παράγοντας $epsilon$, ώστε να μπορέσει ο Πράκτορας να επισκεφθεί νέες καταστάσεις ανεξάρτητα από τις μέχρι τώρα δράσεις του.

Αυτό έχει ως αποτέλεσμα, μία τυχαία ενέργεια, να οδηγεί το Πράκτορα σε καταστάσεις με σημαντική παραγωγή αρνητικών σημάτων, τα οποία ο Πράκτορας θα προσπαθήσει να αντισταθμίσει με τις δικές του ενέργειες. Έτσι, από την γραφική των επιβραβεύσεων, μπορούμε να εξάγουμε συμπεράσματα για τον τρόπο με τον οποίο καταφέρνει ο Πράκτορας να αντεπεξέλθει και να βελτιώσει σταδιακά την στρατηγική του, αντισταθμίζοντας παράλληλα τα αρνητικά σήματα



Σχήμα 6.3: Σύγλιση Q - Network

Σχήμα 6.4: Αθροιστικές Ανταμοιβές Q - Network ανά εποχή

Φαίνεται, λοιπόν, ότι οι πολύ απότομες μεταβολές στο loss, συνοδεύονται και από αντίστοιχα

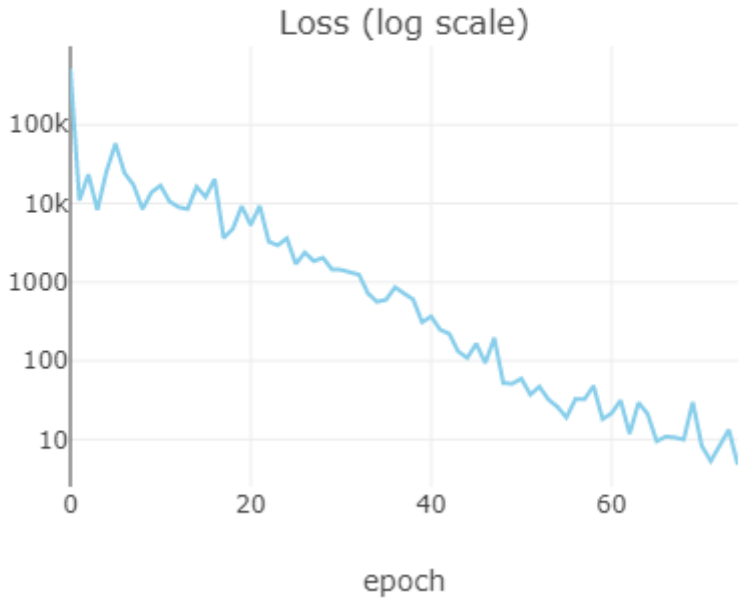
απότομες μεταβολές στον τρόπο με τον οποίο ο πράκτορας δρα και λαμβάνει τις ανταμοιβές του. Έτσι, παρόλο που ο Q-Learning, όπως φαίνεται στον παρακάτω πίνακα, καταφέρνει να αποκομίσει κάποια κέρδη στην μετοχή της Google, μπορούμε να συμπεράνουμε πως το Δίκτυο, δεν μπορεί να κατανοήσει επαρκώς την πληροφορία που λαμβάνει σαν είσοδο, για να δράσει κατάλληλα. Έτσι, καθίσται απαραίτητη η χρήση ενός βελτιωμένου μοντέλου Πράκτορα, όπως είναι ο DQN,

Reward Function	Google	Acuity
Profit	1524	-1465
Roi	2032	-1947
Net Worth	2501	-2988

