ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Διπλωματική Εργασία

Δημιουργία αλγορίθμου για αυτόματη αγοραπωλησία μετοχών μέσα στην ημέρα για το δείκτη S&P 500

Λέανδρος Καΐπης

Επιβλέπων καθηγητής: Δρ. Ευστράτιος Γεωργόπουλος

Πάτρα, Ιούλιος 2025

Δημιουργία αλγορίθμου για αυτόματη αγοραπωλησία μετοχών μέσα στην ημέρα για το δείκτη S&P 500

Λέανδρος Καΐπης

Επιτροπή Επίβλεψης Πτυχιακής / Διπλωματικής Εργασίας

Επιβλέπων Καθηγητής:

Δρ. Ευστράτιος Γεωργόπουλος

Καθηγητής

Τμήμα Διοίκησης Επιχειρήσεων

Επιστήμης & Οργανισμών

Πανεπιστημίου Πελοποννήσου

Συνεργαζόμενο Εκπαιδευτικό Προσωπικό (ΣΕΠ)

Ελληνικό Ανοιχτό Πανεπιστήμιο (ΕΑΠ)

Συν-Επιβλέπων Καθηγητής:

Δρ. Γρηγόριος Ν. Μπεληγιάννης

Καθηγητής

Τμήμα Επιστήμης & Τεχνολογίας Τροφίμων

Πανεπιστήμιο Πατρών

Συν-Επιβλέπων Καθηγητής:

Δρ. Χαϊρή Κιούρτ

Ερευνητής

Ινστιτούτο Επεξεργασίας Λόγου

Ερευνητικό Κέντρο Αθηνά

Πάτρα, Ιούλιος 2025

Στους γονείς μου

Απόστολο και Σοφία Καΐπη

Στην αγαπημένη μου σύζυγο

Σταυρούλα Ζησοπούλου

και την κόρη μας Ηλέκτρα.

Ευχαριστίες

Ευχαριστώ τον καθηγητή Ευστράτιο Γεωργόπουλο για την εμπιστοσύνη και την ανάθεση ενός τόσο ενδιαφέροντος θέματος. Οι συμβουλές, η καθοδήγηση και η άψογη συνεργασία μας συνέβαλαν καθοριστικά για την περάτωση της παρούσας διπλωματικής εργασίας. Θα ήθελα επίσης να ευχαριστήσω τα μέλη της τριμελούς επιτροπής, τον καθηγητή Γρηγόριο Ν. Μπεληγιάννη και τον Δρ. Χαϊρή Κιούρτ για τον χρόνο που αφιέρωσαν να μελετήσουν την εργασία καθώς και για τις εύστοχες παρατηρήσεις τους.

Τις ευχαριστίες μου επίσης οφείλω στον υποψήφιο διδάκτορα Θωμά Αμοργιανιώτη για την εξαιρετική συνεργασία και τις χρήσιμες συμβουλές του πάνω στο γνωστικό αντικείμενο της εργασίας.

Θα ήθελα να εκφράσω την ευγνωμοσύνη μου στους γονείς μου Απόστολο και Σοφία Καΐπη. Τέλος, ευχαριστώ θερμά την σύζυγό μου Σταυρούλα Ζησοπούλου για την υπομονή και υποστήριξη της σε αυτό το ταξίδι γνώσης και κυρίως το σημαντικότερο δημιούργημά μας, την κόρη μας Ηλέκτρα.

Περίληψη

Η τεχνητή νοημοσύνη αναδεικνύεται ως καταλύτης καινοτομίας σε πολλές βιομηχανίες και ιδιαίτερα στον τομέα των χρηματοοικονομικών. Η χρήση αλγορίθμων τεχνητής νοημοσύνης, όπως η ενισχυτική μάθηση, έχει μεταμορφώσει τον τρόπο που οι εταιρείες επενδύσεων αναλύουν και αξιοποιούν τα δεδομένα της αγοράς προκειμένου να βελτιώσουν τις στρατηγικές τους και να μεγιστοποιήσουν τις αποδόσεις τους. Με τη βοήθεια αυτών των τεχνολογιών, οι επενδυτές μπορούν να ελαχιστοποιήσουν τις επιπτώσεις του ανθρώπινου παράγοντα και να αποφεύγουν συναισθηματικές επιρροές που επηρεάζουν την κρίση τους.

Η ενισχυτική μάθηση αποτελεί ένα πολύτιμο εργαλείο στον τομέα των χρηματοοικονομικών, επιτρέποντας την ανάπτυξη ευφυών πρακτόρων που βελτιώνουν τη διαδικασία συναλλαγών και επενδύσεων. Η παρούσα εργασία διερευνά την εφαρμογή της ενισχυτικής μάθησης σε χρηματοοικονομικούς τομείς, όπως η αγοραπωλησία μετοχών. Στο πλαίσιο αυτό, αναπτύσσονται πράκτορες που βασίζονται στο Dueling Double Deep Q-Network (DDDQN) και αξιοποιούν ειδικά διαμορφωμένες συναρτήσεις επιβράβευσης, επιτρέποντας την ανάπτυξη προσαρμοσμένων τακτικών που ενισχύουν τη συνολική απόδοση και ανθεκτικότητα του επενδυτικού τους χαρτοφυλακίου.

Η μελέτη περιλαμβάνει τη σχεδίαση ενός περιβάλλοντος προσομοίωσης που αναπαριστά την αγορά ενός χρηματιστηρίου, στο οποίο οι πράκτορες καλούνται να αξιολογούν τις στρατηγικές τους με βάση την κερδοφορία, τη σταθερότητα και την αποτελεσματικότητα των αποφάσεών τους. Ιδιαίτερη έμφαση δίνεται στην συνάρτηση επιβράβευσης, η οποία καθοδηγεί τη διαδικασία εκμάθησης της στρατηγική που θα ακολουθούν οι πράκτορες κατά τη διάρκεια της εκπαίδευσής τους ώστε να μπορούν να εκτιμούν και αν χρειαστεί να αναπροσαρμόζουν τις τακτικές τους για να επιτύχουν καλύτερα αποτελέσματα.

Μέσω της εφαρμογής αυτών των τεχνικών, η εργασία αναδεικνύει τη σημασία της ενισχυτικής μάθησης στην αποδοτική διαχείριση και βελτιστοποίηση επενδυτικών στρατηγικών, προσφέροντας πρακτικά οφέλη για τη βελτίωση των αποτελεσμάτων στις συναλλαγές.

Λέξεις – Κλειδιά

Τεχνητή Νοημοσύνη, Μηχανική Μάθηση, Ενισχυτική Μάθηση, Ευφυείς Πράκτορες, Q-learning, Dueling Double Deep Q-Network

Abstract

Artificial Intelligence is emerging as a catalyst for innovation in many industries, especially in the financial sector. The use of Artificial Intelligence algorithms, such as reinforcement learning, has transformed the way investment firms analyze and leverage market data to improve their strategies and maximize their returns. With the help of these technologies, investors can minimize the impact of human factors and avoid emotional influences that affect their judgment.

Reinforcement learning is a valuable tool in finance, enabling the development of intelligent agents that improve the trading and investment process. This paper explores the application of reinforcement learning in financial domains such as stock trading. In this context, agents based on the Dueling Double Deep Q-Network (DDDQN) are developed that exploit customized reward functions, enabling the development of customized tactics that enhance the overall performance and resilience of their investment portfolio.

The study involves the design of a simulation environment representing a stock market in which agents are asked to evaluate their strategies based on the profitability, stability and efficiency of their decisions. Particular emphasis is placed on reward functions, which guide the learning process of the strategy that agents will follow during their training so that they can evaluate and, if necessary, adjust their tactics to achieve better results.

Through the application of these techniques, the paper highlights the importance of reinforcement learning in the efficient management and optimization of investment strategies, offering practical benefits for improving trading results.

Keywords

Artificial Intelligence, Machine Learning, Reinforcement Learning, Intelligent Agents, Q-learning, Dueling Double Deep Q-Network

Περιεχόμενα

Κατάλογος Εικόνων / Σχημάτων

Κατάλογος Πινάκων

Ψευδοκώδικας

Κεφάλαιο 1

1.1 Εισαγωγή

Στην εποχή που ζούμε η τεχνητή νοημοσύνη (Artificial Intelligence, AI) και η μηχανική μάθηση (Machine learning, ML), παίζουν καθοριστικό ρόλο στις σύγχρονες τεχνολογίες και έχουν γίνει αναπόσπαστο κομμάτι της καθημερινότητάς μας. Από προτάσεις προϊόντων σε ηλεκτρονικά καταστήματα και τα αυτόνομα οχήματα, μέχρι τα ρομπότ και τις διαγνωστικές ιατρικές εφαρμογές, οι τεχνολογίες αυτές διαμορφώνουν το μέλλον μας με τρόπους που ήταν αδιανόητοι πριν μερικά χρόνια. Η εισαγωγή αυτών των τεχνολογιών έχει βελτιώσει την αποδοτικότητα και την καινοτομία σε διάφορους τομείς όπως η υγεία, η εκπαίδευση, η βιομηχανία και τα χρηματοοικονομικά.

Ειδικά στον κλάδο των χρηματοοικονομικών, η δυναμική φύση του περιβάλλοντος απαιτεί συνεχή παρακολούθηση, εξαιτίας πιθανών διακυμάνσεων της αγοράς. Κάτι τέτοιο είναι σχεδόν αδύνατον, αν όχι ανέφικτο, να επιτευχθεί χωρίς την χρήση αλγορίθμων.

Στην παρούσα διπλωματική περιγράφεται η δημιουργία αλγορίθμου για την αυτόματη αγοραπωλησία μετοχών μέσα στην ημέρα για τον δείκτη S&P 500. Η αυτόματη αγοραπωλησία μετοχών, γνωστή ως αλγοριθμικό trading (algorithmic trading) έχει εξελιχθεί σε βασικό εργαλείο για επενδυτές, hedge funds και χρηματοοικονομικά ιδρύματα που στοχεύουν στη μεγιστοποίηση των αποδόσεων και την ελαχιστοποίηση των κινδύνων.

Στο πρώτο μέρος της παρούσας εργασίας γίνεται ανάπτυξη των βασικών εννοιών και αλγορίθμων σε θεωρητικό επίπεδο εστιάζοντας στην ενισχυτική μάθηση (Reinforcement Learning), ένα κλάδο της μηχανικής μάθησης, όπου ο πράκτορας μαθαίνει μέσω αλληλεπίδρασης με το περιβάλλον ώστε να μεγιστοποιήσει το τελικό κέρδος του.

Στο δεύτερο μέρος παρουσιάζεται αναλυτικά ο σχεδιασμός του αλγορίθμου, όπως επίσης τα εργαλεία και οι βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίησή του.

Στο τρίτο μέρος γίνεται μετάβαση από την σχεδίαση στην υλοποίηση του αλγορίθμου για την αυτόματη αγοραπωλησία μετοχών μέσα στην ημέρα για τον δείκτη S&P 500.

Τέλος στο τέταρτο μέρος παρατίθενται τα αποτελέσματα και συμπεράσματα της παρούσας διπλωματικής εργασίας.

1.2 Περιγραφή της διπλωματικής και ορισμός του προβλήματος

1.2.1 Περιγραφή και στόχος

Το πρόβλημα της αυτόματης αγοραπωλησίας μετοχών μέσα στην ημέρα, γνωστό και ως intraday trading, αναφέρεται στην αυτόματη εκτέλεση συναλλαγών (αγορών και πωλήσεων) μετοχών σε σύντομα χρονικά διαστήματα εντός μιας ημέρας. Η ημερήσια διαπραγμάτευση απαιτεί ακρίβεια, καθώς αξιοποιεί μικρές διακυμάνσεις στην τιμή των μετοχών, επιδιώκοντας τη μεγιστοποίηση των κερδών χωρίς να διατηρούνται οι θέσεις στο τέλος της ημέρας. Παραδοσιακά, η πρόβλεψη τιμών μετοχών βασίζεται σε στατιστικά μοντέλα και τεχνικές ανάλυσης, όπου η ανάλυση ιστορικών τιμών και δεικτών, όπως ο μέσος όρος ή η απόκλιση, αποτελεί τη βάση για την εκτίμηση μελλοντικών κινήσεων (Moody et al., 1998). Παρόλο που αυτά τα μοντέλα είναι αποτελεσματικά, παρουσιάζουν περιορισμούς καθώς δεν μπορούν να προσαρμόζονται άμεσα στις ταχέως μεταβαλλόμενες αγορές. Η ενισχυτική μάθηση προσφέρει μια πιο δυναμική λύση, δίνοντας τη δυνατότητα στο μοντέλο να μαθαίνει από την εμπειρία και να αναπροσαρμόζει τις αποφάσεις του συνεχώς σε πραγματικό χρόνο (Deng et al.,2017).

Στόχος της διπλωματικής είναι η δημιουργία μοντέλου για βραχυπρόθεσμη πρόβλεψη και αυτοματοποίηση της διαδικασίας αγοράς και πώλησης μετοχών. Το μοντέλο θα βασίζεται στην ενισχυτική μάθηση και συγκεκριμένα στον αλγόριθμο Dueling Double Deep Q-Network. Με αυτό τον αλγόριθμο, το μοντέλο θα αξιολογεί την κατάσταση της αγοράς και θα εκτιμά την αναμενόμενη αξία κάθε πιθανής ενέργειας (αγορά, πώληση ή αναμονή) ώστε να προβλέπει τις «κινήσεις» των μετοχών. Επίσης θα του επιτρέψει να βελτιώνει διαρκώς τη στρατηγική του, μαθαίνοντας από τις συναλλαγές και θα προσαρμόζεται στις διακυμάνσεις των τιμών (Liang et al, 2018). Η ενισχυτική μάθηση επιτρέπει στο μοντέλο να μαθαίνει από τις εμπειρίες του και να βελτιώνει τις αποφάσεις του με την πάροδο του χρόνου. Το μοντέλο αυτόματης αγοραπωλησίας μετοχών που θα αναπτύξουμε, θα εφαρμοστεί σε δεδομένα του δείκτη S&P 500.

Η βραχυπρόθεσμη πρόβλεψη μετοχών είναι κρίσιμη για την επίτευξη κερδοφορίας στις χρηματιστηριακές συναλλαγές. Σε ένα δυναμικό χρηματοοικονομικό περιβάλλον, η

αυτοματοποίηση της διαδικασίας αγοράς και πώλησης μετοχών μπορεί να βελτιώσει την αποδοτικότητα και να μειώσει τον κίνδυνο απώλειας.

1.2.2 Ορισμός του προβλήματος της αυτόματης αγοραπωλησίας μετοχών μέσα στην ημέρα

Ορίζουμε ένα αρχικό κεφάλαιο ώστε να διαχειριστεί ο πράκτορας (agent). Ανά χρονικές στιγμές μέσα στην ημέρα θα γίνεται επιλογή από τον πράκτορα που αλληλοεπιδρά με ένα δυναμικό περιβάλλον (που περιλαμβάνει διάφορους παράγοντες της αγοράς, όπως τις τιμές των μετοχών) για αγορά ή πώληση μετοχών με στόχο την μεγιστοποίηση του κέρδους. Έχει οριστεί κατώτατο όριο απώλειας κεφαλαίου, που αν ξεπεραστεί, ο αλγόριθμος θα τερματίζει. Επίσης όπως είναι λογικό, ο αλγόριθμος δεν μπορεί να τρέχει επ' άπειρον όποτε έχει οριστεί συγκεκριμένη διάρκεια εκτέλεσης.

1.3 Χρηματιστήριο

Ως Χρηματιστήριο εννοούμε την οργανωμένη αγορά, όπου επενδυτές και μεσίτες διενεργούν αγοραπωλησίες μετοχών και άλλων χρηματοοικονομικών προϊόντων, οι τιμές των οποίων ορίζονται από την προσφορά και ζήτηση.

Οι επενδυτές θα πρέπει να γνωρίζουν διάφορους χρηματιστηριακούς όρους κάποιοι από τους οποίους είναι οι παρακάτω:

- Αγορά (Buy): Η διαδικασία αγοράς μετοχών ή άλλων χρηματοοικονομικών προϊόντων.
- Πώληση (Sell): Η διαδικασία πώλησης μετοχών ή άλλων χρηματοοικονομικών προϊόντων.
- Δανεισμός (Margin Trading): Η πρακτική δανεισμού χρημάτων από χρηματιστηριακή εταιρεία για την αγορά μετοχών, χρησιμοποιώντας τις υπάρχουσες μετοχές ως εγγύηση.
- Εντολή Αγοράς (Buy Order): Εντολή προς τον χρηματιστή να αγοράσει μετοχές σε συγκεκριμένη τιμή ή χαμηλότερη.

- Εντολή Πώλησης (Sell Order): Εντολή προς τον χρηματιστή να πουλήσει μετοχές σε συγκεκριμένη τιμή ή υψηλότερη.
- Εντολή Stop-Loss: Εντολή πώλησης μετοχών όταν η τιμή πέσει κάτω από ένα συγκεκριμένο επίπεδο, για να περιοριστούν οι απώλειες.
- Εντολή Limit: Εντολή αγοράς ή πώλησης μετοχών σε συγκεκριμένη τιμή ή καλύτερη.
- Χαρτοφυλάκιο (Portfolio): Η συλλογή όλων των επενδύσεων που κατέχει ένας επενδυτής.
- Μερίσματα (Dividends): Τα κέρδη που διανέμονται στους μετόχους μιας εταιρείας.
- Απόδοση (Yield): Το ποσοστό απόδοσης μιας επένδυσης σε σχέση με το κόστος της.
- Long Position (Μακροχρόνια Θέση): Όταν ένας επενδυτής αγοράζει μετοχές με την προσδοκία ότι η τιμή τους θα αυξηθεί στο μέλλον. Ο επενδυτής κερδίζει από την αύξηση της τιμής των μετοχών.
- Short Position (Βραχυχρόνια Θέση): Όταν ένας επενδυτής δανείζεται μετοχές και τις πουλάει με την προσδοκία ότι η τιμή τους θα πέσει. Ο επενδυτής κερδίζει από την πτώση της τιμής των μετοχών, καθώς μπορεί να τις αγοράσει πίσω σε χαμηλότερη τιμή και να τις επιστρέψει.

1.3.1 Μετοχή

Στις χρηματοπιστωτικές αγορές, μια μετοχή είναι μια μονάδα ίσης ιδιοκτησίας μεριδίων στο μετοχικό κεφάλαιο μιας εταιρείας. Το μετοχικό κεφάλαιο αναφέρεται στο σύνολο των μετοχών μιας εταιρείας. Ο κάτοχος μετοχών μιας εταιρείας, ονομάζεται μέτοχος της εταιρείας.

1.3.2 Όγκος συναλλαγών (Trading volume)

Ο όγκος συναλλαγών είναι ο συνολικός αριθμός ενός περιουσιακού στοιχείου που διαπραγματεύτηκε (αγορά ή πώληση) κατά τη διάρκεια μιας δεδομένης χρονικής περιόδου. Αποτελεί μια σημαντική μέτρηση που πρέπει να λαμβάνεται υπόψη κατά τις συναλλαγές,

καθώς παρέχει πολύτιμες πληροφορίες για τη ρευστότητα και το ενδιαφέρον των επενδυτών για ένα περιουσιακό στοιχείο.

Ο όγκος αναφέρεται στον αριθμό των μονάδων μιας συγκεκριμένης μετοχής που αγοράζονται και πωλούνται εντός ενός συγκεκριμένου χρονικού πλαισίου, όπως μέσα σε μία ημέρα ή εβδομάδα.

Υψηλός όγκος συναλλαγών υποδηλώνει ότι ένα περιουσιακό στοιχείο διακινείται ενεργά, κάτι που δείχνει μεγάλο ενδιαφέρον από τους επενδυτές και αυξημένη ρευστότητα. Αυτό σημαίνει ότι είναι εύκολο να αγοραστούν ή να πωληθούν μετοχές χωρίς να προκαλούνται μεγάλες διακυμάνσεις στην τιμή.

Χαμηλός όγκος συναλλαγών μπορεί να υποδεικνύει έλλειψη ενδιαφέροντος ή ρευστότητας, γεγονός που δυσκολεύει τους επενδυτές να αγοράσουν ή να πουλήσουν μετοχές στην τιμή που επιθυμούν. Σε τέτοιες περιπτώσεις, ακόμα και μικρές συναλλαγές μπορεί να οδηγήσουν σε μεγαλύτερες διακυμάνσεις στην τιμή.

1.3.3 Βιβλίο εντολών (Order book)

Το βιβλίο εντολών είναι το μητρώο όλων των εντολών αγοράς και πώλησης που καταχωρούνται από τους εγκεκριμένους χρήστες του ηλεκτρονικού συστήματος διαπραγμάτευσης ενός χρηματιστηρίου.

- Προσφορά (Sell orders): Περιλαμβάνει τις εντολές πώλησης που έχουν καταχωριστεί από τους επενδυτές, δείχνοντας τις τιμές στις οποίες οι πωλητές είναι διατεθειμένοι να πουλήσουν ένα συγκεκριμένο περιουσιακό στοιχείο.
- Ζήτηση (Buy orders): Περιλαμβάνει τις εντολές αγοράς που δείχνουν τις τιμές στις οποίες οι αγοραστές είναι διατεθειμένοι να αγοράσουν το περιουσιακό στοιχείο.

Το βιβλίο εντολών ταξινομεί αυτές τις εντολές βάση την τιμή και την ώρα καταχώρισης για κάθε κινητή αξία, προσφέροντας μια σαφή εικόνα της τρέχουσας προσφοράς και ζήτησης. Μέσω αυτού, οι εντολές αντιστοιχίζονται, εξασφαλίζοντας την ομαλή λειτουργία των συναλλαγών στην αγορά.

1.3.4 Τιμές στο βιβλίο εντολών

Τα στοιχεία που συνήθως καταχωρούνται στο βιβλίο εντολών περιλαμβάνουν:

- Άνοιγμα (Open): Η τιμή της μετοχής κατά το άνοιγμα της αγοράς. Είναι η πρώτη τιμή που καταγράφεται κατά το άνοιγμα των συναλλαγών.
- Υψηλό (High): Η υψηλότερη τιμή που έφτασε η μετοχή κατά τη διάρκεια της ημερήσια συνεδρίας.
- Χαμηλό (Low): Η χαμηλότερη τιμή που έφτασε η μετοχή κατά τη διάρκεια της ημερήσια συνεδρίας.
- Όγκος (Volume): Ο συνολικός όγκος των μετοχών που ανταλλάχθηκαν κατά τη διάρκεια της ημερήσια συνεδρίας.
- Κλείσιμο (Close): Η τελική τιμή στην οποία κλείνει η μετοχή στο τέλος της ημερήσιας συνεδρίας.