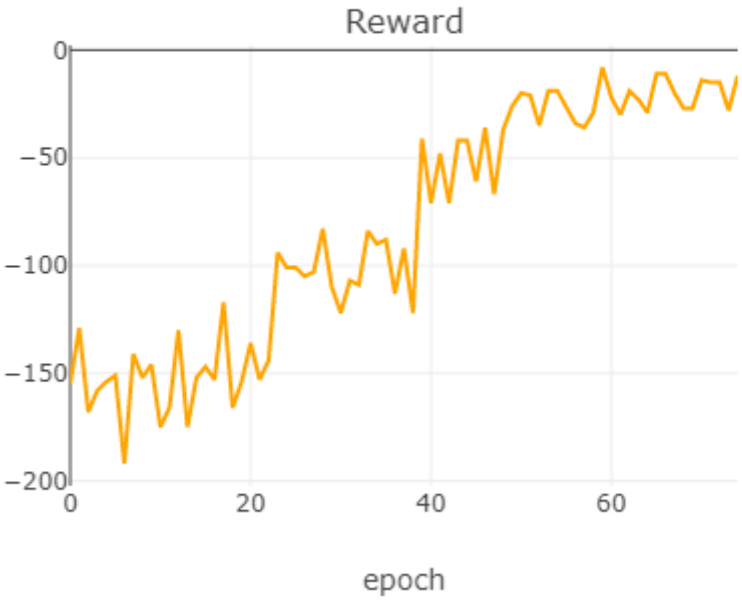
Πίνακας 6.1: Q-Network Κέρδη

6.2.2 Deep Q Network

Αντίστοιχα, για τον DQN είναι σημαντικό να αξιολογήσουμε το τρόπο σύγκλισης καθώς και τον τρόπο με τον οποίο λαμβάνει τις ανταμοιβές του ο Πράκτορας.



Σχήμα 6.5: Σύγκλιση Deep Q Network



Σχήμα 6.6: Αθροιστικές Ανταμοιβές Deep Q Network ανά εποχή.

Όπως φαίνεται ο Deep Q Network, μπορεί να συγκλίνει σταδιακά με πολύ μικρές μεταβολές. Συγκεκριμένα, η χρήση του Target-Network, βοήθησε σημαντικά να εξομαλυνθεί ο τρόπος σύγκλισης του αλγορίθμου. Για την επίτευξη της σύγκλισης χρειάστηκαν 79 εποχές, δηλαδή σχεδόν 160 χρόνια πραγματοποίησης αγοροπωλησιών.

Αντίστοιχα με πριν, είναι σημαντικό να εξετάσουμε τον τρόπο με τον οποίο ο Πράκτορας λαμβάνει τις ανταμοιβές του καθώς και τα κέρδη του.

Reward Function	Google	Acuity
Profit	1521	-1120
Roi	2458	-1952
Net Worth	3442	-2772

Πίνακας 6.2: DQN Κέρδη

Είναι εμφανές, πως η αρχιτεκτονική που επιλέχθηκε για τον DQN, αποδίδει πολύ καλύτερα από αυτή του Q-Network. Με την ομαλή σύγκλιση, παρατηρείται και αντίστοιχη ομαλή λήψη των ανταμοιβών. Παράλληλα, η χρήση του Experience Replay, συντέλεσε σημαντικά στην καλύτερη αντιστάθμιση των αρνητικών σημάτων που λαμβάνει ο Πράκτορας, κάτι που δεν ήταν δυνατό να επιτευχθεί από τον Q-Network. Παρόλα αυτά είναι κι εδώ εμφανής η αρνητική αποκόμιση των κερδών στην μετοχή της Acuity, κάτι που θα εξεταστεί στο κεφάλαιο αξιολόγισης των στρατηγικών.

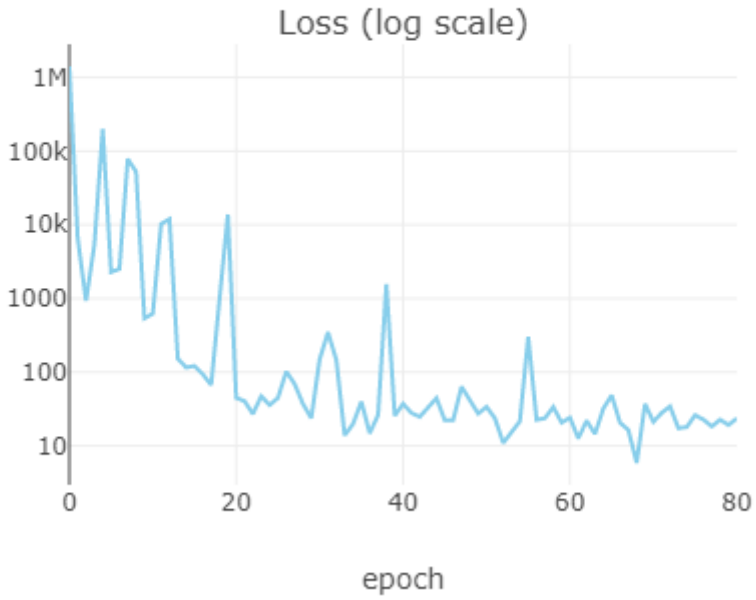
Είναι σημαντικό σε αυτό το σημείο να σημειωθεί, πως η καλύτερη συμπεριφορά που παρατηρείται στον DQN, συνοδεύεται από μία πολύ σημαντική αύξηση στον χρόνο που χρειάζεται για την σύγκλιση του Αλγορίθμου. Συγκεκριμένα, ενώ ένα απλό Q-Network, έφτασε στην σύγκλιση μέσα σε κάποια λεπτά για να ολοκληρώσει την εκπαίδευσή του, για τον DQN, χρειάστηκε αρκετή ώρα, χρόνος σχεδόν δεκαπλάσιος της προηγούμενης αρχιτεκτονικής.

Ο χρόνος αυτός, αν και πολύ μεγαλύτερος από του Q-Network, διατηρήθηκε σε αυτό το επίπεδο, λόγω της καλής ποσοτικοποίησης του χώρου καταστάσεων και δράσεων, δηλαδή τον περιορισμό της

ιστορικότητας στις τιμές, στις προηγούμενες 90 ήμερες καθώς και τον αντίστοιχο περιορισμό των δράσεων σε τρεις. Αν σκεφτεί κανείς ότι ο Πράκτορας προσπαθεί να βρει την βέλτιστη δράση για κάθε δυνατή κατάσταση, μία αύξηση σε έναν από τους δύο αυτούς χώρους , θα μπορούσε να έχει εκθετική αύξηση στο χρόνο εκπαίδευσης.

6.2.3 Double and Dueling DQN

Στη συνέχεια, εξετάζονται οι επιπλέον βελτιστοποιήσεις που έγιναν στην αρχιτεκτονική του DQN. Συγκεκριμένα,



Σχήμα 6.7: Σύγκλιση Double Dueling DQN

Σχήμα 6.8: Αθροιστικές Ανταμοιβές Double Dueling DQN ανά εποχή.

Για τον Double Dueling DQN, μπορούν να εξαχθούν δύο βασικά συμπεράσματα, από τις παραπάνω γραφικές. Συγκεκριμένα, ο αλγόριθμος συγκλίνει πολύ πιο γρήγορα απ'οτι ο DQN, όμως η σύγκλισή του, παρουσιάζει κάποιες απότομες μεταβολές. Αυτές οφείλονται στο γεγονός ότι για τον υπολογισμό του Στόχου, ενώ η δράση υπολογίζεται από το Predicion Network, ο υπολογισμός του Q-Value γίνεται πλέον από το Target Network. Οπως έχει αναφερθεί οι παράμετροι του Target Network δεν ανανεώνονται σε κάθε επανάληψη, σε αντίθεση με αυτές του Prediction Network, που χρησιμοποιείται για τον υπολογισμό της ίδια ποσότητας στον DQN.

Παράλληλα, ο υπολογισμός του Q-Value, σαν άθροισμα των ποσοτήτων $V(s)$ και $A(s,a)$, οδηγεί τον Πράκτορα στην καλύτερη κατανόηση των καταστάσεων στις οποίες βρίσκεται, καθώς επίσης και στην καλύτερη κατανόηση στην επίδραση που έχουν οι δράσεις του. Έτσι, κατανοεί πιο γρήγορα την στρατηγική που χρειάζεται να ακολουθήσει ώστε να αντιμετωπίσει τα αρνητικά σήματα.

Reward Function	Google	Acuity
Profit	2007	922
Roi	2977	-1354
Net Worth	3640	-2598

Πίνακας 6.3: Double and Dueling DQN Κέρδη.

6.2.4 Αξιολόγηση Στρατηγικών

Στη συνέχεια, θα εξετάσουμε τις διαφορετικές στρατηγικές οι διαμορφώνουν οι Πράκτορες, λόγω των διαφορετικών συναρτήσεων επιβράβευσης.

Επιτρέποντας στους συγκεκριμένους Πράκτορες να δράσουν στην δοκιμαστική περίοδο, για τις μετοχές της Google και της Acuity. Η τιμή των δύο αυτών μετοχών έχουν τελείως διαφορετική πορεία. Συγκεκριμένα, της Google εχει ανοδική, ενώ της Acuity έχει καθοδική. Έτσι, θα μπορέσουμε να εξάγουμε ποιοτικά συμπεράσματα, για την συμπεριφορά ενός Πράκτορα, σε διαφορετικές συνθήκες.

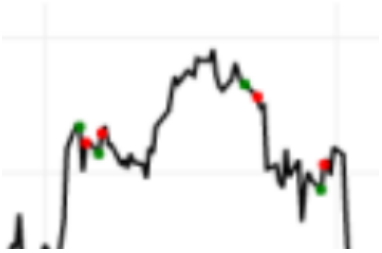
Κέρδος

Αρχικά, θα εξετάσουμε την στρατηγική του που δημιουργήθηκε από την χρήση την συνάρτησης επιβράβευσης με βάση το Κέρδους.