1.3.5 Δείκτης χρηματιστηρίου

Οι χρηματιστηριακοί δείκτες είναι χρήσιμα εργαλεία που χρησιμοποιούνται σε διάφορους τομείς της χρηματοοικονομικής ανάλυσης και των επενδύσεων. Εκτός από τη μέτρηση της συνολικής αξίας μιας αγοράς ή ενός συγκεκριμένου τομέα της, με βάση τις μέσες τιμές των μετοχών, παρέχουν κρίσιμες πληροφορίες για την απόδοση, τον κίνδυνο και την τάση της αγοράς.

Κάποιες κύριες χρήσεις είναι:

- Ως μέσο υπολογισμού της απόδοσης και του κινδύνου μιας αγοράς στο σύνολό της ή ενός τμήματος της.
- Ως μέτρο σύγκρισης της απόδοσης μιας μετοχής.
- Ως βάση παροχής πληροφοριών για την τεχνική ανάλυση μετοχών.
- Για την εξαγωγή συμπερασμάτων σχετικά με την κίνηση των χρηματιστηρίων και τη συσχέτισή τους με άλλα οικονομικά φαινόμενα.
- Ως μέθοδος υπολογισμού του συστημικού κινδύνου.

1.3.6 Δείκτης S&P 500

Ο δείκτης S&P 500 περιλαμβάνει 500 κορυφαίες εταιρείες που είναι εισηγμένες στο χρηματιστήριο των Ηνωμένων Πολιτειών και αποτυπώνει περίπου το 80% κάλυψης της διαθέσιμης κεφαλαιοποίησης της αγοράς. Όπως όλοι οι σημαντικοί δείκτες, ο S&P 500 χρησιμοποιεί το πρότυπο GICS (Global Industry Classification Standard, Παγκόσμιο πρότυπο ταξινόμησης κλάδων/τομέων) για να κατατάξει τις εταιρείες σε τομείς όπως ενέργεια, υγεία, χρηματοοικονομικές υπηρεσίες, πληροφορική και λιανικό εμπόριο.

1.3.7 Χρηματιστήριο και Μηχανική Μάθηση

Σε οποιοδήποτε κλάδο υπάρχει ο ανθρώπινος παράγοντας, έτσι και στην αγοραπωλησία μετοχών μπορούν να συμβούν λάθη. Μία λύση στο πρόβλημα είναι η χρήση μηχανικής μάθησης με πλεονεκτήματα όπως (Shavandi & Khedmati, 2022):

- Ταχύτητα: Οι υπολογιστές είναι πολύ γρήγοροι στην εκτέλεση συναλλακτικών αποφάσεων.
- Ακρίβεια: Σε διάφορες συναλλαγές οι άνθρωποι είναι επιρρεπείς σε λάθη, ενώ οι υπολογιστές είναι πιο ακριβείς στην εκτέλεση συναλλαγών από τους ανθρώπους.
- Ορθολογικότητα: Οι άνθρωποι μπορεί να λάβουν λανθασμένες εμπορικές αποφάσεις σχετικά με τους συναισθηματικούς και ψυχολογικούς τους παράγοντες, ενώ οι υπολογιστές δεν επηρεάζονται αρνητικά από αυτούς τους παράγοντες στη λήψη αποφάσεων.
- Ικανότητα επεξεργασίας: Οι υπολογιστές είναι πολύ πιο ικανοί από τους ανθρώπους στην επεξεργασία τεράστιου όγκου πληροφοριών σε πραγματικό χρόνο.
- Εγρήγορση: Οι υπολογιστές μπορούν να παρακολουθούν μόνιμα τις αγορές σε σχέση με τον άνθρωπο οπότε να λαμβάνουν άμεσες αντιδράσεις όταν χρειάζεται.

Σε διάφορες δημοσιεύσεις όπως (Shavandi & Khedmati, 2022), (Théate & Ernst, 2021), έχουν αναφερθεί σε πολύ θετικά αποτελέσματα με την χρήση ενισχυτικής μάθησης σε αλγοριθμικούς για συναλλαγές στις χρηματοπιστωτικές αγορές.

1.4 Εισαγωγή στην βραχυπρόθεσμη πρόβλεψη μετοχών και αυτοματοποίηση συναλλαγών

Η βραχυπρόθεσμη πρόβλεψη τιμών των μετοχών αποτελεί έναν από τους πιο ενδιαφέροντες και περίπλοκους τομείς της χρηματοοικονομικής επιστήμης. Ειδικά σε περιβάλλοντα όπου οι αγορές είναι εξαιρετικά ασταθείς και απρόβλεπτες, η δυνατότητα πρόβλεψης μεταβολών στις τιμές των μετοχών μπορεί να αποβεί καθοριστική για την λήψη επενδυτικών αποφάσεων που μεγιστοποιούν τα κέρδη (Zhang et al., 2020). Αυτή η ικανότητα γίνεται ολοένα και πιο κρίσιμη, καθώς η ταχύτητα και ο όγκος των συναλλαγών στις σύγχρονες χρηματοοικονομικές αγορές αυξάνεται. Αυτό συμβαίνει σε μεγάλο βαθμό διότι έχει αυξηθεί η συμμετοχή αλγοριθμικών συναλλαγών (Avellaneda & Stoikov, 2008).

Ο όρος «βραχυπρόθεσμη πρόβλεψη» αναφέρεται στη διαδικασία εκτίμησης των τιμών των μετοχών για κάποιο χρονικό διάστημα, συνήθως μερικών ωρών ή ημερών. Όπως είναι λογικό αυτοί οι αλγόριθμοι που χρησιμοποιούνται σε αυτό τον τομέα πρέπει να είναι ικανοί να αναλύσουν τεράστιους όγκους δεδομένων σε πραγματικό χρόνο, ώστε να διακρίνουν μοτίβα και τάσεις που μπορούν να προσφέρουν χρήσιμες προβλέψεις στις τιμές των μετοχών (Xu, 2021).

1.5 Η αναγκαιότητα της αυτοματοποίησης

Στον κόσμο των χρηματοπιστωτικών συναλλαγών, η ταχύτητα με την οποία λαμβάνονται και εκτελούνται οι αποφάσεις παίζει σημαντικό ρόλο. Σε πολλά χρηματιστήρια παγκοσμίως, ακόμα και μια διαφορά δευτερολέπτων μπορεί να οδηγήσει σε εντελώς διαφορετικά αποτελέσματα (Xu, 2021). Ειδικότερα στην περίπτωση της βραχυπρόθεσμης πρόβλεψης, συνήθως οι συναλλαγές που εκτελούνται από αυτοματοποιημένα συστήματα υπερτερούν σε σχέση με τον άνθρωπο. Οι υπολογιστές μπορούν να επεξεργαστούν τεράστιους όγκους δεδομένων σε ελάχιστο χρόνο. Επίσης η αυτοματοποίηση όχι μόνο αυξάνει την ταχύτητα εκτέλεσης και την ακρίβεια των συναλλαγών, αλλά επιπλέον μειώνει τον κίνδυνο που προκύπτει από ανθρώπινα λάθη ή από λανθασμένες συναισθηματικές εκτιμήσεις. Υπάρχουν φορές όπου οι επενδυτές παίρνουν αποφάσεις βασισμένοι σε ψυχολογικούς παράγοντες όπως ο φόβος της απώλειας ή η αυτοπεποίθηση μετά από

κάποιες επιτυχημένες συναλλαγές (Zhang et al., 2020). Έτσι η αυτοματοποίηση της διαδικασίας αγοραπωλησίας μετοχών κρίνεται απαραίτητη για όποιον θέλει να ανταγωνιστεί στις σύγχρονες αγορές. Τα αυτόματα συστήματα είναι προγραμματισμένα να ακολουθούν προκαθορισμένες στρατηγικές, οι οποίες βασίζονται αποκλειστικά σε δεδομένα και μαθηματικά μοντέλα, αποφεύγοντας αυτούς τους κινδύνους (Avellaneda & Stoikov, 2008).

1.5.1 Προκλήσεις στην αυτοματοποίηση συναλλαγών

Παρά τα πολλά πλεονεκτήματα της αυτοματοποίησης στην χρηματιστηριακή αγορά, υπάρχουν και ορισμένες σημαντικές προκλήσεις που πρέπει να ληφθούν υπόψη. Η βασικότερη είναι η πολυπλοκότητα και το δυναμικό περιβάλλον των χρηματοπιστωτικών αγορών. Οι αγορές επηρεάζονται από πλήθος παραγόντων, όπως οικονομικά νέα, γεωπολιτικές εξελίξεις και απρόβλεπτες καταστάσεις, γεγονός που καθιστά την πρόβλεψη των τιμών ιδιαίτερα δύσκολη (Zhang et al, 2020).

Επιπλέον όπως έχουμε ήδη αναφέρει, οι αλγόριθμοι ενισχυτικής μάθησης απαιτούν τεράστιες ποσότητες δεδομένων για την εκπαίδευσής τους, όπου τα δεδομένα μπορεί να περιλαμβάνουν όχι μόνο ιστορικές τιμές μετοχών, αλλά και δεδομένα όγκου συναλλαγών, δείκτες αγοράς, καθώς και εξωτερικά δεδομένα όπως οικονομικές ειδήσεις, πολιτικές εξελίξεις και γεωπολιτικά δεδομένα. Χωρίς επαρκή δεδομένα, υπάρχει ο κίνδυνος υπό-προσαρμογής (underfitting) του μοντέλου, με αποτέλεσμα να λαμβάνει λανθασμένες αποφάσεις (Xu, 2021). Η ποιότητα και η ποσότητα των δεδομένων, καθώς και οι υπολογιστικοί πόροι που απαιτούνται για την λειτουργία των συστημάτων, είναι καθοριστικής σημασίας για την επιτυχία τους

1.6 Ενισχυτική μάθηση (Reinforcement Learning)

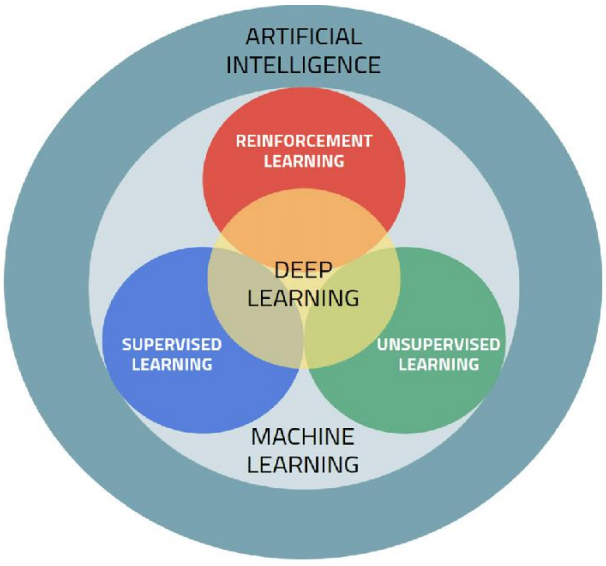
Η ενισχυτική μάθηση όπως προαναφέρθηκε είναι ένας τύπος μηχανικής μάθησης όπου ένας πράκτορας (agent) μαθαίνει να επιλέγει καταστάσεις (states) και ενέργειες (actions) μέσω αλληλεπίδρασης με το περιβάλλον του (environment), ώστε να μεγιστοποιήσει την ανταμοιβή (reward) κατά τον τελικό στόχο που του έχει δοθεί. Κάθε ενέργεια αποφέρει

στον πράκτορα κάποια ανταμοιβή, θετική ή αρνητική. Ο πράκτορας μαθαίνει από τις συνέπειες των επιλογών του, αντί να του καθορίζει κάποιος τι ενέργειες να επιλέξει. Αυτή η τακτική δοκιμής και λάθους είναι βασικό στοιχείο στην ενισχυτική μάθηση.

Τα προβλήματα ενισχυτικής μάθησης περιλαμβάνουν την μάθηση του πράκτορα ώστε να επιλέγει ενέργειες που θα του αποφέρουν την μεγαλύτερη ανταμοιβή δοκιμάζοντας. Οι πράκτορες έχουν σαφής στόχους και μπορούνε να επιλέξουν ενέργειες για να επηρεάσουν το περιβάλλον τους. Σε περιπτώσεις, οι ενέργειες επηρεάζουν όχι μόνο την ανταμοιβή αλλά και την επόμενη κατάσταση, και μέσω αυτής, όλες τις επόμενες ανταμοιβές. Αυτά τα χαρακτηριστικά, ότι δεν υπάρχουν άμεσες οδηγίες για το ποιες ενέργειες πρέπει να γίνουν και οι συνέπειες των ενεργειών, είναι από τα πιο σημαντικά χαρακτηριστικά των προβλημάτων της ενισχυτικής μάθησης.

Από όλες τις μορφές μηχανικής μάθησης, η ενισχυτική μάθηση είναι η πιο κοντινή στο είδος της μάθησης που κάνουν οι άνθρωποι και άλλα ζώα και πολλοί από τους βασικούς αλγόριθμους της ενισχυτικής μάθησης εμπνεύστηκαν αρχικά από βιολογικά συστήματα μάθησης (Sutton & Barto, 2018).

Η ενισχυτική μάθηση έχει πολλές εφαρμογές, με μία από τις σημαντικότερες να βρίσκεται στον τομέα των χρηματοπιστωτικών αγορών, όπου οι πράκτορες μπορούν να μάθουν να βελτιστοποιούν τις συναλλαγές και τις επενδυτικές στρατηγικές τους.



Εικόνα 1: Βασικές κατηγορίες μάθησης στην τεχνητή νοημοσύνη (Karthikeyan & Priyakumar, 2021)

1.6.1 Ενισχυτική μάθηση vs Εποπτευόμενη μάθηση (Supervised Learning)

Η ενισχυτική μάθηση διαφέρει σημαντικά από την εποπτευόμενη μάθηση, η οποία είναι μια μέθοδος μηχανικής μάθησης όπου ένα μοντέλο εκπαιδεύεται χρησιμοποιώντας δεδομένα που έχουν ήδη επισημανθεί με τις σωστές απαντήσεις. Αυτό σημαίνει ότι στην εποπτευόμενη μάθηση, το μοντέλο μαθαίνει από δεδομένα εκπαίδευσης τα οποία έχουν επισημανθεί με τις σωστές απαντήσεις για να προβλέψει σωστά αποτελέσματα για νέα, άγνωστα δεδομένα.

Παρόλο που είναι ένα σημαντικό είδος μάθησης, από μόνο του δεν επαρκεί για τη αντιμετώπιση προβλημάτων που αλληλοεπιδρούν με το περιβάλλον. Σε αλληλεπιδραστικά προβλήματα είναι συχνά ανέφικτο να βρεθούν παραδείγματα επιθυμητής συμπεριφοράς που να καλύπτουν όλες τις πιθανές καταστάσεις στις οποίες πρέπει να δράσει ο πράκτορας. Στην ενισχυτική μάθηση, ο πράκτορας πρέπει να μαθαίνει από τη δική του εμπειρία, ειδικά όταν βρίσκεται σε αχαρτογράφητες περιοχές, όπου η μάθηση μέσω της αλληλεπίδρασης είναι πιο ωφέλιμη. Αυτό συμβαίνει διότι επιτρέπει στον πράκτορα να μαθαίνει αυτόνομα, να εξερευνά νέες καταστάσεις και να προσαρμόζεται στις αλλαγές του περιβάλλοντος, χωρίς την ανάγκη προκαθορισμένων λύσεων ή προσημασμένων δεδομένων.

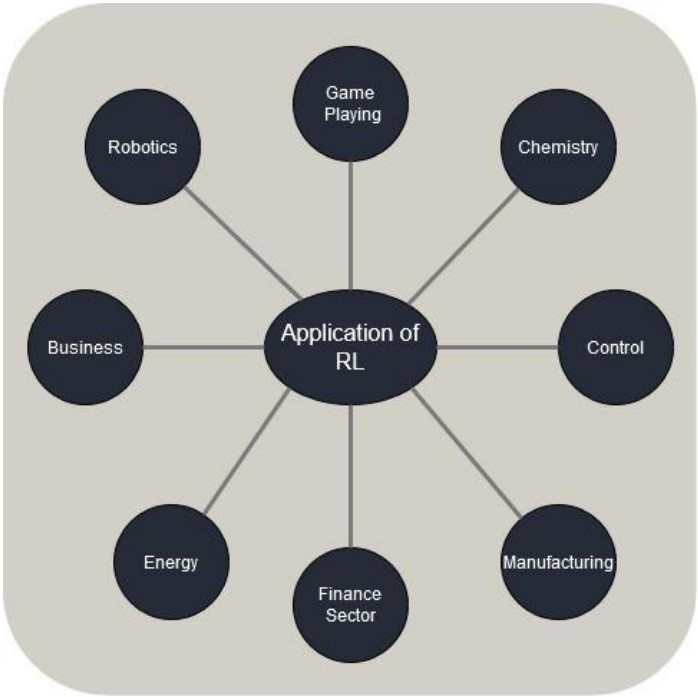
1.6.2 Ενισχυτική μάθηση vs Μη εποπτευόμενη μάθηση (Unsupervised Learning)

Η ενισχυτική μάθηση διαφέρει επίσης από τη μη εποπτευόμενη μάθηση. Στη μη εποπτευόμενη μάθηση, ο αλγόριθμος εκπαιδεύεται σε δεδομένα χωρίς ετικέτες και ο κύριος στόχος είναι η ανακάλυψη κρυμμένων μοτίβων ή δομών στα δεδομένα, όπως ομάδες (clusters) ή σχέσεις μεταξύ των δεδομένων. Από την άλλη, στην ενισχυτική μάθησης, ο πράκτορας αλληλοεπιδρά με το περιβάλλον του και στόχος είναι η βελτιστοποίηση της στρατηγικής ώστε να μεγιστοποιήσει την αθροιστική ανταμοιβή που λαμβάνει, βασιζόμενος στις ενέργειες που επιλέγει.

1.6.3 Πεδίο εφαρμογής ενισχυτικής μάθησης

Η ενισχυτική μάθηση είναι ένας από τους σημαντικότερους και πιο εξελικτικούς κλάδους της τεχνητής νοημοσύνης και της μηχανικής μάθησης. Ως εκ τούτου έχει πεδίο εφαρμογής σε καίριες τεχνολογίες όπως:

- Στην ρομποτική, όπου η ενισχυτική μάθηση χρησιμοποιείται για να διδάξει τα ρομπότ πώς να κινούνται και να αλληλοεπιδρούν με τον φυσικό κόσμο.
- Σε παιχνίδια και προσομοιώσεις με εκπληκτικά αποτελέσματα όπως θα αναφέρουμε και παρακάτω.
- Στην ενέργεια και στο περιβάλλον, όπου με την βοήθεια της ενισχυτικής μάθησης μπορεί να γίνει προσαρμογή της κατανάλωσης του ρεύματος ώστε να μειωθεί η σπατάλη και το κόστος.
- Στην αυτόνομη οδήγηση όπου οι αλγόριθμοι ενισχυτικής μάθησης μαθαίνουν πώς να οδηγούν με ασφάλεια για την αποφυγή ατυχημάτων όπου το περιβάλλον είναι δυναμικό, όπως η κυκλοφορία πεζών και άλλων οχημάτων.
- Στις χρηματοπιστωτικές αγορές, όπου οι διακυμάνσεις των τιμών είναι απρόβλεπτες.



Εικόνα 2: Πεδία χρήσης ενισχυτικής μάθησης

1.6.4 Πλεονεκτήματα ενισχυτικής μάθησης

Κάποια από τα πλεονεκτήματα της ενισχυτικής μάθησης είναι τα παρακάτω:

- Ικανότητα να μαθαίνει μέσω της εμπειρίας και να προσαρμόζει τις αποφάσεις σε πραγματικό χρόνο
- Μπορεί να χρησιμοποιηθεί για την επίλυση σύνθετων προβλημάτων όπως εκείνων που αφορούν τη λήψη αποφάσεων, τον έλεγχο και τη βελτιστοποίηση.
- Λόγω της ευέλικτης προσέγγισή της, μπορεί να συνδυαστεί με άλλες τεχνικές μηχανικής μάθησης, όπως η βαθιά μάθηση, για τη βελτίωση της απόδοσης.
- Τα δεδομένα εκπαίδευσης λαμβάνονται μέσω της άμεσης αλληλεπίδρασης του πράκτορα με το περιβάλλον με αποτέλεσμα την μάθηση από την εμπειρία, να προσαρμόζεται σε πραγματικές καταστάσεις και να εξερευνεί ώστε να εκμεταλλευτεί καταστάσεις για βελτιστοποιημένες λύσεις.
- Το μοντέλο διορθώνει τα σφάλματα που προέκυψαν κατά τη διαδικασία εκπαίδευσης.
- Το μοντέλο είναι ικανό να χειριστεί περιβάλλοντα που δεν είναι ντετερμινιστικά, δηλαδή τα αποτελέσματα των ενεργειών δεν μπορούν να είναι πάντα προβλέψιμα. Αυτό είναι χρήσιμο σε εφαρμογές του πραγματικού κόσμου όπου το περιβάλλον μπορεί να αλλάζει με την πάροδο του χρόνου ή να είναι αβέβαιο.

1.6.5 Μειονεκτήματα ενισχυτικής μάθησης

Όπως κάθε τύπος μάθησης, έτσι και η ενισχυτική μάθηση έχει μειονεκτήματα, κάποια από τα οποία είναι:

- Χρειάζεται πολλούς υπολογιστικούς πόρους και χρόνο για την εκπαίδευση των αλγορίθμων της.
- Έχει ανάγκη από πολλά δεδομένα ώστε να εκπαιδευτούν αποτελεσματικά οι αλγόριθμοι ενισχυτικής μάθησης.
- Εξαρτάται σε μεγάλο βαθμό από την ποιότητα της συνάρτησης ανταμοιβής. Εάν η συνάρτηση ανταμοιβής είναι κακώς σχεδιασμένη, ο πράκτορας ενδέχεται να μην μάθει την επιθυμητή συμπεριφορά.

- Η δημιουργία και ρύθμιση ενός μοντέλου ενισχυτικής μάθησης μπορεί να είναι χρονοβόρα και πολύπλοκη.
- Μπορεί να είναι δύσκολη στην απασφαλμάτωση και την ερμηνεία. Δεν είναι πάντα σαφές γιατί ο πράκτορας συμπεριφέρεται με έναν συγκεκριμένο τρόπο, πράγμα που δυσκολεύει τη διάγνωση και τη διόρθωση προβλημάτων.

1.7 Γιατί Ενισχυτική Μάθηση σε χρηματοοικονομικές συναλλαγές

Στην ενισχυτική μάθηση ένα από τα κύρια χαρακτηριστικά της που το συναντάμε σε όντα με νοημοσύνη, είναι ότι έχει την δυνατότητα να μαθαίνει και όχι απλά να εκτελεί, που σε ένα δυναμικό περιβάλλον όπως του χρηματιστηρίου είναι απαραίτητο.

Για συναλλαγές που στηρίζονται σε ανάλυση δεδομένων η διαδικασία βασίζεται συνήθως σε δύο βασικά βήματα. Την πρόβλεψη και την εκμετάλλευση των προβλέψεων για τη λήψη αποφάσεων. Η πρόβλεψη μπορεί να αφορά διάφορες χρονικές στιγμές, από λεπτά και ώρες έως ημέρες και εβδομάδες. Για να προβλεφθούν τιμές και τάσεις της αγοράς, χρησιμοποιούνται διάφορες τεχνικές κάποιες από τις οποίες θα δούμε παρακάτω.

1.7.1 Προσαρμοστικότητα και μάθηση μέσω αλληλεπίδρασης

Η επιλογή της ενισχυτικής μάθησης (RL) για την ανάπτυξη στρατηγικών αγοραπωλησίας μετοχών δεν είναι τυχαία, αλλά πηγάζει από την ίδια τη φύση του προβλήματος και τα μοναδικά χαρακτηριστικά της ενισχυτικής μάθησης. Οι χρηματοοικονομικές αγορές είναι δυναμικά, πολύπλοκα συστήματα, όπου οι βέλτιστες αποφάσεις εξαρτώνται από μια συνεχή ροή πληροφοριών και αλλάζουν με την πάροδο του χρόνου. Η ενισχυτική μάθηση προσφέρει ένα πλαίσιο ιδανικά προσαρμοσμένο σε αυτές τις προκλήσεις για τους εξής λόγους:

- Μίμηση της ανθρώπινης μάθησης: Ένα από τα ισχυρότερα πλεονεκτήματα της ενισχυτικής μάθησης είναι ότι μιμείται τον τρόπο με τον οποίο οι έμπειροι επενδυτές μαθαίνουν – δηλαδή μέσω της συνεχούς δοκιμής, του εντοπισμού λαθών και της παρατήρησης των συνεπειών των αποφάσεών τους. Ο πράκτορας της RL αλληλεπιδρά με ένα περιβάλλον προσομοίωσης της αγοράς, εκτελεί συναλλαγές

(δράσεις), λαμβάνει ανταμοιβές ή ποινές (κέρδη/ζημιές, προσαρμοσμένες στον κίνδυνο), και σταδιακά βελτιώνει τη στρατηγική του (policy) για να μεγιστοποιήσει την ανταμοιβή του. Αυτή η διαδικασία μάθησης από την εμπειρία είναι θεμελιώδης για την προσαρμογή στις συνεχώς μεταβαλλόμενες συνθήκες της αγοράς.

- Λήψη διαδοχικών αποφάσεων: Το trading είναι εγγενώς ένα πρόβλημα διαδοχικής λήψης αποφάσεων. Κάθε απόφαση αγοράς, πώλησης ή αναμονής επηρεάζει όχι μόνο την άμεση ανταμοιβή αλλά και τις μελλοντικές καταστάσεις και ευκαιρίες. Η ενισχυτική μάθηση είναι σχεδιασμένη για να χειρίζεται τέτοια προβλήματα, καθώς ο πράκτορας μαθαίνει να λαμβάνει υπόψη τις μακροπρόθεσμες συνέπειες των ενεργειών του. Αυτό επιτυγχάνεται μέσω της εκπαίδευσης και βελτιστοποίησης της Q-function (action-value function), η οποία εκτιμά την συνολική αναμενόμενη μελλοντική ανταμοιβή για κάθε ζεύγος κατάστασης-δράσης (με την καθοδήγηση της συνάρτησης ανταμοιβής, η οποία θα αναλυθεί σε μετέπειτα κεφάλαιο).
- Αυτονομία και προσαρμοστικότητα: Σε αντίθεση με άλλες μεθόδους, για παράδειγμα της εποπτευόμενης μάθησης που απαιτούν επισημασμένα δεδομένα (π.χ., ιστορικά σημεία όπου μία συγκεκριμένη δράση – αγορά, πώληση ή αναμονή, θεωρήθηκε βέλτιστη), η ενισχυτική μάθηση επιτρέπει στον πράκτορα να ανακαλύψει στρατηγικές αυτόνομα. Ο πράκτορας καθοδηγούμενος από τη συνάρτηση ανταμοιβής που έχει σχεδιαστεί για να αντικατοπτρίζει τους τελικούς στόχους (δηλαδή την μεγιστοποίηση των κερδών), μαθαίνει μια πολιτική (policy) λήψης αποφάσεων και να προσαρμόζεται σε νέα μοτίβα της αγοράς που μπορεί να μην υπήρχαν στα αρχικά δεδομένα εκπαίδευσης. Η ικανότητα του πράκτορα να εκπαιδεύεται και να βελτιστοποιεί την απόδοση για τελικά κέρδη, του δίνει ένα σημαντικό πλεονέκτημα έναντι άλλων μοντέλων.
- Διαχείριση της αβεβαιότητας: Οι χρηματοοικονομικές αγορές χαρακτηρίζονται από αβεβαιότητα. Η ενισχυτική μάθηση μπορεί να ενσωματώσει αυτή την αβεβαιότητα, καθώς ο πράκτορας μαθαίνει μια πολιτική που είναι εύρωστη (robust) σε ένα εύρος πιθανών εξελίξεων της αγοράς. Μέσω της εξερεύνησης, ο πράκτορας μπορεί να δοκιμάσει διαφορετικές προσεγγίσεις και να μάθει πώς να αντιδρά σε απρόβλεπτες καταστάσεις.

Ενώ άλλες τεχνικές μηχανικής μάθησης, όπως η εποπτευόμενη μάθηση για πρόβλεψη τιμών, μπορούν να είναι χρήσιμες ως μέρος μιας ευρύτερης στρατηγικής trading, η

ενισχυτική μάθηση προσφέρει ένα ολοκληρωμένο πλαίσιο για την εκμάθηση της ίδιας της διαδικασίας λήψης αποφάσεων σε ένα δυναμικό και ανταποδοτικό περιβάλλον, καθιστώντας την κατάλληλη για την ανάπτυξη αυτόνομων συστημάτων αλγοριθμικού trading.

1.7.2 Τεχνικές πρόβλεψης τιμών και τάσεων

Χρονοσειρές στις χρηματοοικονομικές εφαρμογές

Η κατανόηση των χρονοσειρών είναι απαραίτητη για τον σχεδιασμό και την ανάπτυξη αλγορίθμων ενισχυτικής μάθησης, καθώς οι χρονοσειρές περιγράφουν τη δυναμική των αγορών και τις μεταβολές των τιμών με την πάροδο του χρόνου. Στο πλαίσιο των χρηματοοικονομικών εφαρμογών, η ανάλυση χρονοσειρών επιτρέπει στον πράκτορα να μάθει και να προσαρμόζεται σε μοτίβα και τάσεις, βελτιώνοντας τη στρατηγική του.

Βασικά χαρακτηριστικά των χρονοσειρών (Time series analysis)

Όλες οι χρονοσειρές ανεξαρτήτως περιγραφόμενου μεγέθους (τιμή κλεισίματος μετοχής, πωλήσεις αντικειμένων κτλ.) παρουσιάζουν ορισμένα βασικά χαρακτηριστικά.

- Στασιμότητα (Stationary): Μια χρονοσειρά λέγεται στατική όταν σε δύο διαδοχικές χρονικές στιγμές τα στατιστικά μεγέθη μίας χρονοσειράς (μέση τιμή, διακύμανση) δεν έχουν μεταβληθεί. Αυτό είναι σημαντικό για τη σταθερότητα και την προβλεψιμότητα του μοντέλου διότι πολλές μεθοδολογίες ανάλυσης και πρόβλεψης απαιτούν στατικές χρονοσειρές.
- Τάση (Trend): Η τάση δείχνει τη γενική πορεία στην οποία κινούνται τα δεδομένα σε μια συγκεκριμένη χρονική περίοδο. Η αναγνώριση της τάσης, επιτρέπει στον αλγόριθμο να εκτιμά καλύτερα τη μακροπρόθεσμη κατεύθυνση της αγοράς. Έχει παρατηρηθεί ότι οι τάσεις μπορεί να αυξηθούν, να μειωθούν ή να είναι σταθερές.
    - Αυξητική τάση (Uptrend): Όταν τα δεδομένα αυξάνονται συνολικά με την πάροδο του χρόνου. Χαρακτηρίζεται από συνεχείς υψηλότερες κορυφές (Higher highs) και υψηλότερα χαμηλά (Higher lows). Ένα παράδειγμα η

συνεχής αύξηση των πωλήσεων ενός προϊόντος κατά τη διάρκεια ενός έτους.

- Μειωτική Τάση (Downtrend): Όταν τα δεδομένα μειώνονται συνολικά με την πάροδο του χρόνου. Χαρακτηρίζεται από χαμηλότερες κορυφές (Lower highs) και χαμηλότερα χαμηλά (Lower lows). Ένα παράδειγμα είναι η πτώση της τιμής μιας μετοχής κατά τη διάρκεια αρκετών μηνών.
- Πλαγιοκαθοδική τάση (Sideways Trend): Όταν τα δεδομένα κινούνται γύρω από μια σταθερή τιμή, χωρίς σαφή αυξητική ή μειωτική κατεύθυνση. Αυτή η τάση εμφανίζεται όταν η αγορά βρίσκεται σε ισορροπία. Ένα παράδειγμα όταν οι σταθερές τιμές του πετρελαίου κατά τη διάρκεια ενός μήνα, χωρίς μεγάλες διακυμάνσεις.

- Περιοδικότητα ή Εποχικότητα (Seasonal): Η εποχικότητα αναφέρεται στην εμφάνιση κανονικών και προβλέψιμων μοτίβων σε συγκεκριμένα χρονικά διαστήματα, όπως οι πωλήσεις που αυξάνονται την περίοδο των γιορτών. Η εποχικότητα έχει σταθερή και γνωστή περίοδο. Οι εποχικές χρονοσειρές ονομάζονται μερικές φορές περιοδικές χρονοσειρές (Hyndman, 2011). Ανιχνεύοντας εποχικά μοτίβα, ο πράκτορας μπορεί να προσαρμόσει την στρατηγική του ώστε να εκμεταλλευτεί επαναλαμβανόμενες ευκαιρίες αγοράς και πώλησης.
- Κυκλικότητα (Cyclicality): Αναφέρεται σε επαναλαμβανόμενα μοτίβα ή διακυμάνσεις που εμφανίζονται σε συγκεκριμένα χρονικά διαστήματα, αλλά όχι απαραίτητα με την ίδια διάρκεια ή συχνότητα. Η αναγνώριση τέτοιων κυκλικών φάσεων, βοηθά στην ανάλυση μεγαλύτερων οικονομικών κύκλων και την προσαρμογή του πράκτορα για μακροπρόθεσμες μεταβολές της αγοράς.
- Λευκός θόρυβος (White noise): Μια χρονοσειρά λευκού θορύβου δεν έχει τάση, περιοδικότητα ή προβλέψιμα μοτίβα. Τέτοιες χρονοσειρές είναι εντελώς τυχαίες και δεν περιέχουν αυτοσυσχέτιση μεταξύ των παρατηρήσεων. Αυτό σημαίνει ότι δεν υπάρχει προβλεψιμότητα ή μοτίβο που να μπορεί να αξιοποιηθεί.
- Αυτοσυσχέτιση (Autocorrelation): Είναι το μέτρο του κατά πόσο οι τιμές της χρονοσειράς επηρεάζονται από προηγούμενες τιμές. Η ανάλυση της αυτοσυσχέτισης είναι σημαντική για τον προσδιορισμό εάν οι προηγούμενες τιμές μπορούν να χρησιμοποιηθούν για την πρόβλεψη μελλοντικών κινήσεων.

- Μεταβλητότητα (Volatility): Η μεταβλητότητα είναι κρίσιμη για την εκτίμηση του κινδύνου και την ανάπτυξη στρατηγικών συναλλαγών. Οι αλγόριθμοι που χρησιμοποιούν ενισχυτική μάθηση πρέπει να κατανοούν και να προσαρμόζονται σε περιόδους υψηλής και χαμηλής μεταβλητότητας για τη βελτιστοποίηση της στρατηγικής τους.
- Μέθοδοι αποδόμησης (Decomposition Methods): Οι τεχνικές αποδόμησης επιτρέπουν την ανάλυση μιας χρονοσειράς σε επιμέρους συνιστώσες, όπως τάση, εποχικότητα και θόρυβο, για καλύτερη κατανόηση της δομής της.

Μηχανική Μάθηση (Machine Learning)

Η μηχανική μάθηση περιλαμβάνει την χρήση αλγορίθμων που μπορούν να μάθουν από τα δεδομένα και να κάνουν προβλέψεις ή αποφάσεις χωρίς να προγραμματιστούν ρητά. Αυτοί οι αλγόριθμοι είναι σχεδιασμένοι να αναγνωρίζουνε πρότυπα, να εξάγουν γνώσεις από τα δεδομένα και να προσαρμόζονται καθώς μαθαίνουν περισσότερα.

Στην πρόβλεψη της χρηματιστηριακής αγοράς, τα μοντέλα μηχανικής μάθησης αναλύουν ιστορικά δεδομένα αγοράς για να εντοπιστούν πρότυπα και σχέσεις μεταξύ διάφορων παραγόντων. Αυτά τα μοντέλα μπορούν να προσαρμοστούν και να βελτιωθούν με την πάροδο του χρόνου. Είναι πολύτιμα εργαλεία για την πρόβλεψη των τιμών των μετοχών και τη διαχείριση κινδύνου. Κάποια μοντέλα όπως έχουν αναφερθεί παραπάνω είναι η εποπτευόμενη μάθηση, μη εποπτευόμενη μάθηση και η ενισχυτική μάθηση.

Ανάλυση Συναισθήματος (Sentiment analysis)

Η ανάλυση συναισθήματος αποτελεί μια τεχνική που ανήκει στο data mining και χρησιμοποιείται για την αναγνώριση και αξιολόγηση συναισθημάτων χρηστών πάνω σε διάφορα θέματα όπως ειδησεογραφικά άρθρα, οικονομικές αναφορές, σχόλια σε φόρουμ επενδυτών και αναρτήσεις στα μέσα κοινωνικής δικτύωσης. Κατανοώντας αν το συναίσθημα που εκφράζεται είναι θετικό, αρνητικό ή ουδέτερο, βοηθά στον εντοπισμό τυχόν τάσεων. Αυτή η προσέγγιση είναι ιδιαίτερα χρήσιμη στον τομέα της χρηματιστηριακής αγοράς καθώς επενδυτές και αλγόριθμοι μπορούν να κατανοήσουν τη συνολική διάθεση της αγοράς και να επενδύσουν ανάλογα.

Η ενσωμάτωση της ανάλυσης συναισθήματος σε έναν αλγόριθμο ενισχυτικής μάθησης μπορεί να προσφέρει πρόσθετες πληροφορίες στον πράκτορα, βοηθώντας τον να λαμβάνει πιο ενημερωμένες αποφάσεις.

Τεχνική Ανάλυση (Technical analysis)

Η τεχνική ανάλυση μιας μετοχής, ενός δείκτη ή μιας ολόκληρης αγοράς, είναι μια μεθοδολογία για την ανάλυση και την πρόβλεψη της κατεύθυνσης των τιμών, μέσω της μελέτης δεδομένων της αγοράς, κυρίως των τιμών και του όγκου. Για να επιτευχθεί κάτι τέτοιο, οι αναλυτές χρησιμοποιούν γραφήματα και μια σειρά από τεχνικούς δείκτες.

Με την τεχνική ανάλυση αναζητούμε πιθανές συμπεριφορές στο παρελθόν που να μας βοηθήσουν ώστε να εφαρμόσουμε μια επενδυτική στρατηγική, βασιζόμενοι στο σκεπτικό ότι συμπεριφορές τείνουν να επαναλαμβάνονται με παρόμοιο τρόπο.

Ποσοτική Ανάλυση (Quantitative analysis)

Η ποσοτική ανάλυση είναι μια προσέγγιση που χρησιμοποιεί μαθηματικές και στατιστικές μεθόδους στη χρηματοοικονομική και τη διαχείριση επενδύσεων. Βοηθά στην αξιολόγηση μέσω της πρόβλεψης. Περιλαμβάνει αρκετές τεχνικές κάποιες από τις οποίες είναι οι παρακάτω:

- Ανάλυση παλινδρόμησης (Regression analysis): Η ανάλυση παλινδρόμησης είναι κοινή τεχνική στην ποσοτική ανάλυση. Είναι μια στατική μέθοδος που αναλύει τη σχέση μεταξύ μιας εξαρτημένης μεταβλητής και μίας ή περισσοτέρων ανεξάρτητων μεταβλητών. Στην χρηματοοικονομική ανάλυση, αυτή η τεχνική μπορεί να χρησιμοποιηθεί για να κατανοηθεί πως οι οικονομικοί δείκτες ή άλλοι παράγοντες επηρεάζουν την τιμή μιας μετοχής ή ενός χρηματοοικονομικού προϊόντος.
- Γραμμικός προγραμματισμός (Linear programming): Ο γραμμικός προγραμματισμός είναι μια μέθοδος βελτιστοποίησης που χρησιμοποιείται και στην χρηματοοικονομική ανάλυση για την επίλυση προβλημάτων, όπως η κατανομή πόρων, εξισορρόπηση χαρτοφυλακίου και επίτευξη επενδυτικών στόχων υπό ένα δεδομένο σύνολο περιορισμών όπως η διαχείριση κινδύνου και η απόδοση.

- Εξόρυξη δεδομένων (Data mining): Η εξόρυξη δεδομένων είναι ένας συνδυασμός προγραμματισμού και στατιστικών μεθόδων. Οι τεχνικές εξόρυξης δεδομένων χρησιμοποιούνται για την αξιολόγηση πολύ μεγάλων συνόλων δεδομένων για την εύρεση μοτίβων ή συσχετίσεων που κρύβονται σε αυτά. Με την ανάλυσή τους μπορούν να εντοπιστούν τάσεις, ανωμαλίες και μοτίβα στις τιμές των μετοχών ή άλλων χρηματοοικονομικών στοιχείων, επιτρέποντας καλύτερες προβλέψεις και επενδυτικές αποφάσεις.

1.7.3 Εκμετάλλευση των προβλέψεων

Χρησιμοποιώντας ενισχυτική μάθηση για χρηματοοικονομικές συναλλαγές, ορίζουμε τεχνικές ώστε να προβλέψουν τις τιμές των μετοχών, μια κατάλληλη συνάρτηση ανταμοιβής και ένα στόχο, οπότε δεν χρειάζεται να διαχειριστούμε ρητά για την εκμετάλλευση των προβλέψεων, αλλά με την διαδικασία δοκιμής και λάθους ο πράκτορας θα αναπτύξει στρατηγική και θα την προσαρμόζει ανάλογα ώστε να του αποφέρει το μέγιστο δυνατό κέρδος. Αυτή η «ελευθερία» είναι χαρακτηριστική στην ενισχυτική μάθηση και μας αποδεσμεύει από το να ορίσουμε όλες τις πιθανές ενέργειες κάτι που σε ένα δυναμικό περιβάλλον είναι μη ρεαλιστικό.

1.8 Θεωρητικό υπόβαθρο

1.8.1 Δομή ενισχυτικής μάθησης

Η δομή της ενισχυτικής μάθησης περιλαμβάνει βασικά συστατικά, όπως το περιβάλλον (environment), τον πράκτορα (agent), την πολιτική (policy), το σήμα ανταμοιβής (reward), καθώς και τη λειτουργεία τιμής (value function). Αυτά τα στοιχεία αλληλεπιδρούν για να δημιουργήσουν έναν βρόχο ανατροφοδότησης, μέσα από τον οποίο ο πράκτορας προσαρμόζεται και βελτιστοποιεί τη συμπεριφορά του προς τη μέγιστη δυνατή απόδοση. Μία από τις πλέον βασικές πηγές για την κατανόηση της ενισχυτικής μάθησης και της δομής της είναι το βιβλίο των (Sutton & Barto, 2018).

Περιβάλλον (Environment)

Στην ενισχυτική μάθηση το περιβάλλον είναι αυτό στο οποίο λειτουργεί ο πράκτορας. Είναι ένα μοντέλο ή μια προσομοίωση με την οποία ο πράκτορας αλληλοεπιδρά λαμβάνοντας δράσεις, ανταμοιβές ή ποινές και μεταβαίνοντας σε νέες καταστάσεις με βάση αυτές τις δράσεις. Το περιβάλλον παρέχει στον πράκτορα την αρχική του κατάσταση και την τρέχουσα κατάσταση μετά από κάθε ενέργεια. Καθορίζει επίσης την ανταμοιβή που σχετίζεται με κάθε ζεύγος κατάστασης-δράσης.

Τα περιβάλλοντα ενισχυτικής μάθησης είναι απαραίτητα για την εκπαίδευση των πρακτόρων. Παρέχουν τον απαραίτητο βρόχο ανατροφοδότησης, επιτρέποντας στον πράκτορα να μαθαίνει από τις ενέργειές του και να βελτιώνει την πολιτική του με την πάροδο του χρόνου. Η πολυπλοκότητα και η ποικιλομορφία αυτών των περιβαλλόντων μπορεί να επηρεάσει σημαντικά τη διαδικασία μάθησης και την απόδοση του εκπαιδευμένου πράκτορα.

Πράκτορας (Agent)

Ο πράκτορας στην ενισχυτική μάθηση, είναι το στοιχείο που λαμβάνει την απόφαση για το ποια ενέργεια θα γίνει.

Για να λάβει αυτή την απόφαση, ο πράκτορας μπορεί να χρησιμοποιήσει οποιαδήποτε παρατήρηση από το περιβάλλον και οποιουσδήποτε εσωτερικούς κανόνες διαθέτει. Αυτοί οι εσωτερικοί κανόνες μπορεί να είναι οτιδήποτε, αλλά τυπικά στην ενισχυτική μάθηση, αναμένει την τρέχουσα κατάσταση να παρέχεται από το περιβάλλον, ώστε αυτή η κατάσταση να έχει την ιδιότητα Markov, και στη συνέχεια επεξεργάζεται αυτή την κατάσταση χρησιμοποιώντας μια συνάρτηση πολιτικής που αποφασίζει ποια ενέργεια πρέπει να λάβει.

Στην ενισχυτική μάθηση φροντίζουμε για το χειρισμό ενός σήματος ανταμοιβής (που λαμβάνεται από το περιβάλλον) και τη βελτιστοποίηση του πράκτορα προς τη μεγιστοποίηση της αναμενόμενης ανταμοιβής όπως προαναφέρθηκε. Για να γίνει αυτό, ο πράκτορας θα διατηρεί κάποια δεδομένα τα οποία επηρεάζονται από τις ανταμοιβές που έλαβε στο παρελθόν και θα τα χρησιμοποιεί για να κατασκευάσει μια καλύτερη πολιτική.