Σχήμα 6.9: Αποτελέσματα συνάρτησης επιβράβευσης Κέρδους στην μετοχή της Google.

Σχήμα 6.10: Αποτελέσματα συνάρτησης επιβράβευσης Κέρδους στην μετοχή της Acuity.

Σχήμα 6.11: Google Profit Closer Look.



Σχήμα 6.12: Acuity Profit Closer Look.

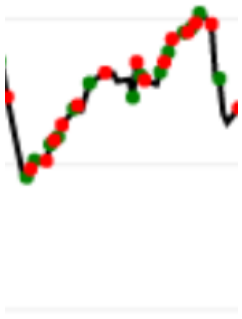
Στη συγκεκριμένη συνάρτηση, είναι εμφανές πως, ο πράκτορας επιθυμεί να πραγματοποιήσει αγοροπωλησίες, όμως προσπαθεί, αυτές να είναι σύντομες, ώστε να είναι σίγουρος για την απόκτηση κέρδους. Προσπαθεί παράλληλα, να αποφύγει τα αρνητικά σήματα κάνοντας εκτεταμένη χρήση της δράσης Hold. Παρόλα αυτά, το γεγονός ότι δεν λαμβάνει σήμα κατά την δράση της πώλησης, δεν του επιτρέπει να επιφέρει υψηλά κέρδη.

RoI

Στην συνέχεια, εξετάζεται η αντίστοιχη στρατηγική που δημιουργήθηκε χρησιμοποιώντας την συνάρτηση επιβράβευσης με βάση το RoI. Συγκεκριμένα,

Σχήμα 6.13: Αποτελέσματα συνάρτησης επιβράβευσης RoI στην μετοχή της Google.

Σχήμα 6.14: Αποτελέσματα συνάρτησης επιβράβευσης RoI στην μετοχή της Acuity.



Σχήμα 6.15: Google RoI, closer look.

Σχήμα 6.16: Acuity RoI, closer look.

Οπως αναφέρθηκε, σκοπός της συγκεκριμένης συνάρτησης, ήταν να δώσει μια εικόνα στον Πράκτορα, για ποιότητα των πωλήσεών του. Με αυτό τον τρόπο, θα μπορούσε να εξάγει και αντίστοιχα συμπεράσματα για τις πωλήσεις του. Όμως, φαίνεται πως τελικά ο Πράκτορας, εκμεταλλεύθηκε με διαφορετικό τρόπο την συγκεκριμένη πληροφορία. Συγκεκριμένα, κατέληξε στο συμπέρασμα, πως οι συνεχείς αγοροπωλησίες, θα του επιφέρουν μεγαλύτερες επιβραβεύσεις. Έτσι, δεν κάνει καθόλου χρήση της δράσης Hold. Το γεγονός αυτό, δεν του επιτρέπει να παράξει κέρδη στην μετοχή η τιμή της οποίας κινείται αρνητικά.

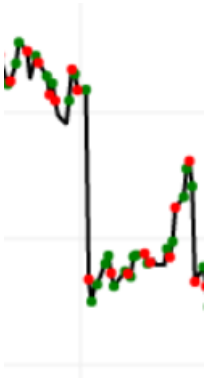
Net Worth Percentage

Τέλος, μπορούμε να παρατηρήσουμε τη συμπεριφορά του Πράκτορα, ο οποίος κάνει χρήση της συνάρτησης επιβράβευσης με βάση το Net Worth,

Σχήμα 6.17: Αποτελέσματα συνάρτησης επιβράβευσης Net Worth στην μετοχή της Google.

Σχήμα 6.18: Αποτελέσματα συνάρτησης επιβράβευσης Net Worth στην μετοχή της Acuity.

Σχήμα 6.19: Google Net Worth, closer look.



Σχήμα 6.20: Acuity Net Worth, closer look.

Στην προκειμένη περίπτωση, έγινε προσπάθεια για ομαλοποίηση της συνάρτησης, ώστε ο Πράκτορας να λαμβάνει σήματα επιβράβευσης για όλες του τις δράσεις. Αυτό οδηγεί τον Πράκτορα, στην επιθυμία να αγοράσει σε διαστήματα που η τιμή της μετοχής ανεβαίνει και να πουλήσει όταν η τιμή έχει καθοδική πορεία. Έτσι, ο Πράκτορας πάλι αγνοεί την δράση του Hold, για χάρη των πιο απότομα μεταβαλλόμενων σημάτων επιβράβευσης των Buy και Sell. Το γεγονός ότι αυτό τον καθιστά

ευάλωτο σε μετοχές όπως αυτές της Acuity, στην οποία οι απότομες μεταβολές ανοδικές μεταβολές τον οδηγούν σε λανθασμένες αγορές με καταστροφικές συνέπειες για τα κέρδη του.

	Google	Google	Google	Acuity	Acuity	Acuity
	Profit	Roi	Net Worth	Profit	Roi	Net Worth
Q-Network	1524	2032	2501	-1465	-1947	-2988
DQN	1521	2458	3442	-1120	-1952	-2772
DDDQN	**2007**	2977	**3640**	**922**	-1354	**-2598**

Πίνακας 6.4: Συγκεντρωτικός Πίνακας Κερδών ανά Πράκτορα-Συνάρτηση Επιβράβευσης-Μετοχής

	Google	Google	Google	Acuity	Acuity	Acuity
	Profit	Roi	Net Worth	Profit	Roi	Net Worth
Q-Network	11.83%	15.78%	19.42%	-11.37%	-15.12%	-23.20%
DQN	12.13%	19.08%	26.73%	-8.69%	-15.15%	-21.52%
DDDQN	**15.58%**	23.11%	**28,26%**	**7.16%**	-10.51%	**-20.17%**

Πίνακας 6.5: Συγκεντρωτικός Πίνακας Ετήσιου Ποσοστιαίου κέρδους

Σύμφωνα, λοιπόν, με τα παραπάνω, μπορούμε να εξάγουμε συμπεράσματα, αρχικά σχετικά με τις συναρτήσεις επιβράβευσης, αλλά και για την εφαρμογή της ενισχυτικής μάθησης στις αλγοριθμικές συναλλαγές γενικότερα.

Κεφάλαιο 7

Συμπεράσματα

Αρχικά από τις δοκιμές που έγιναν και τις συναρτήσεις επιβράβευσης που δοκιμάστηκαν, η συνάρτηση ανταμοιβής που κάνει χρήση του Profit, παρήγαγε τα μικρότερα κέρδη, είναι όμως πιο σταθερή στις απότομες μεταβολές της τιμής της μετοχής. Συγκεκριμένα, η συνάρτηση αυτή δίνει επιβραβεύσεις μόνο για δράσεις των αγορών όταν αυτές παράγουν θετικά ή αρνητικά κέρδη. Έτσι, ο πράκτορας δεν λαμβάνει επιβραβεύσεις για τις δράσεις της Αγοράς και της Στάσης. Φυσικά για να πουλήσει μετοχές και να βγάλει κέρδος πρέπει πρώτα να έχει αγοράσει, κάτι που φαίνεται πως ο Πράκτορας το κατανοεί. Όμως, οι αραιές επιβραβεύσεις που λαμβάνει δεν του επιτρέπουν να πραγματοποιήσει σωστές αγορές ώστε να πετύχει υψηλά κέρδη. Αντίθετα, η συμμετρική φύση της συνάρτησης, δηλαδή, είτε 1 ή -1 σε περίπτωση κέρδους η ζημίας, οδηγούν τον Πράκτορα στην εκμετάλλευση της δράσης Hold, ώστε να αποφύγει αρνητικά σήματα. Αυτό, του επιτρέπει να ανταπεξέρχεται στην καθοδική τιμή και στις απότομες μεταβολές της μετοχής της Acuity.

Η συνάρτηση ανταμοιβής RoI, παράγει καλύτερα κέρδη από προηγούμενη, όμως, δεν γίνεται πλέον η δράση του Hold. Αντίθετα, προτιμάει να αγοράσει ώστε να πουλήσει κατευθείαν και να βγάλει όσο το δυνατόν περισσότερο κέρδος γίνεται. Παρόλο που έγινε προσπάθεια, ενημέρωσης του Πράκτορα για το πόσο καλές είναι οι πωλήσεις του, φαίνεται ότι ο Πράκτορας, τελικά κατέληξε στο συμπέρασμα, πώς με τη γρήγορη αγοραπωλησία, θα παράξει μεγαλύτερα κέρδη. Όμως, η μη χρήση της δράσης Hold, τον καθιστά αδύνατο να καταφέρει να να αντιδράσει σωστά στις απότομες μεταβολές της μετοχής της Acuity. Οι θετικές απότομες μεταβολές της τιμής της μετοχής ξεγελούν τον Πράκτορα σε αγορές, όμως επειδή η μετέπειτα πορεία της μετοχής είναι καθοδική, η πώληση που θα πραγματοποιήσει ο Πράκτορας, θα είναι ζημιογόνα.