Ένα ενδιαφέρον γεγονός σχετικά με τον ορισμό ενός πράκτορα είναι ότι το όριο πράκτορα/περιβάλλοντος δεν είναι πάντα τόσο ξεκάθαρο. Για παράδειγμα, για ένα ρομπότ,

ο πράκτορας δεν είναι συνήθως ολόκληρο το ρομπότ, αλλά το συγκεκριμένο πρόγραμμα που εκτελείται στην κεντρική μονάδα επεξεργασίας (CPU) του ρομπότ και λαμβάνει την απόφαση για τη δράση. Σε μερικές περιπτώσεις που η διάκριση μπορεί να μην έχει σημασία θα λέγαμε ότι «το ρομπότ κινεί το χέρι του για να επιτύχει το στόχο», ενώ με αυστηρότερους όρους ενισχυτικής μάθησης θα έπρεπε να πούμε ότι «ο πράκτορας που τρέχει στην CPU του ρομπότ δίνει εντολή στους κινητήρες του βραχίονα να κινηθούν για να επιτύχουν το στόχο».

Πολιτική (Policy)

Μια πολιτική είναι μια στρατηγική ή ένα σύνολο κανόνων που ακολουθεί ένας πράκτορας για να λαμβάνει αποφάσεις σε ένα περιβάλλον. Καθορίζει την αντιστοίχιση από τις καταστάσεις του κόσμου στις ενέργειες που πρέπει να κάνει ο πράκτορας. Ουσιαστικά, μια πολιτική καθοδηγεί τον πράκτορα σχετικά με το ποια ενέργεια να επιλέξει όταν συναντήσει μια συγκεκριμένη κατάσταση. Οι πολιτικές μπορεί να είναι απλές, περιλαμβάνοντας σταθερές ενέργειες για κάθε κατάσταση, ή σύνθετες, ενσωματώνοντας υπολογισμούς και μηχανισμούς μάθησης για τον προσδιορισμό των βέλτιστων ενεργειών.

Ανταμοιβή και τιμή (Reward and Value)

Η ανταμοιβή είναι ένα άμεσο σήμα ανατροφοδότησης που παρέχει το περιβάλλον στον πράκτορα μετά από κάθε ενέργεια που εκτελεί (Sutton & Barto, 2018). Αντιπροσωπεύει την επιτυχία ή αποτυχία μιας ενέργειας σε μια δεδομένη κατάσταση, επιβραβεύοντας ή τιμωρώντας, ανάλογα την περίπτωση. Η ανταμοιβή είναι ένα από τα βασικά στοιχεία που καθοδηγούν τη μάθηση του πράκτορας, καθώς του παρέχει πληροφορίες για το κατά πόσο οι ενέργειές του οδηγούν στο επιθυμητό αποτέλεσμα (Kaelbling et al., 1996).

- Άμεση ανταμοιβή: Συνήθως, η ανταμοιβή αφορά την άμεση απόδοση μιας δράσης σε μια κατάσταση και μπορεί να είναι είτε θετική (ανταμοιβή) είτε αρνητική (ποινή), ενθαρρύνοντας ή αποθαρρύνοντας συγκεκριμένες ενέργειες (Sutton & Barto, 2018).
- Στόχος: Σκοπός του πράκτορα είναι να μεγιστοποιήσει το συνολικό ποσό της ανταμοιβή που θα λάβει στο μέλλον, προσπαθώντας να βρει μια στρατηγική που να του επιτρέπει να λαμβάνει την μέγιστη δυνατή ανταμοιβή (Silver et al., 2021).

Η ανταμοιβή ενθαρρύνει τον πράκτορα να συγκρίνει και να επιλέγει ενέργειες που τον φέρνουν πιο κοντά στο στόχο του (Kaelbling et al., 1996).

Η τιμή είναι μια εκτίμηση της μελλοντικής ανταμοιβή και αποτελεί θεμελιώδη έννοια για την αξιολόγηση της μακροπρόθεσμης αποτελεσματικότητας μιας κατάστασης ή ενέργειας (Sutton & Barto, 2018). Αντί να επικεντρώνεται μόνο στην άμεση ανταμοιβή, η τιμή ενσωματώνει τις μελλοντικές ανταμοιβές, προσφέροντας στον πράκτορα μια εκτίμηση της συνολικής απόδοσης μιας κατάστασης ή μιας δράσης (Mnih et al., 2015).

- Τιμή κατάστασης (State Value): Είναι η αναμενόμενη συνολική ανταμοιβή που μπορεί να λάβει ο πράκτορας ξεκινώντας από μια συγκεκριμένη κατάσταση και ακολουθώντας την πολιτική του.
- Τιμή κατάστασης-δράσης (Action-Value ή Q-Value): Είναι η αναμενόμενη ανταμοιβή που θα λάβει ο πράκτορας για μια συγκεκριμένη δράση σε μια κατάσταση, λαμβάνοντας υπόψη τις μελλοντικές ενέργειες του πράκτορα.

Η έννοια της τιμής επιτρέπει στον πράκτορα να μαθαίνει στρατηγικές που ενσωματώνουν τόσο τις άμεσες όσο και τις μελλοντικές ανταμοιβές, προσαρμόζοντας τη συμπεριφορά του προς τη βέλτιστη απόδοση με την πάροδο του χρόνου (Silver et al., 2021).

1.9 Αλγόριθμοι ενισχυτικής μάθησης και τεχνητά νευρωνικά δίκτυα

1.9.1 Q-learning

Ο αλγόριθμος Q-learning ο οποίος δημιουργήθηκε από τον Chris Watkins to 1989 κατά τη διάρκεια του διδακτορικού του (Watkins, 1989), είναι ένας αλγόριθμος ενίσχυσης μάθησης χωρίς μοντέλα για να μάθει την αξία μιας ενέργειας σε μια συγκεκριμένη κατάσταση. Δεν απαιτεί ένα μοντέλο του περιβάλλοντος και μπορεί να χειριστεί προβλήματα με στοχαστικές μεταβάσεις και ανταμοιβές χωρίς να απαιτεί προσαρμογές (Li, 2023).

Για οποιαδήποτε πεπερασμένη διαδικασία απόφασης Markov(MDPs) όπου χρησιμοποιούνται για να μοντελοποιήσουν περιβάλλοντα, οι αποφάσεις λαμβάνονται σε διαδοχικά βήματα και κάθε κατάσταση και δράση έχει μία πιθανότητα μετάβασης σε μια νέα κατάσταση και μία αντίστοιχη ανταμοιβή, ο Q-learning βρίσκει μια βέλτιστη πολιτική με την έννοια της μεγιστοποίησης της αναμενόμενης τιμής της συνολικής ανταμοιβής σε

όλα τα διαδοχικά βήματα, ξεκινώντας από την τρέχουσα κατάσταση (Mello, n.d). Ο Q-learning μπορεί να προσδιορίσει μια βέλτιστη πολιτική επιλογής δράσης για κάθε δεδομένη πεπερασμένη διαδικασία απόφασης Markov, δεδομένου άπειρου χρόνου εξερεύνησης και μιας εν μέρει τυχαίας πολιτικής. Το "Q" αναφέρεται στη συνάρτηση που υπολογίζει ο αλγόριθμος τις αναμενόμενες ανταμοιβές για μια ενέργεια που πραγματοποιείται σε μια δεδομένη κατάσταση (Matiisen, 2015).

Ο αλγόριθμος Q-learning λειτουργεί σε ένα περιβάλλον που αποτελείται από καταστάσεις(states) και ενέργειες (actions). Ο πράκτορας (agent) επιλέγει ενέργειες για να μεταβεί από μία κατάσταση σε μία άλλη. Κάθε ενέργεια που εκτελεί ο πράκτορας αποφέρει μια ανταμοιβή (reward), η οποία όπως έχει αναφερθεί, μπορεί να είναι θετική ή αρνητική. Ο αλγόριθμος χρησιμοποιεί μία συνάρτηση Q (Q-function) ώστε να υπολογίζει την αναμενόμενη ανταμοιβή για μία ενέργεια σε μία συγκεκριμένη κατάσταση. Η συνάρτηση ενημερώνεται συνεχώς με βάση τις ανταμοιβές που λαμβάνει ο πράκτορας. Ο πράκτορας θα χρησιμοποιήσει έναν πίνακα Q (Q-table) για να κάνει την καλύτερη δυνατή ενέργεια με βάση την αναμενόμενη ανταμοιβή για κάθε κατάσταση στο περιβάλλον.

Ο πίνακας Q είναι μια δομή δεδομένων από το συνόλων ενεργειών και καταστάσεων και χρησιμοποιείται ο αλγόριθμο Q-learning για να ενημερώνονται οι τιμές στον πίνακα.

Κάθε κελί του πίνακα περιέχει μια τιμή Q (Q-values) που αντιπροσωπεύει την αναμενόμενη μελλοντική ανταμοιβή για την εκτέλεση μιας συγκεκριμένης δράσης σε μια συγκεκριμένη κατάσταση.

Οι τιμές Q (Q-values) είναι αριθμητικές τιμές που αποθηκεύονται στον Q-table και αντιπροσωπεύουν την ποιότητα μιας δράσης σε μια συγκεκριμένη κατάσταση. Αυτές οι τιμές ενημερώνονται συνεχώς καθώς ο αλγόριθμος μαθαίνει από την αλληλεπίδρασή του με το περιβάλλον. Η ενημέρωση των τιμών Q γίνεται με βάση την εξίσωση Bellman, η οποία λαμβάνει υπόψη την τρέχουσα ανταμοιβή και την αναμενόμενη μελλοντική ανταμοιβή.

Η συνάρτηση Q χρησιμοποιώντας την εξίσωση Bellman είναι η εξής:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[R(s,a) + \gamma \cdot max_a Q(s',a) - Q(s,a)] \qquad (1.1)$$

- (s): Η τρέχουσα κατάσταση.
- (a): Η ενέργεια που εκτελείται.

- $Q(s,a)$ είναι η τιμή Q για ένα δεδομένο ζεύγος κατάστασης-ενέργειας
- $R(s,a)$ είναι η άμεση ανταμοιβή για την ανάληψη δράσης a στην κατάσταση s
- γ: είναι ο παράγοντας έκπτωσης, που αντιπροσωπεύει τη σημασία των μελλοντικών ανταμοιβών (discount factor)
- $max_a Q(s',a)$ είναι η μέγιστη τιμή Q για την επόμενη κατάσταση s′ και όλες τις πιθανές ενέργειες
- (α): Ο ρυθμός μάθησης (learning rate).
- (s'): Η νέα κατάσταση μετά την εκτέλεση της ενέργειας.

```
Parameters: learning rate(a), reward discount factor(γ), exploration
rate(ε)
1)  Initialize Q(s, a) for all states s ∈ S and actions a ∈ A  to zero.
2)  Loop for each episode:
3)    Initialize the environment (set initial state s)
4)    Select an initial action (a) (optional: set to 0 or use an
    exploration policy).
5)    Loop for each time step:
6)    Take action a.
7)    Observe the reward (r) and the new state (s')
8)    Update the Q-table using the equation:
          Q(s, a) ← Q(s, a) + α [r + γ*maxa' Q(s', a') - Q(s, a)]
9)    Select the next action (a') using the ε-greedy policy.
10)   Set s ← s', a ← a' .
11) Until the episode ends (goal reached, maximum steps, or terminal
    state).
```

Ψευδοκώδικας 1: Q-learning

Όπως φαίνεται στον ψευδοκώδικα του Q-learning (βλέπε Ψευδοκώδικας 1), αρχικοποιείται ο πίνακας $Q(s,a)$ με μηδενικές ή τυχαίες τιμές. Ξεκινάει ένα επεισόδιο (δοκιμή), όπου ο πράκτορας ξεκινά από μία αρχική κατάσταση s. Επιλέγεται η πρώτη ενέργεια a (συνήθως χρησιμοποιείται μια πολιτική εξερεύνησης όπως ε-greedy). Στα βήματα 5-10 ο πράκτορας εκτελεί την ενέργεια α και λαμβάνει ανταμοιβή r. Παρατηρεί τη νέα κατάσταση s'. Ενημερώνει την Q-τιμή χρησιμοποιώντας τη βασική εξίσωση του Q-learning. Επιλέγει την επόμενη ενέργεια α' βάσει μιας στρατηγικής εξερεύνησης (π.χ ε-greedy), και μεταβαίνει στη νέα κατάσταση και συνεχίζει μέχρι να τελειώσει το επεισόδιο. Το επεισόδιο τελειώνει αν

φτάσουμε σε μια τελική κατάσταση (στόχος, μέγιστος αριθμός βημάτων ή τερματική κατάσταση).

1.9.2 Double Q-learning

Όπως αναφέρεται σε ορισμένα στοχαστικά περιβάλλοντα, ο αλγόριθμος Q-learning δεν αποδίδει πολύ καλά (Hasselt, 2010). Αυτή η κακή απόδοση οφείλεται σε μεγάλες υπερεκτιμήσεις των τιμών δράσης. Αυτές οι υπερεκτιμήσεις προκύπτουν επειδή ο αλγόριθμος Q-learning χρησιμοποιεί τη μέγιστη τιμή δράσης που έχει παρατηρήσει για να εκτιμήσει τη μελλοντική αναμενόμενη αξία, κάτι που συχνά οδηγεί σε μη ρεαλιστικές εκτιμήσεις της απόδοσης.

```
1) Initialize Q^A, Q^B, s
2) repeat
3)    Choose a, based on Q^A(s, ·) and Q^B(s, ·), observe r, s'
4)    Choose (e.g random) either UPDATE(A) or UPDATE(B)
5)    if UPDATE(A) then
6)       Define a* ← argmax_a Q^A(s', a)
7)       Q^A(s, a) ← Q^A(s, a) + α [r + γQ^B(s', a*) - Q^A(s, a)]
8)    else if UPDATE(B) then
9)       Define b* argmax_a Q^B(s', a)
10)      Q^B(s, a) ← Q^B(s, a) + α [r + γQ^A(s', b*) - Q^B(s, a)]
11)   end if
12)   s ← s'
13) until end
```

Ψευδοκώδικας 2: Double Q-learning

Ο Double Q-learning είναι ένας αλγόριθμος ενισχυτικής μάθησης εκτός πολιτικής που αποφεύγει τις υπερεκτιμήσεις των τιμών δράσης που παρατηρούνται στον Q-learning. Ο αλγόριθμος χρησιμοποιεί δύο συναρτήσεις Q, την Q^A και την Q^B, οι οποίες ενημερώνονται με διαφορετικά σύνολα εμπειριών. Κάθε συνάρτηση Q ενημερώνεται με μια τιμή από την άλλη συνάρτηση Q για την επόμενη κατάσταση. Όπως φαίνεται στον ψευδοκώδικα (βλέπε Ψευδοκώδικας 2), η δράση α^* στη γραμμή 6 είναι η μέγιστη τιμή δράσης στην κατάσταση s' δηλαδή $\alpha^* = arg\max_a Q^A(s', a)$, σύμφωνα με τη συνάρτηση τιμής Q^A. Ωστόσο, αντί να χρησιμοποιηθεί η τιμή $Q^A(s', a^*) = max_a Q^A(s', a)$ για την ενημέρωση του Q^A, όπως θα έκανε ο Q-learning, χρησιμοποιείται η τιμή $Q^B(s', a^*)$.Εφόσον το Q^B ενημερώθηκε για

το ίδιο πρόβλημα, αλλά με διαφορετικό σύνολο δειγμάτων εμπειρίας, αυτό μπορεί να ληφθεί υπόψη ως μια αμερόληπτη εκτίμηση για την αξία αυτής της ενέργειας. Μια παρόμοια ενημέρωση χρησιμοποιείται για το Q^B, χρησιμοποιώντας b^* και Q^A. Είναι σημαντικό και οι δύο συναρτήσεις Q να μαθαίνουν από ξεχωριστά σύνολα εμπειριών, αλλά μπορεί να επιλέγει μια ενέργεια για την εκτέλεση χρησιμοποιώντας και τις δύο συναρτήσεις τιμών.

Η δράση με τη μέγιστη τιμή σε μια κατάσταση επιλέγεται από την Q^A, αλλά η ενημέρωση γίνεται χρησιμοποιώντας την τιμή από την Q^B. Αυτός ο διαχωρισμός μειώνει την υπερεκτίμηση και επιτρέπει στον αλγόριθμο να συγκλίνει στην βέλτιστη πολιτική.

1.9.3 Τεχνητά νευρωνικά δίκτυα (Artificial neural networks)

Τα τεχνητά νευρωνικά δίκτυα αποτελούν ένα μεγάλο τομέα της τεχνητής νοημοσύνης. Είναι εμπνευσμένα από τη δομή και τη λειτουργεία των βιολογικών νευρωνικών δικτύων του εγκεφάλου.

Ένα τεχνητό νευρωνικό δίκτυο αποτελείται από υπολογιστικούς κόμβους ή μονάδες που ονομάζονται τεχνητοί νευρώνες ή νευρώνια, οι οποίοι μοντελοποιούν «χαλαρά» τους νευρώνες του εγκεφάλου. Είναι διασυνδεδεμένοι μεταξύ τους με ακμές, οι οποίες μοντελοποιούν τις συνάψεις στον εγκέφαλο.

Κάθε νευρώνας λαμβάνει σήματα (πραγματικοί αριθμοί), τους επεξεργάζεται και στέλνει ένα σήμα στους συνδεδεμένους νευρώνες. Η έξοδος του κάθε νευρώνα υπολογίζεται από κάποια γραμμική ή μη συνάρτηση του αθροίσματος των εισόδων του, που ονομάζεται συνάρτηση ενεργοποίησης. Η ισχύς του σήματος σε κάθε σύνδεση καθορίζεται από ένα βάρος, το οποίο προσαρμόζεται κατά τη διάρκεια της διαδικασίας μάθησης μέσω ενός αλγορίθμου βελτιστοποίησης, όπως η μέθοδος της οπισθοδιάδοσης τους λάθους (backpropagation). Αυτό επιτρέπει στο δίκτυο να μαθαίνει από τα δεδομένα και να βελτιώνει την απόδοσή του για τις συγκεκριμένες εργασίες που του έχουν ανατεθεί. Μέσω της διαδικασίας μάθησης τα βάρη τροποποιούνται έτσι ώστε το δίκτυο να μπορεί να ανταποκρίνεται καλύτερα στα δεδομένα που του παρέχονται, προσαρμόζοντας τη συμπεριφορά του με βάση τον τύπο του προβλήματος ή της εφαρμογής.

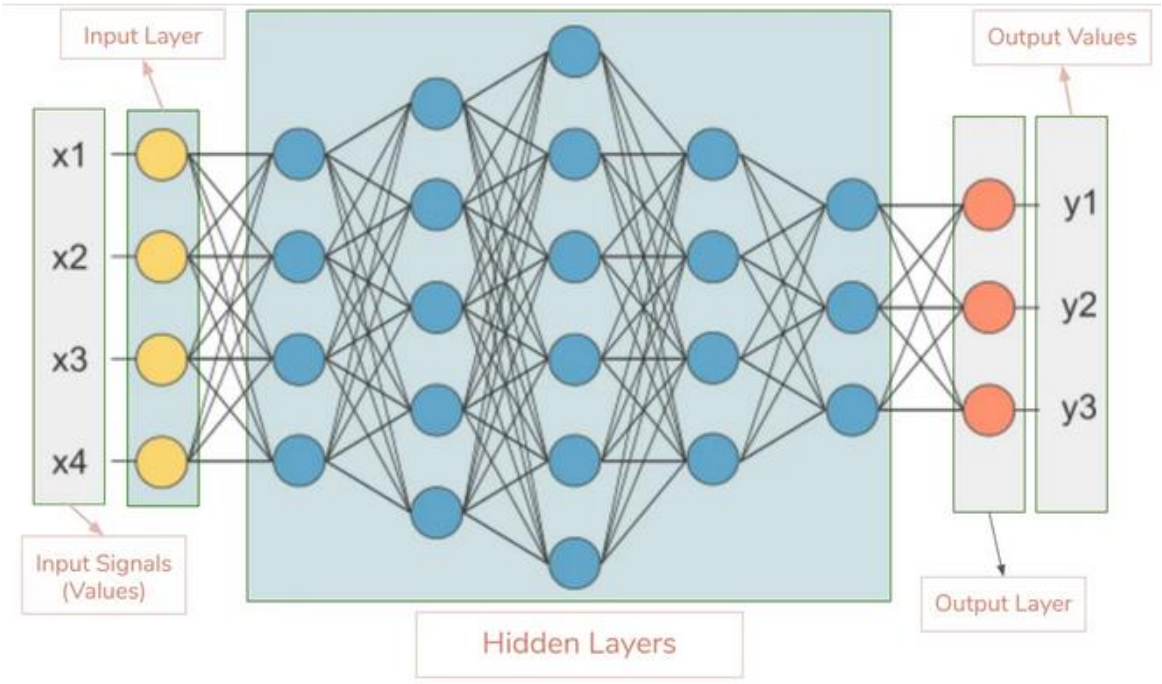
Ένα βασικό νευρωνικό δίκτυο έχει διασυνδεδεμένους τεχνητούς νευρώνες σε τρία επίπεδα

Επίπεδο εισόδου (Input layer): Πρόκειται για το πρώτο επίπεδο σε ένα νευρωνικό δίκτυο που λαμβάνει τα αρχικά δεδομένα (τιμές). Οι νευρώνες σε αυτό το επίπεδο δεν εφαρμόζουν καμία πράξη στα δεδομένα, δεν έχουν βάρη ή συναπτικά βάρη (biases) και τα μεταβιβάζουν στο επόμενο επίπεδο.

Κρυφό επίπεδο (Hidden layer): Μεταξύ των επιπέδων εισόδου και εξόδου, μπορεί να υπάρχουν ένα ή περισσότερα κρυφά επίπεδα. Αυτά τα επίπεδα εκτελούν πολύπλοκους υπολογισμούς στα δεδομένα που λαμβάνουν από το επίπεδο εισόδου. Κάθε νευρώνας σε ένα κρυφό επίπεδο λαμβάνει εισόδους από όλους τους νευρώνες στο προηγούμενο επίπεδο, εφαρμόζοντας ένα σταθμισμένο άθροισμα και ένα συναπτικό βάρος (bias). Το αποτέλεσμα περνά από μια συνάρτηση ενεργοποίησης, που καθορίζει την έξοδο του νευρώνα.

Επίπεδο εξόδου (Output layer): Είναι το τελευταίο επίπεδο του δικτύου και λαμβάνει είσοδο από το τελευταίο κρυφό επίπεδο. Οι νευρώνες σε αυτό το επίπεδο παράγουν την τελική έξοδο του δικτύου. Ο αριθμός των νευρώνων και το εύρος των τιμών εξόδου εξαρτώνται από το πρόβλημα που προσπαθούνε να λύσουν.

Ένα δίκτυο ονομάζεται συνήθως βαθύ νευρωνικό δίκτυο (βλέπε Εικόνα 3) αν έχει τουλάχιστον δύο κρυφά επίπεδα.



Εικόνα 3: Βασική διάταξη νευρωνικού δικτύου. RavenProtocol. (2017, December 4).

Τύποι νευρωνικών δικτύων

Υπάρχουν διάφοροι τύποι νευρωνικών δικτύων με το καθένα να είναι κατάλληλο για διάφορα προβλήματα. Μερικοί από τους πιο συχνά χρησιμοποιούμενους τύπους νευρωνικών δικτύων αναφέρονται παρακάτω.

Νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης (Feedforward neural network (FNN))

Τα νευρωνικά δίκτυα εμπρόσθιας τροφοδότησης χαρακτηρίζονται από την κατεύθυνση της ροής της πληροφορίας μεταξύ των επιπέδων του. Η ροή είναι μονοκατευθυντική, που σημαίνει ότι η πληροφορία ρέει μόνο προς μία κατεύθυνση, προς τα εμπρός, από τους κόμβους εισόδου, μέσω των κρυφών κόμβων (αν υπάρχουν) και προς τους κόμβους εξόδου, χωρίς κύκλους ή βρόχους. Είναι ο πρώτος τύπος τεχνητού νερωνικού δικτύου που εφευρέθηκε από τους Warren McCulloch and Walter Pitts το 1943. Μπορούν να χρησιμοποιηθούν για αναγνώριση προτύπων, εργασίες ταξινόμησης, ανάλυση παλινδρόμησης, αναγνώριση εικόνας, πρόβλεψη χρονοσειρών.

Συνελικτικό νευρωνικό δίκτυο (Convolutional neural network (CNN))

Τα συνελικτικά νευρωνικά δίκτυα ειδικεύονται στην επεξεργασία δεδομένων που έχουν τοπολογία που μοιάζει με πλέγμα, όπως μια εικόνα. Τα CNN λειτουργούν διαδοχικά, ακολουθώντας μια συγκεκριμένη σειρά από στάδια επεξεργασίας. Τα βασικά χαρακτηριστικά των CNN είναι τα συνελικτικά επίπεδα (Convolutional layers) τα οποία επίπεδα εφαρμόζουν φίλτρα στην είσοδο για να εξάγουν τοπικά χαρακτηριστικά. Το αποτέλεσμα της συνελικτικής λειτουργίας δημιουργεί χάρτες χαρακτηριστικών (feature maps) που δείχνουν τις ανιχνεύσιμες περιοχές. Τα συσσωρευτικά επίπεδα (Pooling layers), τα οποία μειώνουν τη διάσταση των δεδομένων διατηρώντας τα σημαντικά στοιχεία και αφαιρώντας τον θόρυβο και τέλος τα πλήρως συνδεδεμένα επίπεδα (fully connected layers) στα οποία κάθε νευρώνας συνδέεται με όλους τους νευρώνες του προηγούμενου επιπέδου. Οι εξαγόμενες πληροφορίες συνδυάζονται για να κάνουν την τελική πρόβλεψη ή

ταξινόμηση όπου αυτή η διαδοχική προσέγγιση τα καθιστά ιδανικά για αναγνώριση εικόνας, ανάλυση βίντεο και επεξεργασία φυσικής γλώσσας.

Αναδρομικό νευρωνικό δίκτυο (Recurrent neural network (RNN))

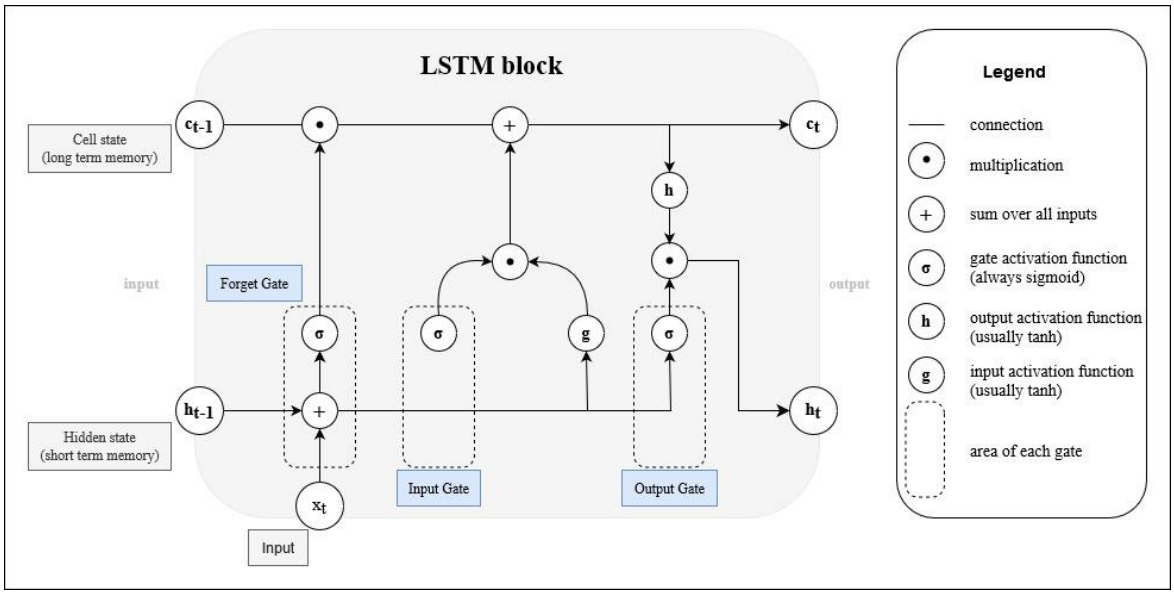
Τα αναδρομικά νευρωνικά δίκτυα χρησιμοποιούνται συνήθως για διαδοχική επεξεργασία δεδομένων. Επεξεργάζονται δεδομένα σε πολλαπλά χρονικά βήματα, καθιστώντας τα κατάλληλα για μοντελοποίηση και επεξεργασία χρονοσειρών. Το δομικό στοιχείο των RRNs είναι η αναδρομική μονάδα. Αυτή η μονάδα διατηρεί μια κρυφή κατάσταση, ουσιαστικά μια μορφή μνήμης, η οποία ενημερώνεται σε κάθε χρονικό βήμα με βάση την τρέχουσα είσοδο και την προηγούμενη κρυφή κατάσταση. Αυτός ο βρόχος ανατροφοδότησης επιτρέπει στο δίκτυο να μαθαίνει από τις προηγούμενες εισόδους και να ενσωματώνει τη γνώση αυτή στην τρέχουσα επεξεργασία του.

Τα πρώτα μοντέλα RNNs αντιμετώπιζαν αρκετά προβλήματα που περιόριζαν την απόδοσή τους, όπως η εξαφάνιση/έκρηξη κλίσης (Vanishing/Exploding gradients) και η περιορισμένη ικανότητα διαχείρισης μακροπρόθεσμων εξαρτήσεων. Η εξαφάνιση κλίσης είναι ένα φαινόμενο που συμβαίνει κατά την εκπαίδευση των νευρωνικών δικτύων, ιδιαίτερα των RNNs, όπου κατά τη διάρκεια της πίσω διάδοσης, οι κλίσεις (gradients) υπολογίζονται για να προσαρμόσουν τα βάρη. Καθώς οι κλίσεις περνούν προς τα πίσω μέσω των επαναληπτικών επιπέδων, πολλαπλασιάζονται με αυτά τα μικρά νούμερα με αποτέλεσμα τα βάρη των πρώτων επιπέδων να μην ενημερώνονται σημαντικά, επειδή οι κλίσεις γίνονται πολύ μικρές. Αυτό οδηγεί το μοντέλο να μην μπορεί να μάθει μακροπρόθεσμες εξαρτήσεις στα δεδομένα. Από την άλλη η έκρηξη κλίσης είναι ένα φαινόμενο όπου οι κλίσεις των συναρτήσεων απώλειας αυξάνονται υπερβολικά κατά τη διάρκεια της πίσω διάδοσης, τα βάρη του νευρωνικού δικτύου αυξάνονται υπερβολικά, καθιστώντας την εκπαίδευση του μοντέλου ασταθή. Το μοντέλο μπορεί να «εκραγεί», δηλαδή να παράγει μεγάλες και λανθασμένες τιμές εξόδου.

Έχουν προταθεί διάφορες λύσεις για να λύσουν τα παραπάνω προβλήματα μία από τις οποίες είναι τα LSTM δίκτυα.

Μακροχρόνια βραχυπρόθεσμη μνήμη (LSTM (Long Short-Term Memory))

Η μακροχρόνια βραχυπρόθεσμη μνήμη είναι ένας τύπος RNN. Σε αντίθεση με ένα παραδοσιακό RNN, το οποίο έχει μια απλή δομή εισόδου, κρυφής κατάστασης και εξόδου, ένα LSTM έχει μια πιο σύνθετη δομή με πρόσθετα κύτταρα μνήμης και πύλες που του επιτρέπουν να θυμάται ή να ξεχνά επιλεκτικά πληροφορίες από προηγούμενα χρονικά βήματα (βλέπε Εικόνα 4).



Εικόνα 4: Αρχιτεκτονική ενός LSTM. (Van Houdt et al., 2020)

Η αρχιτεκτονική ενός LSTM μπορεί να απεικονιστεί ως μια σειρά επαναλαμβανόμενων μπλοκ ή κελιών, καθένα από τα οποία περιέχει ένα σύνολο διασυνδεμένων κόμβων. Μια υψηλού επιπέδου ανάλυση αρχιτεκτονικής είναι:

- Είσοδος (Input layer): Κάθε χρονική στιγμή, το LSTM λαμβάνει ένα διάνυσμα εισόδου x_t που αντιπροσωπεύει την τρέχουσα παρατήρηση ή σύμβολο στην ακολουθία δεδομένων.
- Κρυφή κατάσταση (Hidden state): Το LSTM διατηρεί ένα διάνυσμα κρυφής κατάστασης h_t, το οποίο λειτουργεί ως τρέχουσα «μνήμη» του δικτύου. Στην αρχή της ακολουθίας, η κρυφή κατάσταση αρχικοποιείται σε μηδενικά.
- Κατάσταση κυττάρων (Cell state): Το LSTM διατηρεί ένα διάνυσμα κατάστασης κυττάρων c_t, που είναι υπεύθυνο για την αποθήκευση και επεξεργασία

μακροπρόθεσμων πληροφοριών κατά τη διάρκεια της ακολουθίας, ενσωματώνοντας δεδομένα από προηγούμενες χρονικές στιγμές. Η κατάσταση κυττάρων αρχικοποιείται σε μηδενικά στην αρχή της ακολουθίας.

- Πύλες Ελέγχου (Gates): Το LSTM χρησιμοποιεί τρεις τύπους πυλών για τον έλεγχο ροής των πληροφοριών.
- Έξοδος (Output): Σε κάθε χρονικό βήμα, το LSTM παράγει ένα διάνυσμα εξόδου συνήθως ίδιο με το h_t, το οποίο αντιπροσωπεύει την πρόβλεψη ή την κωδικοποίηση της τρέχουσας εισόδου από το δίκτυο.

Πύλες ελέγχου:

- Forget Gate: Αυτή η πύλη δέχεται την προηγούμενη κρυφή κατάσταση h_{t-1} και την τρέχουσα είσοδο x_t και εξάγει ένα διάνυσμα τιμών μεταξύ του 0 και 1. Αυτές οι τιμές καθορίζουν πόσο από την προηγούμενη κατάσταση του κελιού θα διαγραφεί και πόσο θα διατηρηθεί. Έτσι το LSTM μπορεί να επιλέξει ποιες πληροφορίες από το προηγούμενο χρονικό βήμα να «ξεχάσει» ή να «θυμάται».
- Input Gate: Αυτή η πύλη δέχεται την προηγούμενη κρυφή κατάσταση h_{t-1} και την τρέχουσα είσοδο x_t και εξάγει ένα διάνυσμα τιμών μεταξύ του 0 και 1. Αυτές οι τιμές καθορίζουν πόσο από την τρέχουσα είσοδο θα προστεθεί στην κατάσταση του κελιού. Η πύλη εισόδου επιτρέπει στο LSTM να «προσθέτει» ή να «απορρίπτει» επιλεκτικά νέες πληροφορίες στην κατάσταση του κελιού.
- Output Gate: Αυτή η πύλη δέχεται την προηγούμενη κρυφή κατάσταση h_{t-1} και την τρέχουσα είσοδο x_t και την τρέχουσα κατάσταση του κελιού c_t και εξάγει ένα διάνυσμα τιμών μεταξύ 0 και 1. Αυτές οι τιμές καθορίζουν πόσο από την τρέχουσα κατάσταση του κελιού θα εξαχθεί ως η τρέχουσα κρυφή κατάσταση h_t. Η πύλη εξόδου επιτρέπει στο LSTM να «εστιάζει» ή να «αγνοεί» επιλεκτικά ορισμένα τμήματα της κατάστασης του κελιού κατά τον υπολογισμό της εξόδου.

Ο συνδυασμός της κατάστασης των κελιών, της κρυφής κατάστασης και των πυλών επιτρέπει στο LSTM να «θυμάται» ή να «ξεχνά» επιλεκτικά πληροφορίες με την πάροδο του χρόνου. Αυτές οι δυνατότητες καθιστούν τα LSTM ιδανική επιλογή για χρηματοοικονομικές εφαρμογές, όπως η πρόβλεψη τιμών μετοχών, όπου είναι απαραίτητο

να εντοπίζονται και να μοντελοποιούνται μακροχρόνιες τάσεις και μοτίβα σε ιστορικά δεδομένα.

1.10 Αλγόριθμοι ενισχυτικής μάθησης με τεχνητά νευρωνικά δίκτυα

1.10.1 Deep Q-Network

Το Deep Q-Network (DQN) είναι ένα μοντέλο βαθιάς μάθησης που χρησιμοποιείται για την εκμάθηση πολιτικών ελέγχου απευθείας από υψηλής διάστασης αισθητηριακές εισόδους, όπως τα βίντεο. Οι πιο επιτυχημένες προσεγγίσεις εκπαιδεύονται απευθείας από ακατέργαστα δεδομένα, χρησιμοποιώντας ελαφριές ενημερώσεις που βασίζονται σε στοχαστική κλίση. Το DQN χρησιμοποιεί βαθύ νευρωνικό δίκτυο που εκπαιδεύεται με μια παραλλαγή του αλγορίθμου Q-learning (Mnih et al., 2015).

Αναπτύχθηκε από την ομάδα DeepMind της Google το 2013 και δημοσιεύθηκε το 2015. Η καινοτομία του ήταν η χρήση ενός βαθέως νευρωνικού δικτύου για την εκμάθηση πολιτικών ελέγχου απευθείας από ακατέργαστα δεδομένα, κάτι που δεν είχε επιτευχθεί προηγουμένως. Μπορεί να εφαρμοστεί για παράδειγμα, σε παιχνίδια όπως το Atari 2600, επιτυγχάνοντας καλύτερα αποτελέσματα από τον άνθρωπο σε μερικά από αυτά (Mnih et al., 2013). Επιπλέον, μπορεί να χρησιμοποιηθεί σε τομείς όπως η ρομποτική, η αυτόνομη οδήγηση καθώς το DQN μπορεί να μάθει αυτόνομα από την αλληλεπίδραση με το περιβάλλον, αλλά και στην διαχείριση ενέργειας . Για παράδειγμα, έχει χρησιμοποιηθεί για τη βελτιστοποίηση της κατανάλωσης ενέργειας σε έξυπνα δίκτυα (smart grids) (Glavic & Saric, 2017).

Το DQN εισήγαγε δύο βασικές τεχνικές για τη βελτίωση της σταθερότητας και της απόδοσης. Την αναπαραγωγή εμπειριών (experience replay) και το δίκτυο στόχου (target network). Η αναπαραγωγή εμπειριών αποθηκεύει τις εμπειρίες του πράκτορα σε μια μνήμη replay, ενώ το δίκτυο στόχου χρησιμοποιείται για την ενημέρωση των τιμών Q, μειώνοντας την πιθανότητα αστάθειας κατά την εκπαίδευση.

Μερικά πλεονεκτήματα του DQN είναι η αντιμετώπιση προκλήσεων όπως τα συσχετισμένα δεδομένα και οι μη σταθερές κατανομές δεδομένων μέσω της χρήσης του μηχανισμού αναπαραγωγής εμπειριών (experience replay). Αυτός ο μηχανισμός αποθηκεύει τις εμπειρίες του πράκτορα (καταστάσεις, ενέργειες, ανταμοιβές, επόμενες καταστάσεις) σε

μια μνήμη replay, επιτρέποντας την τυχαία δειγματοληψία κατά την εκπαίδευση και βελτιώνοντας τη σταθερότητα και την αποδοτικότητα του αλγορίθμου.

Παρά τα πλεονεκτήματα του, το DQN αντιμετωπίζει προκλήσεις όπως η ανάγκη για μεγάλο αριθμό δεδομένων και η δυσκολία στην εκμάθηση σε περιβάλλοντα με υψηλή διαστατικότητα (η κατάσταση του περιβάλλοντος περιγράφεται από πολλά χαρακτηριστικά). Επιπλέον, η εκπαίδευση του DQN μπορεί να είναι χρονοβόρα και απαιτεί σημαντικούς υπολογιστικούς πόρους.

1.10.2 Double Deep Q-Network

Στον DQN, το ίδιο νευρωνικό δίκτυο χρησιμοποιείται τόσο για την επιλογή της καλύτερης δράσης όσο και για την αξιολόγηση της αξίας αυτής της δράσης. Αυτό σημαίνει ότι το δίκτυο επιλέγει την δράση με την υψηλότερη εκτιμώμενη τιμή Q και ταυτόχρονα χρησιμοποιεί αυτή την εκτιμώμενη τιμή για να ενημερώσει τις παραμέτρους του. Αυτή η διαδικασία μπορεί να οδηγήσει σε υπερεκτίμηση των τιμών Q. Για την αντιμετώπιση του προβλήματος προτάθηκε ο Double DQN (Hado et al., 2015).

Ο Double Deep Q-Network (DDQN) χρησιμοποιεί δύο νευρωνικά δίκτυα. Το ένα δίκτυο χρησιμοποιείται για την επιλογή της δράσης (online network) και το άλλο για την αξιολόγηση της δράσης (target network). Το δίκτυο επιλογής δράσης επιλέγει την καλύτερη δράση για μια δεδομένη κατάσταση, και στη συνέχεια το δίκτυο αξιολόγησης δράσης χρησιμοποιείται για να υπολογίσει την τιμή Q για αυτήν τη δράση. Αυτός ο διαχωρισμός μειώνει την πιθανότητα υπερεκτίμησης των τιμών Q, μειώνοντας έτσι την πιθανότητα να επιλεγούν δράσεις που φαίνονται καλές αλλά δεν είναι πραγματικά οι βέλτιστες, οδηγώντας σε πιο σταθερή και αξιόπιστη εκπαίδευση.

1.10.3 Dueling Deep Q-Network (Dueling DQN)

Το Dueling Deep Q-Network αποτελεί μια σημαντική βελτίωση στην ενισχυτική μάθηση σε σύγκριση με τα κλασικά Deep Q-Networks. Εισήχθη για να βελτιώσει την απόδοση των πρακτόρων ενισχυτικής μάθησης, ειδικά σε περιβάλλοντα όπου πολλές ενέργειες έχουν παρόμοια αποτελέσματα.(Wang et al., 2015). Βελτιώνει την «αντιληπτική» ικανότητα του

πράκτορα, μαθαίνοντάς του να ξεχωρίζει αν μια κατάσταση είναι γενικά καλή ή κακή, και στη συνέχεια να αξιολογεί ποια δράση προσφέρει το μεγαλύτερο πλεονέκτημα για αυτή την κατάσταση.

Αρχιτεκτονική μονομαχίας (Dueling Architecture)

Στην αρχιτεκτονική μονομαχίας η βασική ιδέα είναι να διαχωριστεί η εκτίμηση της αξίας της κατάστασης από την εκτίμηση των πλεονεκτημάτων των ενεργειών. Ο διαχωρισμός της αξίας της κατάστασης και των πλεονεκτημάτων των ενεργειών επιτρέπει στο δίκτυο να μαθαίνει πιο αποδοτικά, καθώς μπορεί να εστιάσει ξεχωριστά σε κάθε πτυχή.

Η ροή αξίας (Value Stream): Εκτιμά την αξία της τρέχουσας κατάστασης. Η αξία της κατάστασης αντιπροσωπεύει πόσο καλή είναι η κατάσταση ανεξάρτητα από τις ενέργειες που μπορούν να ληφθούν. Η ροή αυτή αποτελείται από ένα ή περισσότερα πλήρως συνδεδεμένα επίπεδα που καταλήγουν σε έναν μοναδικό νευρώνα που παράγει την εκτίμηση της αξίας $V(s)$.

Η ροή πλεονεκτήματος (Advantage Stream): Εκτιμά το πλεονέκτημα κάθε δράσης ($A(s,a)$) σε σχέση με την τρέχουσα κατάσταση. Το πλεονέκτημα μιας ενέργειας δείχνει αν είναι καλύτερη ή χειρότερη μια ενέργεια σε σχέση με τη μέση απόδοση των ενεργειών στην ίδια κατάσταση. Η ροή αυτή αποτελείται από ένα ή περισσότερα πλήρως συνδεδεμένα επίπεδα που καταλήγουν σε τόσους νευρώνες όσες είναι οι πιθανές ενέργειες.

Αυτά τα δύο μέρη συνδυάζονται για να υπολογίσουν την τελική τιμή Q για κάθε ενέργεια ($Q(s,a)$).

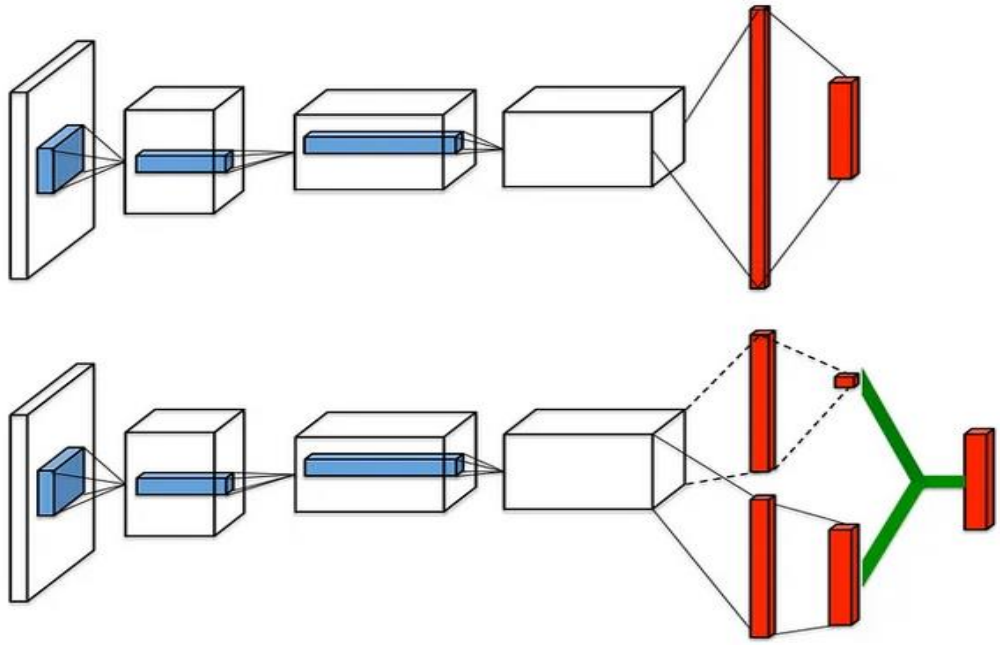
$$Q(s,a) = V(s) + \left(A(s,a) - \frac{1}{|A|}\sum_{a'} A(s,a') \right) \quad (1.2)$$

$Q(s,a)$: Αυτή είναι η συνάρτηση Q, η οποία αντιπροσωπεύει την αναμενόμενη συνολική ανταμοιβή που θα λάβει ένας πράκτορας αν ξεκινήσει από την κατάσταση (s), εκτελέσει την ενέργεια (a), και στη συνέχεια ακολουθήσει την πολιτική του.