Τέλος, είναι εμφανές ότι η συνάρτηση που κάνει χρήση του Net Worth, παράγει τα πιο κερδοφόρα αποτελέσματα όταν η τιμή της μετοχής έχει ανοδική πορεία. Στη συγκεκριμένη συνάρτηση, έγινε προσπάθεια ομαλοποίησης των ανταμοιβών που λαμβάνει ο Πράκτορας, ώστε να λαμβάνει σήματα για όλες του τις ενέργειες. Όμως, το σήμα της συνάρτησης αυτής είναι άμεσα συνδεδεμένο με την πορεία της τιμής της μετοχής. Έτσι, εάν ο Πράκτορας, έχει αγοράσει μετοχές που η τιμή τους ανεβαίνει, όπως συμβαίνει στη μετοχή της Google, λαμβάνει θετικά σήμα και επιθυμεί να αγοράσει. Φυσικά, σε διαστήματα που η τιμή της μετοχής κατεβαίνει, επιθυμεί να πουλήσει, ώστε να βγάλει κέρδος και να μην επωμιστεί ζημία. Όμως, αντίστοιχα με την συνάρτηση που κάνει χρήση του RoI, οι απότομες μεταβολές της μετοχής της Acuity, οδηγεί τον Πράκτορα σε λανθασμένες αγορές και παρόλο που μετά από κάθε λανθασμένη αγορά προσπαθεί να ανακάμψει, τελικά δεν τα καταφέρνει.

Είναι εμφανές, λοιπόν, πόσο δύσκολη είναι η δημιουργία μίας συνάρτησης ανταμοιβής, η οποία να οδηγεί τον Πράκτορα στο επιθυμητό αποτέλεσμα. Η συνάρτηση ανταμοιβής, αποτελεί τη μοναδική πηγή πληροφορίας (feedback) που λαμβάνει ο Πράκτορας για τις δράσεις του και το πόσο καλές ήταν. Όσο περισσότερη πληροφορία λαμβάνει ο Πράκτορας, τόσο καλύτερα θα καταφέρει να προσαρμόσει τις δράσεις του. Είναι, λοιπόν, απαραίτητο η συνάρτηση επιβράβευσης να μην παράγει αραιά σήματα, αλλά να είναι και ομαλή. Παρόλα αυτά, η δημιουργία μίας τέτοιας συνάρτησης είναι από τα δυσκολότερα προβλήματα που αντιμετωπίζονται συχνά στην ενισχυτική μάθηση.

Επιπλέον, αξίζει να σημειωθεί πως η Ενισχυτική Μάθηση μπορεί να εφαρμοστεί στον τομέα των συναλλαγών στο Χρηματιστήριο, όμως η εφαρμογή της παρουσιάζει αρκετές προκλήσεις. Πέραν της δημιουργίας κατάλληλης συνάρτησης επιβράβευσης, το γεγονός ότι η αλλαγή της τιμής των μετοχών δεν είναι άμεσα συνδεδεμένη με τις δράσεις του Πράκτορα, καθιστά ιδιαίτερα δύσκολη την κατανό-

ηση των νέων καταστάσεων στις οποίες μεταβαίνει. Έτσι, η Χρηματιστηριακή αγορά, αναπαρίσταται σαν πρόβλημα Μερικώς Παρατηρήσιμης Διαδικασίας απόφασης Markov, μία ανοικτή περιοχή έρευνας.

Ένα ακόμη πολύ σημαντικό συμπέρασμα, που προέκυψε από την παρούσα διπλωματική, είναι ο χρόνος ο οποίος απαιτείται για την εκπαίδευση του Πράκτορα. Συγκεκριμένα, παρόλο που έγινε προσπάθεια σωστής ποσοτικοποίησης των καταστάσεων και των δράσεων του περιβάλλοντος, απαιτήθηκε αρκετή ώρα για την εκπαίδευση ενός Πράκτορα. Αυτό συνεπάγεται τη δύσκολη κλιμάκωση σε αντίστοιχο πρόβλημα του οποίου ο χώρος καταστάσεων ή ο χώρος δράσεων θα είναι πολύ μεγαλύτερος

Τέλος, αξίζει να σημειωθεί πως η χρήση Νευρωνικών Δικτύων στους Αλγορίθμους Ενισχυτικής μάθησης, καθιστά εφικτή την εφαρμογή τους σε προβλήματα που συναντώνται στον πραγματικό κόσμο, όπως οι Συναλλαγές στο Χρηματιστήριο. Παρότι επιλύουν σημαντικά θέματα της κλασσικής ενισχυτικής μάθησης, η χρήση τους συνεπάγεται την ύπαρξη νέων προβλημάτων τα οποία επιλύονται σταδιακά όσο προχωράει η έρευνα στο τομέα.