$V(s)$: Αυτή είναι η συνάρτηση αξίας της κατάστασης (s). Αντιπροσωπεύει την αναμενόμενη συνολική ανταμοιβή που θα λάβει ένας πράκτορας αν ξεκινήσει από την κατάσταση (s) και ακολουθήσει την πολιτική του.

$A(s, a)$: Αυτή είναι η συνάρτηση πλεονεκτήματος για την ενέργεια (a) στην κατάσταση (s). Αντιπροσωπεύει πόσο καλή είναι η δράση (a) σε σχέση με τις άλλες δράσεις που θα μπορούσε να εκτελέσει ο πράκτορας στην κατάσταση (s).

$\frac{1}{|A|}\sum_{a'} A(s, a')$: Αυτός ο όρος αντιπροσωπεύει τον μέσο όρο του πλεονεκτήματος όλων των πιθανών ενεργειών στην κατάσταση (s). Εδώ, $(|A|)$ είναι ο αριθμός των πιθανών ενεργειών και $(A(s, a'))$ είναι το πλεονέκτημα της ενέργειας (a') στην κατάσταση (s).



Εικόνα 5: DQN VS Dueling Network Architectures. Andey, H. (2022, June 15).

Η αρχιτεκτονική μονομαχίας (βλέπε Εικόνα 5) έχει χρησιμοποιηθεί σε διάφορες εφαρμογές όπως για παιχνίδια Atari 2600 (Wang et al., 2015), αλλά και σε συναλλαγές κρυπτονομισμάτων ή μετοχών όπως αναφέρεται στο άρθρο Deep Reinforcement Learning for Trading Cryptocurrencies της σελίδα medium όπου μπορεί να χρησιμοποιηθεί για την ανάπτυξη πρακτόρων που λαμβάνουν αποφάσεις σε πραγματικό χρόνο για την αγορά και πώληση κρυπτονομισμάτων ή μετοχών.

Η συγκεκριμένη αρχιτεκτονική ενισχύει τη δυνατότητα των πρακτόρων να εκμεταλλεύονται τις ευκαιρίες στις αγορές με μεγαλύτερη ακρίβεια, συμβάλλοντας στην ανάπτυξη στρατηγικών που μεγιστοποιούν την απόδοση και μειώνουν τον κίνδυνο, ειδικά

σε περιβάλλοντα με έντονες διακυμάνσεις τιμών (Wang et al., 2015). Οι πράκτορες έχουν δείξει πως μπορούν να εφαρμόσουν τη γνώση τους σε διαφορετικά περιβάλλοντα, προσαρμόζοντας συνεχώς τις στρατηγικές τους με βάση την εμπειρία που αποκτούν.

1.10.4 Dueling Double Deep Q-Network (Dueling DDQN)

Το Dueling Double Deep Q-Network (DDDQN) αποτελεί μια εξέλιξη της αρχιτεκτονικής Dueling DQN, ενσωματώνοντας τον μηχανισμό Double Q-learning για τη μείωση του φαινομένου της υπερεκτίμησης τιμών (overestimation bias). Ενώ το Dueling DQN εισάγει τη διάκριση μεταξύ της ροής αξίας (Value Stream) και της ροής πλεονεκτήματος (Advantage Stream) για καλύτερη απόδοση εκτίμησης της συνάρτησης Q, το DDDQN βελτιώνει περαιτέρω τη διαδικασία μάθησης διαχωρίζοντας την επιλογή της δράσης από την αξιολόγησή της κατά τον υπολογισμό της τιμής στόχου.

Συγκεκριμένα, για τον υπολογισμό της τιμής στόχου y (ή $Q(s,a)^*$) που χρησιμοποιείται για την εκπαίδευση του online δικτύου, ακολουθούνται τα εξής βήματα:

1. Επιλογή δράσης από το online δίκτυο: Το Online δίκτυο (Q_{online}) χρησιμοποιείται πρώτα για να επιλέξει την ενέργεια ($a*$) η οποία εκτιμάται ότι θα αποφέρει την υψηλότερη Q-τιμή στην επόμενη κατάσταση s':

$$a^* = \underset{a'}{\operatorname{argmax}}\, Q_{online}(s', a') \tag{1.3}$$

2. Αξιολόγηση της επιλεγμένης δράσης από το Target δίκτυο: Στη συνέχεια, το Target δίκτυο (Q_{target}) χρησιμοποιείται για να αξιολογήσει την Q-τιμή αυτής της επιλεγμένης ενέργειας a^* στην επόμενη κατάσταση s'.
3. Υπολογισμός της τιμής-στόχου: Η τιμή στόχου y διαμορφώνεται ως εξής:

$$y = r + \gamma \cdot Q_{target}(s', a^*) = r + \gamma \cdot Q_{target}\left(s', \underset{a'}{\operatorname{argmax}} Q_{online}(s', a')\right) \tag{1.4}$$

Όπου:

r: η άμεση ανταμοιβή που λαμβάνει ο πράκτορας μετά την ενέργεια a στην κατάσταση s.

γ (gamma): Ο παράγοντας έκπτωσης (discount factor), που καθορίζει πόσο σημαντικές είναι οι μελλοντικές ανταμοιβές.

α^*: Η ενέργεια που επιλέχθηκε από το Online δίκτυο ως η βέλτιστη στην κατάσταση s'.

argmax: Η δράση a' που σύμφωνα με το online δίκτυο, αναμένεται να δώσει τη a'
μέγιστη Q- τιμή στην επόμενη κατάσταση s'.

$Q_{online}(s', a')$: Η εκτιμώμενη Q-τιμή από το online δίκτυο για την επόμενη κατάσταση s' και μια πιθανή ενέργεια a'.

$Q_{target}(s', \dots)$: Η εκτίμηση της αξίας από το target δίκτυο για τη δράση που επέλεξε το online δίκτυο.

Αυτή η προσέγγιση, όπου το δίκτυο που επιλέγει την καλύτερη επόμενη δράση (Q_{online}) είναι διαφορετικό από το δίκτυο που αξιολογεί την αξία αυτής της δράσης (Q_{target}), είναι ο πυρήνας του μηχανισμού Double Q-learning. Βοηθά στη μείωση της τάσης για υπερεκτίμηση των Q-τιμών, οδηγώντας σε πιο σταθερή και αξιόπιστη μάθηση. Αυτό καθιστά το DDDQN ιδιαίτερα κατάλληλο για πολύπλοκα και αβέβαια περιβάλλοντα, όπως οι χρηματοοικονομικές αγορές.

1.11 Συμπεράσματα

Σε αυτό το κεφάλαιο έγινε μια προσπάθεια εισαγωγής στην ενισχυτική μάθηση και την εφαρμογή της σε διάφορους τομείς, που την καθιστούν ιδανική με εξαιρετικά αποτελέσματα, όπως ταχύτητα και λήψη αποφάσεων σε περιβάλλοντα με υψηλή αβεβαιότητα, όπως οι χρηματοπιστωτικές αγορές. Αλγόριθμοι όπως ο Dueling Double Deep Q-Network θεωρείται ιδανική επιλογή για χρηματοπιστωτικές αγορές λόγω της βελτιωμένης ακρίβειας στην λήψη αποφάσεων που προσφέρει μέσω της αρχιτεκτονικής μονομαχίας και βοηθούν στη βελτιστοποίηση στρατηγικών συναλλαγών όπου μπορούν να προσαρμοστούν σε δυναμικά περιβάλλοντα χρηματοπιστωτικών αγορών, όπως θα παρουσιαστεί σε μετέπειτα κεφάλαιο.

Ωστόσο παρά τα θετικά της ενισχυτικής μάθησης και συγκεκριμένα του Dueling DDQN παρουσιάζει ορισμένα μειονεκτήματα. Ένα από τα βασικά είναι η ανάγκη για πολλούς

υπολογιστικούς πόρους και μεγάλο όγκο δεδομένων για εκπαίδευση, κάτι που δεν είναι πάντοτε εφικτό. Επιπλέον η ενισχυτική μάθηση είναι απαιτητική όσον αφορά τη ρύθμιση παραμέτρων και τη δημιουργία κατάλληλων συναρτήσεων επιβράβευσης, ώστε το μοντέλο να μπορεί να διαχειρίζεται αποτελεσματικά σύνθετες συνθήκες αγοράς.

Για την αντιμετώπιση αυτών των ζητημάτων, οι σύγχρονες τεχνικές, όπως η βελτιστοποίηση αλγορίθμων και η χρήση δεδομένων σε πραγματικό χρόνο, μπορούν να βελτιώσουν την απόδοση των μοντέλων ενισχυτικής μάθησης.

Συνεχίζοντας από τη θεωρητική βάση, στο επόμενο κεφάλαιο θα γίνει εφαρμογή αυτών των αλγορίθμων στο σύστημα αυτόματης αγοραπωλησίας μετοχών.

2 Κεφάλαιο 2

2.1 Βιβλιογραφική ανασκόπηση

Η βιβλιογραφική ανασκόπηση εστιάζει σε τεχνικές και μεθόδους βαθιάς μάθησης και ενισχυτικής μάθησης, με κύρια εφαρμογή στον τομέα των χρηματοοικονομικών συναλλαγών. Οι πηγές που μελετήθηκαν καλύπτουν θεωρητικό υπόβαθρο, ανάλυση αλγορίθμων και τελικών αποτελεσμάτων πάνω σε πρακτικές εφαρμογές όπου μπορούν να αξιοποιηθούν στον πραγματικό κόσμο.

Η βαθιά μάθηση όπως έχει ήδη αναφερθεί, αναφέρεται σε πολυεπίπεδα νευρωνικά δίκτυα, ώστε να εξαχθούν σύνθετα πρότυπα και σχέσεις από τα δεδομένα. Το βιβλίο των (Goodfellow et al., 2016) είναι θεμελιώδης για την κατανόηση της βαθιά μάθησης, αναλύοντας εκτενώς τις αρχές και τις τεχνικές της. Ιδιαίτερο ενδιαφέρον έχει η εργασία των (Z. Li et al., 2023), όπου αναλύονται η χρήση των LSTM δικτύων σε συναλλαγές μετοχών και την εξειδίκευσή τους στην αναγνώριση και διατήρηση μακροπρόθεσμων εξειδικευμένων παραλλαγών σε δεδομένα, όπως επίσης την χρησιμότητα του Adam optimizer και της συνάρτησης απώλειας (MSELoss) στη διαδικασία εκπαίδευσης του μοντέλου τους αναδεικνύοντας την σημασία αυτών των εργαλείων για την ακρίβεια των προβλέψεων και την σταθερότητα των αποτελεσμάτων.

Η εργασία των (Sami et al., 2023) δείχνει πόσο σημαντική είναι η επιλογή της συνάρτηση ενεργοποίησης στα LSTM δίκτυα και παρουσιάζει τα αποτελέσματα διαφορετικών συναρτήσεων ενεργοποίησης.

Η ενισχυτική μάθηση όπου επικεντρωνόμαστε στην διπλωματική εργασία είναι μια μέθοδος εκμάθησης ενός πράκτορα μέσω της αλληλεπίδρασής του με το περιβάλλον, με στόχο να μεγιστοποιήσει το κέρδος του όπως έχει ήδη αναφερθεί. Το βιβλίο των (Sutton & Barto, 1998) είναι θεμελιώδες έργο στον τομέα της ενισχυτικής μάθησης όπου γίνεται μια εκτενή ανάλυση, καλύπτοντας βασικές και προχωρημένες τεχνικές. Για την εφαρμογή της σε χρηματοοικονομικές εφαρμογές που αναδεικνύουν την αναγκαιότητα της, υπάρχουν πολλές δημοσιεύσεις. Στο έργο των (Li et al., 2019) γίνεται χρήση και τεχνικών δεικτών όπου φαίνεται η αναγκαιότητα τους για την πρόβλεψη των τιμών μετοχών και την κατανόηση

των αγορών. Γίνεται επίσης αναφορά στην replay memory και την σημαντικότητα της χρήσης της στην εκπαίδευση του μοντέλου.

Η εξέλιξη των DQN αρχιτεκτονικών έχει οδηγήσει στην αύξηση της χρήσης τους λόγω των επιδόσεών τους. Το έργο του (Sewak, 2019) δείχνει την εξέλιξη και τα οφέλη των DQN αρχιτεκτονικών. Η εργασία των (Wang et al., 2015b) κάνει εισαγωγή στον Dueling Q-Network και παρουσιάζει τα θετικά αποτελέσματα της αρχιτεκτονικής και την βελτιωμένη απόδοση έναντι των άλλων DQN αρχιτεκτονικών.

Σημαντική πηγή ήταν επίσης το άρθρο της σελίδας Medium του (Andey, 2022) όπου βρήκαμε μία προσέγγιση διορθώνοντας σημεία του κώδικα που υπολειτουργούσαν και λόγω ότι το άρθρο ήταν βασισμένο για συναλλαγές κρυπτονομισμάτων, έγιναν οι απαραίτητες αλλαγές για συναλλαγές μετοχών βάσει του δείκτη S&P 500.

2.2 Σχεδιασμός αλγορίθμου ενισχυτικής μάθησης

Στην μεγάλη επιτυχία της τεχνολογικής προόδου πάνω στην αυτοματοποίηση συναλλαγών, ευθύνη έχει η ενισχυτική μάθηση, όπου μία από τις πλέον σύγχρονες και αποτελεσματικές μεθόδους ενισχυτικής μάθησης που έχει εφαρμοστεί στις χρηματοπιστωτικές συναλλαγές είναι ο αλγόριθμος Dueling Deep Q-Network (Dueling DQN) (Y. Li et al., 2019b) όπως επίσης η εξέλιξή του ο Dueling Double DQN που αναφέρθηκε σε προηγούμενο κεφάλαιο.

Μέσω της αρχιτεκτονικής μονομαχίας, ο Dueling DDQN επιτρέπει στον πράκτορα να διακρίνει ποιες ενέργειες έχουν την μεγαλύτερη αξία σε κάθε δεδομένη στιγμή, επιτρέποντάς του να επιλέγει την πιο αποδοτική στρατηγική.

Αυτό το κεφάλαιο εστιάζει στην ανάπτυξη μοντέλου για την αυτόματη διαχείριση συναλλαγών. Με την ενσωμάτωση πραγματικών δεδομένων του δείκτη S&P 500, το μοντέλο θα επιδιώξει να ελαχιστοποιήσει τον κίνδυνο και να μεγιστοποιήσει τα κέρδη, δημιουργώντας μια ολοκληρωμένη προσέγγιση στις αλγοριθμικές συναλλαγές.

Ο σχεδιασμός ενός αλγορίθμου ενισχυτικής μάθησης για συναλλαγές μετοχών, είναι μια σύνθετη διαδικασία που απαιτεί την κατανόηση διαφόρων παραγόντων της αγοράς και τη λήψη αποφάσεων με βάση τις ιστορικές και τρέχουσες κινήσεις των τιμών. Περιλαμβάνει την επιλογή κατάλληλης αρχιτεκτονικής, την ενσωμάτωση τεχνικών δεικτών και την

ανάπτυξη αποτελεσματικών στρατηγικών εκπαίδευσης του πράκτορα. Επιπλέον είναι κρίσιμη η σχεδίαση ενός περιβάλλοντος που προσομοιώνει με ακρίβεια την αγορά ενός χρηματιστηρίου.

1) Initialize Parameters
Initialize primary network Q_θ, target network $Q_{\theta'}$ with random weights
Parameters: learning rate (a), discount factor (γ), target update target (τ),epsilon decay (ε_{start}, ε_{end}, ε_{decay})
Define action space A={-1, 1, 0} (short, long, hold)
Load stock market data
2) Training Loop (for each episode):
3) Reset environment and get initial state s_0
4) Set ε= max(ε_{start}, ε_{end}, ε_{decay})
5) While episode is not done:
6) Select action using epsilon-greedy policy:

$$a_t = \begin{cases} random\ action & with\ probability\ \varepsilon \\ \arg\max_a Q_\theta(s_t, a) & otherwise \end{cases}$$

7) Execute action a_t, observe next state s_{t+1} and reward r_t
8) Store transition (s_t, a_t, r_t, s_{t+1}, d_t)in replay memory D
9) Sample mini-batch (s, a, r, s′, d) ~D
10) Compute target Q-value

$$y = r + \gamma \cdot Q_{target}\left(s', \underset{a'}{argmax} Q_{online}(s', a')\right)$$

11) Compute loss and perform gradient descent step on:

$$L = \frac{1}{N}\sum \left(y - Q_{online}(s, a)\right)^2$$

12) Update target network parameters:

$$\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$$

13) If episode ends, store results and break
14) Save trained model Q_θ
15) Validation Phase (After each episode):
16) Reset test environment and get initial state
17) While validation is not done
18) Select action using ε-greedy policy (exploration allowed)
19) Execute action, observe next state and reward
20) Accumulate total validation reward

Ψευδοκώδικας 3: Dueling Double DQN

Ο Ψευδοκώδικας 3 απεικονίζει την αρχική δομή και τη γενική λογική του αλγορίθμου Dueling Double DQN. Οι λεπτομέρειες της τελικής υλοποίησης και οι όποιες προσαρμογές αναλύονται σε επόμενο κεφάλαιο.

Το μοντέλο ακολουθεί δύο κύριες φάσεις.

Φάση εκπαίδευσης (Training phase):

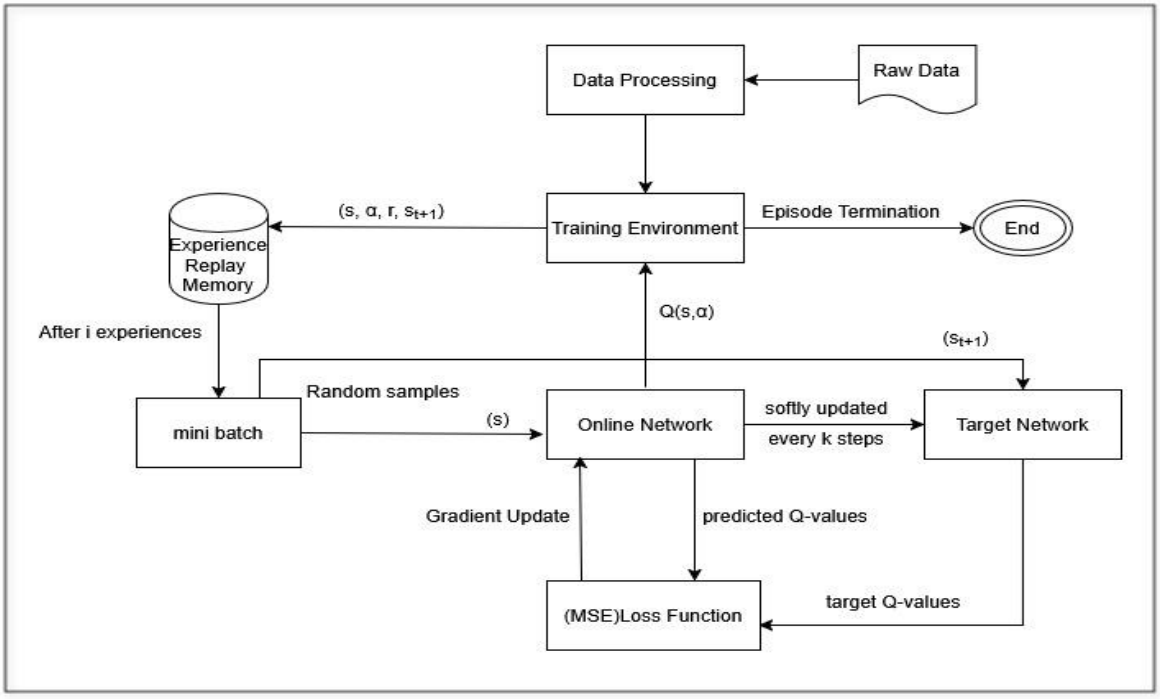
- Αρχικοποίηση δικτύων: Το primary (online) network και το target network αρχικοποιούνται με τυχαία βάρη. Ορίζονται οι παράμετροι όπως ο ρυθμός εκμάθησης (learning rate), ο συντελεστής έκπτωσης (γ) και η παράμετρος ενημέρωσης του target network (τ).
- Επεισόδια εκπαίδευσης: Η εκπαίδευση γίνεται μέσω επαναλαμβανόμενων επεισοδίων, όπου σε κάθε επεισόδιο ο πράκτορας ξεκινά από μια αρχική κατάσταση και αλληλεπιδρά με το περιβάλλον μέχρι να ολοκληρωθεί το χρονικό διάστημα ή να επιτευχθεί ο συγκεκριμένος στόχος.
- Επιλογή δράσης(ε-greedy policy): Ο πράκτορας επιλέγει δράση με πιθανότητα:
    - Τυχαία δράση (εξερεύνηση - exploration) με πιθανότητα ε
    - Η δράση με το μέγιστο Q-value (εκμετάλλευση - exploitation) από το primary network, όταν δεν γίνεται τυχαία επιλογή.
- Εκτέλεση δράσης και καταγραφή εμπειρίας: Αφού η δράση, εκτελείται στο περιβάλλον, λαμβάνεται η νέα κατάσταση και η ανταμοιβή, και αποθηκεύονται στη Replay memory ως tuples της μορφής (s, a, r, s', d)
    - s (state): Η τρέχουσα κατάσταση του περιβάλλοντος (π.χ., ένα διάνυσμα χαρακτηριστικών όπως τιμές μετοχών, δείκτες κ.λπ.).
    - a (action): Η δράση που επέλεξε ο πράκτορας (-1 για short, 0 για hold, 1 για long).
    - r (reward): Η ανταμοιβή που έλαβε ο πράκτορας μετά την εκτέλεση της δράσης.
    - s' (next state): Η επόμενη κατάσταση του περιβάλλοντος μετά την εκτέλεση της δράσης.
    - d (done flag): Boolean μεταβλητή που δείχνει αν το επεισόδιο έχει τελειώσει (1 αν έχει τελειώσει, 0 αν συνεχίζεται).
- Ενημέρωση δικτύου μέσω mini-batch training: Ο πράκτορας δειγματοληπτεί τυχαία mini-batches από την Replay memory και ενημερώνεται το online network.
    - Γίνεται ο υπολογισμός του target Q-value από το target network

  - Υπολογισμός της απώλειας (MSE Loss) μεταξύ του predicted και target Q-value
  - Προσαρμογή βαρών του online network μέσω Gradient Descent
- Ενημέρωση του Target network: Μετά από συγκεκριμένα βήματα, το target network ενημερώνεται
- Επανάληψη μέχρι να ολοκληρωθεί το επεισόδιο: Η διαδικασία συνεχίζεται μέχρι το επεισόδιο να ολοκληρωθεί. Στο τέλος κάθε επεισοδίου, αποθηκεύονται τα αποτελέσματα και το εκπαιδευμένο μοντέλο.

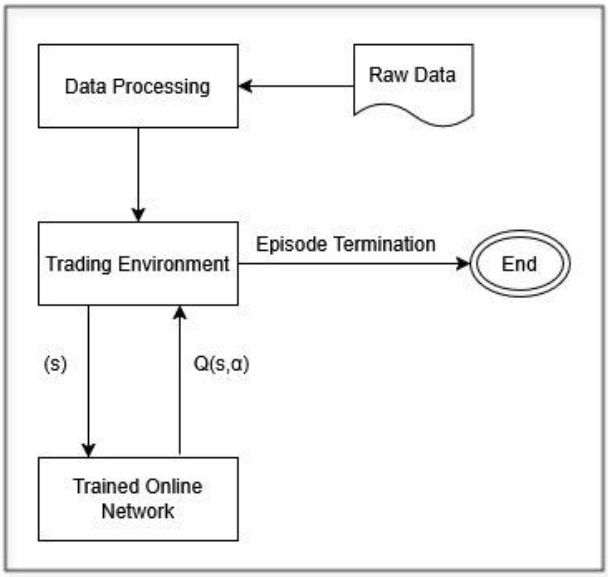
Φάση συναλλαγών (Trading phase)

Μετά την εκπαίδευση , το εκπαιδευμένο μοντέλο χρησιμοποιείται για να πραγματοποιήσει «πραγματικές» συναλλαγές

- Εκκίνηση περιβάλλοντος και επιλογή κατάστασης: Ο πράκτορας ξεκινά από μια αρχική κατάσταση s, η οποία αντιπροσωπεύει τα τρέχοντα δεδομένα της αγορά.
- Λήψη απόφασης μέσω του εκπαιδευμένου online network: Η κατάσταση s δίνεται ως είσοδος στο εκπαιδευμένο online network, το οποίο επιστρέφει Q-values για όλες τις πιθανές ενέργειες $A = \{-1, 1, 0\}$ (short, long, hold).
- Εκτέλεση της καλύτερη δράσης: Η ενέργεια με το μέγιστο Q-value επιλέγεται και εκτελείται στο περιβάλλον.
- Ενημέρωση της κατάστασης και επανάληψη: Η νέα κατάσταση s' ενημερώνεται και ο πράκτορας επαναλαμβάνει τη διαδικασία μέχρι να ολοκληρωθεί το trading session.



Εικόνα 6: Training flowchart



Εικόνα 7: Trading flowchart

2.2.1 Περιβάλλον

Όπως έχει αναφερθεί αρκετές φορές οι διαμόρφωση του περιβάλλοντος παίζει καθοριστικό ρόλο στην απόδοση του μοντέλου. Πρέπει να διαμορφωθεί κατάλληλα ώστε η προσομοίωση της λειτουργίας ενός χρηματιστηρίου να αντιπροσωπεύεται όσο πιο ρεαλιστικά γίνεται. Πρέπει να παρέχει την απαραίτητη πληροφορία και ανατροφοδότηση στον πράκτορα.

Σε κάθε βήμα πρέπει να παρέχονται βασικές πληροφορίες στον πράκτορα όπως, την ενέργεια που εκτέλεσε σε αυτό το βήμα, την τρέχουσα τιμή κλεισίματος, την ανταμοιβή για την ενέργεια που εκτέλεσε, την προηγούμενη θέση και δράση του και την επόμενη κατάσταση. Με αυτές τις πληροφορίες, ο πράκτορας μπορεί να κατανοήσει την τρέχουσα κατάσταση της αγοράς και να προσαρμόσει τις μελλοντικές του αποφάσεις αναλόγως.

Το σύστημά μας δέχεται 5 βασικές τιμές για οποιοδήποτε ψηφιακό στοιχείο αγοραπωλησίας τις οποίες έχουμε αναφέρει παραπάνω τι αντιπροσωπεύουν.

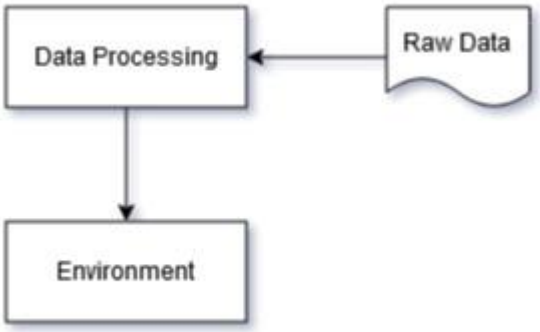
Άνοιγμα (Open), Υψηλό (High), Χαμηλό (Low), Όγκος (Volume), Κλείσιμο (Close).

Αυτές τις τιμές δεν τις περνάμε απευθείας στο περιβάλλον, τις επεξεργαζόμαστε μέσω τεχνικών δεικτών όπως θα δούμε παρακάτω, και χρησιμοποιούμε τις προκύπτουσες πληροφορίες για να δημιουργήσουμε πιο χρήσιμες και καθαρές καταστάσεις για τον πράκτορα.

Τις παραπάνω τιμές δεν τις περνάμε απευθείας στο σύστημά μας για τους εξής λόγους.

- Δυσκολία στην αναγνώριση μοτίβων: Οι ακατέργαστες τιμές (raw data), μπορεί να περιέχουν θόρυβο κάτι που θα δυσκολέψει τον πράκτορα στην αναγνώριση μοτίβων της αγοράς. Οι τιμές από μόνες τους δεν ενσωματώνουν συστηματική γνώση της αγοράς
- Αυξημένη πολυπλοκότητα: Η απουσία επεξεργασμένων δεικτών κάνει την αναζήτηση στρατηγικής περιπλοκότερη, επιβαρύνοντας τη διαδικασία εκμάθησης του πράκτορα, σπαταλώντας πόρους και χρόνο.
- Πρόβλημα κλιμάκωσης τιμών: Οι ακατέργαστες τιμές μπορεί να έχουν πολύ διαφορετικές κλίμακες, καθιστώντας δύσκολη την κανονικοποίηση για ομοιόμορφη είσοδο στο μοντέλο.

- Αποφυγή σημαντικών σημάτων: Οι τεχνικοί δείκτες μπορούν να βοηθήσουν στον εντοπισμό σημάτων που οι ακατέργαστες τιμές δεν μπορούν να αποκαλύψουν. Όπως θα δούμε παρακάτω, μπορούν για παράδειγμα να υποδείξουν αν κάποιο περιουσιακό στοιχείο είναι υπερπωλημένο ή υπεραγορασμένο, κάτι που δεν είναι προφανές μόνο από τις ακατέργαστες τιμές.



Εικόνα 8: Ροή δεδομένων

Εφαρμογή ορίων

Η εφαρμογή ορίων θέσεων στις αλγοριθμικές συναλλαγές είναι κρίσιμη για τη βελτιστοποίηση των αποφάσεων του πράκτορα και για μείωση του κινδύνου απώλειας κεφαλαίου. Κάποιοι από τους κύριους λόγους είναι:

- Διαχείριση κινδύνου: Τα όρια θέσεων βοηθούν στη μείωση πιθανών απωλειών. Ορίζοντας όρια στις θέσεις που μπορούν να είναι ταυτόχρονα ανοιχτές, μειώνουμε την πιθανότητα μεγάλων ζημιών αν η αγορά είναι ζημιογόνα. Είναι ευκολότερη η διαχείριση του χαρτοφυλακίου.
- Αποφυγή υπερσυναλλαγών (Overtrading): Ο περιορισμός σε ταυτόχρονες ανοιχτές θέσεις αποτρέπει τις υπερβολικές συναλλαγές χωρίς στρατηγική βάση. Οι υπερβολικές συναλλαγές μπορούν να αυξήσουν τα κόστη από προμήθειες και απώλεια κερδοφορίας.
- Βελτίωση της στρατηγική λήψης αποφάσεων: Ορίζοντας όρια θέσεων καθοδηγούμε τον αλγόριθμο να επιλέγει σωστότερες συναλλαγές για πιθανά

μελλοντικά κέρδη. Αυτό ενισχύει την ποιότητα των αποφάσεων του, εστιάζοντας σε πιο αποδοτικές συναλλαγές.

- Περιορισμός απώλειας κεφαλαίου: Η διατήρηση περιορισμένων θέσεων βοηθά ώστε να υπάρχει διαθέσιμο κεφάλαιο για μελλοντικές «ευκαιρίες» επενδύσεων.

Ενέργειες

Το περιβάλλον επιτρέπει τις παρακάτω ενέργειες στον πράκτορα τις οποίες έχουμε εξηγήσει την σημασία τους σε προηγούμενο κεφάλαιο.

- Hold
- Short
- Long

Παρακάτω παραθέτουμε παράδειγμα ενεργειών και ορίων θέσεων.

1. Δεδομένα: Αρχικό κεφάλαιο 100.000€. Μέγιστος αριθμός ταυτόχρονων ανοιχτών θέσεων 3. Κόστος ανά συναλλαγή 30€.
2. Άνοιγμα Short θέσης: Τρέχουσα τιμή μετοχής 50€. Ο πράκτορας αποφασίζει η αξία της short θέσης να είναι 30.000€. Άρα $30.000€/50€ = 600$ μετοχές. Ο πράκτορας δανείζεται 600 μετοχές και τις πουλάει στην αγορά. Από την πώληση των 600 μετοχών θα αποκομίσει $600 \cdot 50€ = 30.000€ - 30€$ (κόστος συναλλαγής) $= 29.970€$ καθαρά έσοδα. Τα μετρητά του χαρτοφυλακίου του αυξάνονται σε $100.000€\,(\text{αρχικό κεφάλαιο}) + 29.970€ = 129.970€$. Το ποσό των μετοχών που θα χρωστάει (η θέση του) είναι -600 τις οποίες πρέπει να τις επιστρέψει. Ανοιχτές θέσεις 1.
3. Άνοιγμα Long θέσης (ενώ η προηγούμενη short θέση παραμένει ανοιχτή): Τρέχουσα τιμή μετοχής 40€. Ο πράκτορας αποφασίζει να επενδύσει (αγοράσει μετοχές αξίας) 30.000€. Άρα $30.000€/40€ = 750$ μετοχές. Από την αγορά των 750 μετοχών, τα μετρητά του χαρτοφυλακίου του θα είναι $129.970€(\text{προηγούμενα μετρητά}) - 30.000€(\textit{κόστος μετοχών}) - 30€(\text{κόστος συναλλαγής}) = 99.940€$. Ανοιχτές θέσεις 2.

Μετά από αυτές τις δύο ενέργειες, ο πράκτορας έχει δύο ανοιχτές θέσεις: μία short θέση -600 μετοχών (με τιμή εισόδου 50€) και μία long θέση +750 μετοχών (με τιμή εισόδου 40€). Η καθαρή του έκθεση σε μετοχές είναι +150, αλλά οι δύο θέσεις παρακολουθούνται και διαχειρίζονται ξεχωριστά.

Ως τρέχουσα τιμή παίρνουμε την close price την δεδομένη στιγμή. Σε ένα ρεαλιστικό σενάριο η τιμή θα εξαρτάται από το order book όπως αναφέραμε παραπάνω.

Στο σενάριο μας θα έχουμε έως 3 θέσεις ανοιχτές. Το ποια θα κλείσει ώστε να ανοίξει κάποια άλλη διαθέσιμη, επιλέγεται με τα εξής κριτήρια, , είτε μετά από ένα συγκεκριμένο αριθμό βημάτων, είτε αν ξεπεράσει ένα όριο κέρδους/απώλειας που έχουμε ορίσει, είτε αυτή με το μικρότερο reward.

Καταστάσεις

Μια κατάσταση s την χρονική στιγμή t του περιβάλλοντος, αποτελείται από τις εξής πληροφορίες:

- Τεχνικοί δείκτες, στους οποίους υπάρχει αναλυτική αναφορά σε επόμενες ενότητες της διπλωματικής.

Technical indicators	Description
V-i	Percentage change in volume
R-i	Percentage change in close price
Sig-i	Volatility
BB	Bollinger Bands
RSI	Relative Strength Indicator
MACD	Moving Average Convergence Divergence
STC	Schaff Trend Cycle
SO	Stochastic Oscillator
ATR	Average True Range

Πίνακας 1: Τεχνικοί δείκτες

- Κατάσταση χαρτοφυλακίου

Portfolio ratios
P&L / Initial capital Running capital / P&L Position * Current price / Initial capital Previous action

Πίνακας 2: Portfolio ratios

Οι τεχνικοί δείκτες αποτελούν κρίσιμα εργαλεία στην ανάλυση των δεδομένων του περιβάλλοντος. Καθένας τους έχει σχεδιαστεί για να παρέχει συγκεκριμένες πληροφορίες που συνεισφέρουν στη λήψη αποφάσεων και τη βελτιστοποίηση στρατηγικών. Μέσω της συνδυαστικής χρήσης, μπορούμε να αποκτήσουμε μια πιο ολοκληρωμένη εικόνα για την κατάσταση της αγοράς, τις τάσεις και τη συμπεριφορά του χαρτοφυλακίου.

Οι τεχνικοί δείκτες που επιλέχθηκαν (βλέπε Πίνακας 1), όχι μόνο παρέχουν ακριβείς πληροφορίες για την κατάσταση των τιμών και της αγοράς, αλλά επίσης ενισχύουν τη στρατηγική λήψης αποφάσεων λαμβάνοντας υπόψη τις κινήσεις του χαρτοφυλακίου. Σε επόμενες ενότητες παρουσιάζονται με λεπτομέρεια οι δείκτες αυτοί, η σημασία τους και η συμβολή τους στη βελτίωση των αποφάσεων.

2.2.2 Τεχνικοί Δείκτες

Δείκτης PnL

Ο δείκτης PnL (Profit and Loss) είναι ένας σημαντικός δείκτης που χρησιμοποιείται στις χρηματοπιστωτικές αγορές για να παρακολουθεί και να αξιολογεί την απόδοση των επενδύσεων, των συναλλαγών και των χαρτοφυλακίων. Ο δείκτης PnL αναφέρεται στο συνολικό κέρδος ή ζημιάς που προκύπτει από μία επένδυση. Χρησιμοποιείται για την παρακολούθηση και την διαχείριση κινδύνου βοηθώντας τους επενδυτές να λαμβάνουν ενημερωμένες αποφάσεις για το αν θα κλείσουν ή θα διατηρήσουν ανοιχτές θέσεις (Bluefin, 2023).

$$PnL = \left(\sum_{Long\ position} (current\ price - entry\ price) \cdot position\ size \right) + \left(\sum_{Short\ positions} (entry\ price - current\ price) \cdot position\ size \right) - transaction\ costs \quad (2.1)$$

- Long position
  - Current price: Η τρέχουσα τιμή της μετοχής
  - Entry price: Η τιμή στην οποία αγοράστηκε η μετοχή
  - Position size: Το μέγεθος της θέσης, δηλαδή ο αριθμός των μονάδων του περιουσιακού στοιχείου που αγοράστηκαν.
- Short position
  - Current price: Η τρέχουσα τιμή της μετοχής
  - Entry price: Η τιμή στην οποία πουλήθηκε η μετοχή
  - Position size: Το μέγεθος της θέσης, δηλαδή ο αριθμός των μονάδων του περιουσιακού στοιχείου που αγοράστηκαν.
- Transaction costs: Τα κόστη συναλλαγών, όπως για παράδειγμα προμήθειες.

Δείκτης Sortino Ratio

Ο δείκτης Sortino Ratio είναι μια παραλλαγή του Sharpe Ratio που μετρά την απόδοση ενός χαρτοφυλακίου σε σχέση με το κίνδυνο, λαμβάνοντας υπόψη μόνο την αρνητική μεταβλητότητα (Rollinger & Hoffman, nd). Είναι ιδιαίτερα χρήσιμο για επενδυτικές στρατηγικές που επικεντρώνονται στην αποφυγή μεγάλων πτώσεων και επιδιώκουν σταθερές θετικές αποδόσεις.

$$Sortino\ Ratio = \frac{R_p - R_f}{\sigma_d} \quad (2.2)$$

όπου:

- R_p : Η μέση απόδοση του χαρτοφυλακίου

- R_f : Χωρίς κίνδυνο απόδοσης, (πχ. Επιτόκιο κρατικών ομολόγων που θεωρούνται ασφαλή λόγω της χαμηλής πιθανότητας αποτυχίας τους)
- σ_d: Τυπική απόκλιση των αρνητικών αποδόσεων

Δείκτης Drawdown

Ο δείκτης Drawdown είναι σημαντικός δείκτης για την αξιολόγηση του κινδύνου και της ανθεκτικότητας μια επενδυτικής στρατηγικής. Μετρά την πτώση της αξίας του χαρτοφυλακίου από την μέγιστη κορυφή του σε μια συγκεκριμένη χρονική περίοδο μέχρι το χαμηλότερο σημείο που φτάνει πριν ανακάμψει (Murphy, 1999).

$$Drawndown = \frac{Previous\ Peak\ Value - Trough\ Value}{Previous\ Peak\ Value} \quad (2.3)$$

- Ανώτερη αξία: (Peak Value): Το υψηλότερο σημείο που έχει φτάσει η αξία του χαρτοφυλακίου σε μια συγκεκριμένη χρονική περίοδο
- Κατώτερη αξία (Trough Value): Το χαμηλότερο σημείο που έχει φτάσει η αξία του χαρτοφυλακίου αφού έχει φτάσει στην κορυφή και πριν αρχίσει να ανακάμπτει.

Ποσοστιαία Αλλαγή (Percentage Change)

Είναι εξαιρετικά χρήσιμη για την ανάλυση των μεταβολών σε δεδομένα και χρονοσειρές. Όταν υπάρχουν δεδομένα από δύο διαφορετικά χρονικά σημεία, μπορούμε να υπολογίσουμε πόση μεταβολή υπήρξε κατά τη διάρκεια αυτής της περιόδου (Wikipedia contributors, n.d.). Υπολογίζεται ως:

$$Percentage\ Change = \frac{New\ Value\ -\ Old\ Value}{Old\ Value} \cdot 100 \quad (2.4)$$

Κυλιόμενη Τυπική Απόκλιση (Rolling Standard Deviation)

Είναι ένα στατιστικό μέτρο που χρησιμοποιείται για την ανάλυση χρονοσειρών. Αντί να υπολογίζουμε την τυπική απόκλιση για όλο το σύνολο δεδομένων, η κυλιόμενη τυπική απόκλιση υπολογίζεται σε υποσύνολα (παράθυρα) δεδομένων που κινούνται διαδοχικά

κατά μήκος της χρονοσειράς. Αυτό επιτρέπει την παρακολούθηση τοπικών αλλαγών στη μεταβλητότητα και ενημερώνεται συνεχώς καθώς εισέρχονται νέα δεδομένα. Η χρήση της μπορεί να αποκαλύψει εποχικές τάσεις ή να ανιχνεύσει ανωμαλίες, όπως αυξημένες ή μειωμένες τιμές της κυλιόμενης τυπικής απόκλισης μπορεί να υποδείξουν ασυνήθιστη δραστηριότητα (Sanghavi & Benedict, 2024).

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2} \qquad (2.5)$$

- σ = τυπική απόκλιση
- x_i είναι η i-οστή τιμή του συνόλου δεδομένων
- μ είναι η μέση τιμή του συνόλου δεδομένων
- n είναι ο αριθμός των τιμών στο παράθυρο.

Κινητός μέσος όρος (Moving Average)

Ο κινητός μέσος όρος υπολογίζεται για τον προσδιορισμό της κατεύθυνσης της τάσης μιας μετοχής. Μπορούν να επιλεχθούν διαφορετικές περίοδοι με διαφορετικό μήκος για τον υπολογισμό κινητών μέσων όρων. Οι βραχύτεροι κινητοί μέσοι όροι χρησιμοποιούνται συνήθως για βραχυπρόθεσμες συναλλαγές ενώ η μακροπρόθεσμοι είναι καταλληλότεροι για μακροπρόθεσμες επενδύσεις. Υπάρχουν διάφοροι τύποι κινητών μέσων όρων (Wikipedia contributors, n.d.).

- Απλός κινητός μέσος όρος (Simple Moving Average - SMA): Είναι ένας από τους πιο βασικούς τύπους κινητών μέσων όρων. Χρησιμοποιείται κυρίως για την εξομάλυνση των τιμών μιας χρονοσειράς και την αναγνώριση τάσεων στην αγορά. Υπολογίζεται από τον τύπο

$$SMA = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (2.6)$$

  - n είναι ο αριθμός των περιόδων (πχ ημέρες, εβδομάδες) για τις οποίες υπολογίζεται ο μέσος όρος

- x_i είναι η τιμή της μετοχής στην i-οστή περίοδο

- Εκθετικός κινητός μέσος όρος (Exponential moving average - EMA): Δίνει μεγαλύτερη βαρύτητα στις πρόσφατες τιμές σε μια προσπάθεια να τις κάνει να ανταποκρίνονται καλύτερα σε νέες πληροφορίες. Για τον υπολογισμό ενός εκθετικού κινητού μέσου όρου, υπολογίζεται πρώτα ο απλός κινητός μέσος όρος για μια συγκεκριμένη περίοδο. Στη συνέχεια υπολογίζεται ο πολλαπλασιαστής για τη στάθμιση του EMA, αναφέρεται ως «συντελεστής εξομάλυνσης» $= \frac{2}{n-1}$ όπου n ο αριθμός των περιόδων. Ο τύπος του EMA είναι ο

$$EMA_t = close_{price} \cdot multiplier + EMA_{t-1} \cdot (1 - multiplier) \qquad (2.7)$$

  - $close\_price$: η τιμή κλεισίματος της τρέχουσας περιόδου
  - EMA_t είναι ο EMA την τρέχουσα περιόδου
  - EMA_{t-1} είναι ο EMA της προηγούμενης περιόδου
  - $multiplier$ είναι ο συντελεστής εξομάλυνσης.