Κεφάλαιο 8

Μελλοντική Εργασία

Για την επέκταση της παρούσας εργασίας θα μπορούσαν να ακολουθηθούν πολλές κατευθύνσεις. Αρχικά, θα μπορούσε να μεγαλώσει ο χώρος καταστάσεων, ώστε να περιλαμβάνει περισσότερες πληροφορίες. Για παράδειγμα, θα μπορούσε να περιλαμβάνει είτε οικονομικούς δείκτες που ένας επενδυτής θα θεωρούσε χρήσιμους, είτε πληροφορία από άλλα συστήματα, όπως συστήματα μηχανικής μάθησης που προβλέπουν την κίνηση της τιμής μίας μετοχής χρησιμοποιώντας οικονομικά άρθρα ή ανακοινώσεις των Εταιριών.

Θα μπορούσε επίσης, να γίνει διαφορετική ποσοτικοποίηση του χώρου καταστάσεων και δράσεων ώστε ο Πράκτορας να πραγματοποιεί συναλλαγές διαστήματα ωρών, λεπτών ή και δευτερολέπτων. Παρόλα αυτά, κάτι τέτοιο θα απαιτούσε τη χρήση διαφορετικών αλγορίθμων και αρχιτεκτονικών, ώστε να αντιμετωπιστούν οι πιθανόν αραιώτερες επιβραβεύσεις που θα λαμβάνει ο Πράκτορας σε τέτοιες περιπτώσεις.

Όπως αναφέρθηκε ένα από τα σημαντικότερα κομμάτια στα προβλήματα Ενισχυτικής Μάθησης, αποτελεί η συνάρτηση επιβράβευσης. Έτσι, θα ήταν χρήσιμο, να μελετηθούν κι άλλες συναρτήσεις επιβράβευσης, στις οποίες θα γίνει προσπάθεια περαιτέρω ομαλοποίησης του τρόπου με τον οποίο ο Πράκτορας λαμβάνει τις επιβραβεύσεις σε κάθε του δράση. Φυσικά, το σημείο αυτό χρήζει ιδιαίτερης προσοχής, αφού όσο πιο πολύπλοκη γίνει μία συνάρτηση επιβράβευσης, τόσο δυσκολότερο μπορεί να είναι για τον Πράκτορα η κατανόηση του τρόπου με τον οποίο πρέπει να δράσει ώστε να πετύχει τον επιθυμητό στόχο.

Τέλος, οι Αλγοριθμικές συναλλαγές, αναπαραστήθηκαν ως ένα πρόβλημα Μερικώς Προσβάσιμης Διαδικασίας απόφασης Markov, αφού η διαμόρφωση της τιμής της μετοχής είναι αποτέλεσμα αλληλεπίδρασης μεγάλου αριθμού επενδυτών. Έτσι, παρόλο που εδώ αυτή η αλληλεπίδραση θεωρήθηκε σαν μέρος του περιβάλλοντος, το πρόβλημα θα μπορούσε να αναπαρασταθεί ως ένα πρόβλημα πολλών Πρακτόρων. Με αυτό τον τρόπο, ο κάθε Πράκτορας δεν θα παρατηρεί απλά τις αλλαγές στις τιμές των μετοχών, αλλά θα προσπαθούσε παράλληλα, να κατανοήσει και να εκμεταλλευτεί την στρατηγική των αντιπάλων του.

Βιβλιογραφία

[Arul17] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage and Anil Anthony Bharath, "Deep Reinforcement Learning: A Brief Survey", *IEEE Signal Processing Magazine*, vol. 34, no. 6, p. 26–38, Nov 2017.

[Broc16] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang and Wojciech Zaremba, "OpenAI Gym", *arXiv:1606.01540 [cs]*, June 2016. arXiv: 1606.01540.

[Chen19] James Chen, "Algorithmic Trading Definition", *Investopedia*, 2019.

[Fran18] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare and Joelle Pineau, "An Introduction to Deep Reinforcement Learning", *Foundations and Trends® in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.

[Hass10] Hado V. Hasselt, "Double Q-learning", in J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pp. 2613–2621, Curran Associates, Inc., 2010.

[Hass16] Hado van Hasselt, Arthur Guez and David Silver, "Deep Reinforcement Learning with Double Q-Learning", in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 2094–2100, AAAI Press, 2016. event-place: Phoenix, Arizona.

[Hayk09] Simon O. Haykin, *Neural Networks and Learning Machines, Third Edition*, Pearson Education, McMaster University, Canada, 2009.

[Huan18] Chien Yi Huang, "Financial Trading as a Game: A Deep Reinforcement Learning Approach", *arXiv:1807.02787 [cs, q-fin, stat]*, July 2018. arXiv: 1807.02787.

[Kans18] Satwik Kansal, *Hands-on Reinforcement Learning with TensorFlow*, Pakt, 2018.

[Li18] Yuxi Li, "Deep Reinforcement Learning", *arXiv:1810.06339 [cs, stat]*, October 2018. arXiv: 1810.06339.

[Mnih13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra and Martin Riedmiller, "Playing Atari with Deep Reinforcement Learning", *arXiv:1312.5602 [cs]*, December 2013. arXiv: 1312.5602.

[Mood99] John Moody and Matthew Saffell, "Reinforcement Learning for Trading", in *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II*, pp. 917–923, Cambridge, MA, USA, 1999, MIT Press.

[Necc16] Pierpaolo G. Necchi, "Reinforcement Learning For Automated Trading", 2016.

[Shar19] Rakesh Sharma, "Quantitative Trading Definition", *Investopedia*, 2019.

[Wang15] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot and Nando de Freitas, "Dueling Network Architectures for Deep Reinforcement Learning", *arXiv:1511.06581 [cs]*, November 2015. arXiv: 1511.06581.

[Wier12] Marco Wiering and Martijn van Otterlo, editors, *Reinforcement learning: state-of-the-art*, no. v.12 in Adaptation, learning, and optimization, Springer, Heidelberg ; New York, 2012. OCLC: ocn768170254.

[Wool95] Michael Wooldridge and Nicholas R. Jennings, "Intelligent agents: theory and practice", *The Knowledge Engineering Review*, vol. 10, no. 2, p. 115–152, 1995.

[IB06] Ν. Βασιλειάδης Φ. Κόκκορας Η. Σακελλαρίου Ι. Βλαχάβας, Π. Κεφαλάς, *Τεχνητή Νοημοσύνη*, Εκδόσεις Πανεπιστημίου Μακεδονίας, 2006.

Παράρτημα Α

Ευρετήριο Συμβολισμών

$V(s)$: Συνάρτηση αξίας στην κατάσταση s

$V_k(s)$: Συνάρτηση αξίας στην κατάσταση s, στην επανάληψη k

$V^*(s)$: Βέλτιστη συνάρτηση αξίας στην κατάσταση s

$\pi(s)$: Δράση που επιλέγεται στην κατάσταση s

$V^\pi(s)$: Βέλτιστη συνάρτηση αξίας στην κατάσταση s χρησιμοποιώντας την πολιτική π

$P(s'|s,a)$: Συνάρτηση μετάβασης (πιθανότητας) από την κατάσταση s, στην κατάσταση s', χρησιμοποιώντας την δράση a

$R(s,a,s')$: Συνάρτηση σήματος ενίσχυσης, για την μετάβαση στην κατάσταση s', από την κατάσταση s, επιλέγοντας την δράση a.

$\sum_{j=1}^{q} x_i$: Άθροισμα της μορφής $x_1 + x_2 + ... + x_q$

$\phi(S)$: Συνάρτηση ενεργοποίησης, με είσοδο το σήμα S

$\frac{\partial(f)}{\partial x}$: Μερική Παράγωγος της συνάρτησης f, ως προς την μεταβλητή x

$\phi\,'(S)$: Παράγογος της συνάρτησης ϕ

$\nabla F(x,y,z)$: Παράγωγος συνάρτησης F, ως προς τις μεταβλητές x, y, z

$\max_{a'} Q(s,a')$: Μεγίστη ποσότητα συνάρτησης Q, ως προς τη μεταβλητή a

$E_{s'\sim P(s'|s,a)}[\sum F(s,a,s')]$: Αναμενόμενη τιμή αθροίσματος, της συνάρτησης F, λόγω της πιθανής μετάβασης στην κατάσταση s', από την κατάσταση s, με την δράση a

Παράρτημα Β

Ευρετήριο Τεχνολογιών

Python: Η γλώσσα στην οποία υλοποιήθηκε η εργασία.

Tensorflow: Εργαλείο για την υλοποίηση των Νευρωνικών Δικτύων

Numpy: Βιβλιοθήκη επιστήμων υπολογισμών της Python.

OpenAI Gym: Διεπαφή Περιβάλλοντος Ενισχυτικής Μάθησης από την OpenAI.

Plotly: Εργαλείο γραφικών παραστάσεων για την Python.

Κώδικας Εργασίας https://github.com/SkourasKonst/RL_Thesis