  Ο EMA είναι ευέλικτος και προσαρμόζεται γρήγορα στις αλλαγές των τιμών, καθιστώντας τον χρήσιμο για τον εντοπισμό βραχυπρόθεσμων τάσεων.

- Σύγκλιση – απόκλιση κινητών μέσων όρων (Moving average convergence divergence - MACD): Είναι ένας δείκτης που ακολουθεί την κατεύθυνση της μετοχής και δείχνει τη σχέση σε δύο κινητούς μέσους όρους. Πρώτα υπολογίζονται 2 εκθετική μέσοι όροι, Short-term EMA συνήθως για 12 περιόδους και Long-term EMA συνήθως για 26 περιόδους. Μετά ο MACD υπολογίζεται ως η διαφορά του short-term EMA και του Long-term EMA. Για την περαιτέρω ανάλυση του MACD, εφαρμόζεται πάνω του ένας EMA 9 περιόδων, που ονομάζεται γραμμή σήματος (Signal Line), και χρησιμοποιείται ως σημείο αναφοράς για πιθανή αντιστροφή της τάσης. Στην συνέχεια υπολογίζεται το ιστόγραμμα (histogram) του MACD, το οποίο δείχνει τη διαφορά μεταξύ του MACD και της γραμμής του σήματος. Όταν ο MACD είναι πάνω από τη γραμμή σήματος, το ιστόγραμμα είναι θετικό και δείχνει ανοδική τάση, ενώ αν είναι κάτω από τη γραμμή σήματος δείχνει καθοδική τάση. Στον παραπάνω υπολογισμό, μπορεί να προστεθεί και μία βασική γραμμή αναφοράς (baseline), η οποία είναι ένας επιπλέον EMA 9 περιόδων και μπορεί να λειτουργήσει ως συμπληρωματικό εργαλείο για την αξιολόγηση της τάσης.

Ζώνες Bollinger (Bollinger Bands):

Οι ζώνες Bollinger δείχνουν την μεταβλητότητα της τιμής σε σχέση με προηγούμενες συναλλαγές που έχουν πραγματοποιηθεί. Δείχνουν αν η αγορά έχει υψηλή ή χαμηλή μεταβλητότητα, δηλαδή αν υπάρχουν μικρές ή μεγάλες διακυμάνσεις στη τιμή (Bollinger, 2001). Ο τύπος των ζωνών, καθορίζεται από την κεντρική γραμμή η οποία υπολογίζεται ο απλός μέσος όρος (SMA) μιας χρονοσειράς τιμών για ένα συγκεκριμένο αριθμό περιόδων (πχ 20 ημέρες).

- Η άνω γραμμή(Band) υπολογίζεται από τον τύπο

$$Upper\ Band = SMA + (2 \cdot Standard\ Deviation) \qquad (2.8)$$

- Η κάτω γραμμή (Band) υπολογίζεται από τον τύπο

$$Lower\ Band = SMA - (2 \cdot\ Standard\ Deviation) \qquad (2.9)$$

Όταν οι άνω και κάτω γραμμές αποκλίνουν σημαίνει ότι υπάρχουν μεγάλες διακυμάνσεις τιμών, ενώ όταν συγκλίνουν υπάρχουν μικρές διακυμάνσεις και η αγορά είναι πιο σταθερή.

Δείκτης σχετικής ισχύος (Relative strength index - RSI)

Ο RSI μετράει τη ταχύτητα και την δύναμη των μεταβολών, βοηθώντας στο να εντοπιστούν υπεραγορασμένες ή υπερπουλημένες μετοχές σε μια συγκεκριμένη χρονική περίοδο. Ο δείκτης σχεδιάζεται σε ένα εύρος μεταξύ του 0 και του 100. Μια τιμή μεγαλύτερη από 70 δείχνει ότι η αξία είναι υπεραγορασμένη, ενώ μια τιμή κάτω από το 30 δείχνει ότι είναι υπερπουλημένη (Wikipedia contributors, n.d.). Ο τύπος υπολογισμού είναι

για συγκεκριμένη περίοδο $$RS = \frac{Average\ gain}{Average\ loss} \qquad (2.10)$$

$$RSI = 100 - \frac{100}{1 + RS} \qquad (2.11)$$

Πραγματικό εύρος (True Range - TR)

Ο TR χρησιμοποιείται για να μετρά το εύρος των διακυμάνσεων των τιμών μιας μετοχής, βοηθώντας να εντοπίσουμε πόσο έντονα αλλάζουν οι τιμές (Wilder, 1978). Χρησιμοποιείται για να υπολογιστεί ο ATR όπως θα δούμε παρακάτω . Ο TR υπολογίζεται παίρνοντας το μέγιστο από τα εξής τρία:

$$TR = High - Low \tag{2.12}$$

όπου High το υψηλότερο υψηλό και Low το χαμηλότερο χαμηλό

Την απόλυτη τιμή,

$$TR = |High - Previous\ close| \tag{2.13}$$

όπου previous close η τιμή κλεισίματος της προηγούμενης περιόδου.

Την απόλυτη τιμή,

$$TR = |Low - Previous\ close| \tag{2.14}$$

Μέσος πραγματικός δείκτης (Average True Range - ATR)

Ο ATR χρησιμοποιείται για να υπολογιστεί η μεταβλητότητα των μετοχών για μια χρονική περίοδο (Wilder, 1978). Για τον υπολογισμό βασίζεται στο μέσο όρο των τιμών του TR. Χρησιμοποιώντας τον EMA ο τύπος είναι ο

$$ATR = \frac{\left(ATR_{t-1} \cdot (n-1)\right) + TR_t}{n} \tag{2.15}$$

Χρησιμοποιώντας το SMA ο τύπος είναι

$$ATR = \frac{1}{n}\sum_{i=1}^{n} TR_i \tag{2.16}$$

Η επιλογή στο ποια θα χρησιμοποιηθεί εξαρτάται από τον τρόπο που θέλουμε να ανταποκρίνεται το σύστημα στις αλλαγές της αγοράς. Αν θέλουμε ταχύτερη αντίδραση στις αλλαγές της αγοράς διότι δίνει μεγαλύτερη βαρύτητα στις πιο πρόσφατες τιμές, το EMA

είναι προτιμότερο. Αν θέλουμε μια πιο σταθερή μεταβλητότητα, καθώς δίνει ίση βαρύτητα σε όλες τις περιόδους, το SMA είναι καλύτερο.

Στοχαστικός Ταλαντωτής (Stochastic Oscillator)

Ο στοχαστικός ταλαντωτής μετρά τη θέση της τιμής κλεισίματος σε σχέση με το εύρος των τιμών για μια συγκεκριμένη χρονική περίοδο. Χρησιμοποιείται κυρίως για να εντοπιστούν υπεραγορές ή υπερπωλήσεις στην αγορά. Ο τύπος υπολογισμού είναι (Wikipedia contributors, n.d.).

$$\%K = \left(\frac{close\ price - Lowest\ Low}{Highest\ High - Lowest\ Low}\right) \cdot 100 \qquad (2.17)$$

$$\%D = \frac{\sum_{i=1}^{n} \%k_i}{n} \qquad (2.18)$$

Ο $\%K$ υπολογίζει τη θέση της τιμής κλεισίματος σε σχέση με το εύρος των τιμών

Ο $\%D$ χρησιμοποιεί τον απλό κινητό μέσο όρο (SMA) για τον υπολογισμό των τιμών της $\%K$ για συγκεκριμένη χρονική περίοδο (n)

Κύκλος τάσης Schaff (Schaff Trend Cycle - STC)

Ο STC είναι ένας τεχνικός δείκτης που βοηθά στην αναγνώριση τάσεων και τις πιθανές ανακατανομές που μπορεί να συμβούν στην αγορά. Χρησιμοποιείται επίσης για τον εντοπισμό συνθηκών υπεραγοράς ή υπερπώλησης μιας μετοχή. Ο τρόπος υπολογισμού του, ο οποίος συνδυάζει τον MACD και τον Stochastic Oscillator, βοηθάει στη μείωση των λανθασμένων σημάτων, καθιστώντας τον πιο αξιόπιστο σε σύγκριση με άλλους δείκτες (Schaff, 2008). Ο Τύπος υπολογισμού του είναι

$$STC = 100 \cdot \frac{MACD - \%K(MACD)}{\%D(MACD) - \%K(MACD)} \qquad (2.19)$$

2.2.3 Συνάρτηση ανταμοιβής

Η συνάρτηση ανταμοιβής όπως έχει ήδη αναφερθεί παίζει καθοριστικό ρόλο στην εκπαίδευση του πράκτορα. Η αλληλεπίδραση του πράκτορα με το περιβάλλον και οι αποφάσεις που θα επιλέξει βασίζονται κυρίως στην συνάρτηση ανταμοιβής, επηρεάζοντας αλυσιδωτές αποφάσεις, καθώς μία λάθος επιλογή μπορεί να έχει καταστροφικά αποτελέσματα. Η σημαντικότητα της σχεδίασης μιας δυναμικής συνάρτησης ανταμοιβής που προσαρμόζεται στις αλλαγές του περιβάλλοντος έχει μελετηθεί εκτενώς (Huang et al., 2024). Η δημιουργία μία συνάρτησης ανταμοιβής απαιτεί προσεκτική και πολύπλευρη έρευνα για να υπάρχει ισορροπία μεταξύ της εξερεύνησης και εκμετάλλευσης. Θα πρέπει οι λάθος επιλογές να επιβαρύνονται με κάποια ποινή, τέτοια όμως που να μην δημιουργεί τον πράκτορα διστακτικό στην εξερεύνηση. Στον κώδικά μας η συνάρτηση ανταμοιβής έχει σχεδιαστεί με βάση τους εξής παράγοντες όπως αναλύεται και στην διπλωματική του (Azis, 2024), όπου μεταξύ άλλων εξετάζεται η σημασία του κόστους συναλλαγών σε περιβάλλοντα HFT:

- Αντιμετώπιση κόστους συναλλαγών: Στις συναλλαγές υψηλής συχνότητας (HFT), το κόστος συναλλαγών αποτελεί σημαντικό πρόβλημα λόγω του μεγάλου αριθμού συναλλαγών που εκτελούνται σε σύντομο χρονικό διάστημα. Έτσι, προσθέσαμε έναν επιπλέον παράγοντα στην συνάρτηση ανταμοιβή, ο οποίος παροτρύνει τον πράκτορα να λαμβάνει υπόψη τα κόστη, όχι μόνο για τις οικονομικές επιπτώσεις αλλά και την αβεβαιότητα που δημιουργούν οι συχνές αλλαγές.
- Αλλαγές θέσεων και ανταμοιβής: Η συμπεριφορά του πράκτορα δεν καθορίζεται μόνο από τις σωστές επιλογές όσον αφορά την τάση της μετοχής, αλλά και από την τρέχουσα κατάσταση του χαρτοφυλακίου του. Επιβραβεύουμε και τιμωρούμε τις αλλαγές στο position του πράκτορα, ώστε να καταλαβαίνει πότε οι κινήσεις του αποδίδουν κέρδος, χρησιμοποιώντας το return του χαρτοφυλακίου του μετά από κάθε συναλλαγή.
- Επαναφορά θέσεων: Οι θέσεις του πράκτορα επαναφέρονται στο τέλος κάθε ημέρας (Position = 0), ώστε να έχουμε μια αντιπροσωπευτική εικόνα για την απόδοση του πράκτορα και να τον επιβραβεύσουμε ή να τον τιμωρήσουμε ανάλογα. Πιο συγκεκριμένα, αν ο πράκτορα «χρωστάει» μετοχές (αρνητικό position), τον αναγκάζουμε να επιστρέψει σε μηδενικό position στέλνοντας μια εντολή αγοράς.

Αντίστοιχα, αν ο πράκτορας τερματίζει την ημέρα με θετικό position, γίνεται πώληση των μετοχών που έχει στην κατοχή του.

Με αυτό τον τρόπο, κάθε ημέρα ξεκινά από μηδενική βάση, διευκολύνοντας την αξιολόγηση της απόδοσης του πράκτορα και την προσαρμογή της στρατηγικής του αναλόγως.

- Αποστροφή κινδύνου: Χρησιμοποιώντας έναν σταθερό αρνητικό παράγοντα για τις ζημιές, ενισχύεται η αποφυγή ριψοκίνδυνων ενεργειών.

Όταν το position δεν αλλάζει, η χρήση μίας απόδοσης ως ανταμοιβή προς τον πράκτορα θα τον βοηθήσει ώστε να μάθει να διατηρεί θέσεις όταν η αγορά κινείται προς όφελός του, ενισχύοντας τη στρατηγική παραμονής σε κερδοφόρες τάσεις, να αποφεύγει τις υπερβολικές αλλαγές θέσεων, μειώνοντας το κόστος συναλλαγών όπως επίσης να προσαρμόζει τη στρατηγική του όταν το position δεν είναι ευθυγραμμισμένο με την αγορά, ώστε να περιορίζει τις ζημιές και να μεγιστοποιεί τα κέρδη.

Βάση αυτών που αναφέρθηκαν η συνάρτηση ανταμοιβής υπολογίζεται ως

$$PnL_{based}Reward = \begin{cases} \frac{PnL_t - PnL_{t-1}}{|PnL_t|} \;\; if\ position\ has\ changed\ and\ reward \geq 0 \\ 100 \cdot r \cdot a \;\; if\ position\ has\ not\ changed \\ \frac{PnL_t - PnL_{t-1}}{|PnL_t|} \cdot C \; if\ position\ has\ changed\ and\ reward < 0 \end{cases} \quad (2.20)$$

Όπου:

- r: Η ποσοστιαία απόδοση του επόμενου χρονικού βήματος

$$r = \frac{close\_price_{t+1} - close\_price_t}{close\_price_t} \quad (2.21)$$

- a: Η ενέργεια όπου έχει γίνει $(short(-1), hold(0)\ long(1))$
- Ο πολλαπλασιασμός με το 100 γίνεται για τη μετατροπή της απόδοσης σε ποσοστό.
- C ένας σταθερός αρνητικός παράγοντας.

Το συνολικό $Reward$ διαμορφώνεται

$$Reward = \alpha \cdot PnL\_based\ Reward + \beta \cdot Sortino\ Ratio - \gamma \cdot Drawndown \quad (2.22)$$

όπου α, β, γ οι παράμετροι βαρύτητας.

Η συνολική συνάρτηση ανταμοιβής που διαμορφώνεται περιλαμβάνει τους δείκτες PnL, Sortino Ratio και Drawdown, οι οποίοι επιλέχθηκαν για λόγους που έχουν ήδη αναφερθεί, όπως η ισορροπία μεταξύ εξερεύνησης και εκμετάλλευσης, η βελτιστοποίηση της απόδοσης και η αποφυγή υπερβολικού κινδύνου. Η ενσωμάτωση αυτών των δεικτών στη συνάρτηση ανταμοιβής δικαιολογείται πλήρως από τις αρχές που παρουσιάζονται στο έργο των (Rodinos et al. 2024).

2.2.4 Experience Replay

Η χρήση μνήμης επανάληψης (Replay Memory, βλέπε Εικόνα 9) αποτελεί μια θεμελιώδη πρακτική στην εκπαίδευση πρακτόρων βαθιάς ενισχυτικής μάθησης. Η συγκεκριμένη προσέγγιση επιλέχθηκε καθώς συμβάλει στην αποδοτική αξιοποίηση των δεδομένων που συλλέγονται κατά την αλληλεπίδραση του πράκτορα με το περιβάλλον. Το μοντέλο που χρησιμοποιήθηκε βασίζεται σε μια δομή δεδομένων Deque (Double-Ended Queue), το οποίο παρέχει γρήγορες λειτουργίες προσθήκης και αφαίρεσης στοιχείων και από τα δύο άκρα με χρονική πολυπλοκότητα O(1).

Πλεονεκτήματα της χρήσης Replay Memory:

- Αποφυγή χρονικής συσχέτισης: Η διαδοχική εξάρτηση μεταξύ των καταστάσεων που συλλέγονται κατά την αλληλεπίδραση του πράκτορα με το περιβάλλον μπορεί να οδηγήσει σε φτωχή γενίκευση και μη σταθερή εκπαίδευση. Με τη χρήση μνήμης επανάληψης και τυχαίας δειγματοληψίας εμπειριών, εξαλείφεται η συσχέτιση μεταξύ διαδοχικών παρατηρήσεων.
- Σταθερότητα στην εκπαίδευση: Η αποθήκευση εμπειριών και η εκμάθηση από ένα ευρύ σύνολο δεδομένων βελτιώνει τη σταθερότητα της εκπαίδευσης, καθώς αποφεύγεται η συνεχής προσαρμογή των βαρών του νευρωνικού δικτύου με πολύ λίγα σημεία δεδομένων. Επιπλέον η χρήση μνήμης εμπειριών εξασφαλίζει ότι η εκπαίδευση δεν θα ξεκινήσει αν δεν υπάρχουν επαρκή δεδομένα, καθώς η εκπαίδευση αρχίζει μόνο όταν το πλήθος των αποθηκευμένων εμπειριών υπερβαίνει το κατώφλι που ορίζεται από την παράμετρο LEARN_AFTER (βλέπε Εικόνα 10).

- Αύξηση αποτελεσματικότητας εκμάθησης: Ο πράκτορας χρησιμοποιεί μια μεγάλη δεξαμενή εμπειριών, αντλώντας τυχαία δείγματα από τη μνήμη, μεγέθους BATCH_SIZE, η οποία καθορίζει τον αριθμό των εμπειριών που αντλούνται για κάθε επανάληψη εκμάθησης. Αυτό ενισχύει τη δυνατότητα γενίκευσης και την αναγνώριση μοτίβων.

Η δομή δεδομένων διασφαλίζει ότι η μνήμη έχει περιορισμένη χωρητικότητα την οποία ορίζουμε μέσω της MEMORY_LEN. Όταν γεμίσει, οι παλαιότερες εμπειρίες αντικαθίστανται από τις νεότερες, αποτρέποντας υπερφόρτωση μνήμης.

```
Transition = namedtuple( typename: "Transition",  field_names: ["States", "Actions", "Rewards", "NextStates", "Dones"])

class ReplayMemory():

    def __init__(self, capacity=MEMORY_LEN):
        self.memory = deque(maxlen=capacity)

    def store(self, t):
        self.memory.append(t)

    def sample(self, n=BATCH_SIZE):
        a = random.sample(self.memory, n)
        return a

    def __len__(self):
        return len(self.memory)
```

Εικόνα 9: Replay memory

```python
def learn(self):
    if len(self.memory) <= LEARN_AFTER:
        return 0

    # Sample experiences from memory at defined intervals
    if self.t_step > LEARN_AFTER and self.t_step % LEARN_EVERY == 0:

        batch = self.memory.sample()

        # Convert batch to tensors
        states = T.from_numpy(np.vstack([t.States for t in batch])).float().to(DEVICE)
        actions = T.from_numpy(np.vstack([t.Actions for t in batch])).long().to(DEVICE)
        rewards = T.from_numpy(np.vstack([t.Rewards for t in batch])).float().to(DEVICE)
        next_states = T.from_numpy(np.vstack([t.NextStates for t in batch])).float().to(DEVICE)
        dones = T.from_numpy(np.vstack([t.Dones for t in batch])).float().to(DEVICE)

        # Compute target Q-values using Double DQN approach
        best_actions_online = self.actor_online(next_states).argmax(1).unsqueeze(1)
        next_state_values = self.actor_target(next_states).gather(1, best_actions_online)
        y = rewards + (1 - dones) * GAMMA * next_state_values
        state_values = self.actor_online(states).gather(1, actions.long())

        # Compute loss and optimize network
        actor_loss = self.actor_criterion(y, state_values)
        self.actor_op.zero_grad()
        actor_loss.backward()
        self.actor_op.step()

        # Update target network periodically
        if self.t_step % UPDATE_EVERY == 0:
            self.soft_update(self.actor_online, self.actor_target)
```

Εικόνα 10: Learn method

2.2.5 Δίλημμα εξερεύνηση vs εκμετάλλευση

Στο πλαίσιο της εκπαίδευσης πρακτόρων ενισχυτικής μάθησης, είναι σημαντικό να επιτρέπουμε στον πράκτορα να αναλαμβάνει τυχαίες ενέργειες (εξερεύνηση) αντί να ακολουθεί πάντα την πολιτική που έχει μάθει (εκμετάλλευση). Αυτή η στρατηγική βοηθά τον πράκτορα να ανακαλύπτει πως άλλες ενέργειες, πέρα από αυτές που προτείνονται, επηρεάζουν την ανταμοιβή που λαμβάνει. Η εξερεύνηση μπορεί να οδηγήσει σε καλύτερες ή χειρότερες ανταμοιβές, αλλά προσθέτει αξία στη διαδικασία μάθησης του πράκτορα, επιτρέποντάς του να ανακαλύψει νέες στρατηγικές και να βελτιώσει τις επιλογές του.

Στην αρχή της εκπαίδευσης, απαιτείται περισσότερη εξερεύνηση από την πλευρά του πράκτορα. Καθώς η εκπαίδευση προχωρά, η αναλογία εξερεύνησης προς εκμετάλλευσης αλλάζει, και ο πράκτορας πρέπει να αρχίσει να εκμεταλλεύεται τις δράσεις που έχει μάθει ότι έχουν καλύτερες ανταμοιβές. Αυτό σημαίνει ότι θέλουμε ο πράκτορας να μειώνει σταδιακά την τυχαιότητα στις επιλογές των ενεργειών του.

Η ισορροπία μεταξύ εξερεύνησης και εκμετάλλευσης μπορεί να ελεγχθεί μέσω παραμέτρων:

- EPS_START: Το αρχικό ποσοστό τυχαίων ενεργειών που πραγματοποιεί ο πράκτορας στην αρχή της εκπαίδευσης.
- EPS_END: Το ελάχιστο ποσοστό τυχαίων ενεργειών που θα πραγματοποιεί ο πράκτορας προς το τέλος της εκπαίδευσης
- EPS_DECAY: Ο ρυθμός μείωσης της εξερεύνησης, δηλαδή το πώς το ποσοστό τυχαίων ενεργειών μειώνεται με την πάροδο του χρόνου.

Με αυτές τις παραμέτρους, μπορούμε να ελέγξουμε την μετάβαση του πράκτορα από τη φάση της εξερεύνηση στη φάση της εκμετάλλευσης, διασφαλίζοντας ότι αποκτά ισορροπημένη κατανόηση του περιβάλλοντος και των δυνατοτήτων του, βελτιώνοντας έτσι τη συνολική απόδοση.

2.2.6 Αρχιτεκτονική δικτύου

Η σωστή επιλογή της αρχιτεκτονικής του δικτύου παίζει καθοριστικό ρόλο για την απόδοση και την επιτυχία ενός μοντέλου. Θα πρέπει να σχεδιαστεί κατάλληλα ώστε η επεξεργασία των δεδομένων και οι τιμές που θα εξαχθούν να είναι κατάλληλες για το πρόβλημα που προσπαθούμε να επιλύσουμε. Στο σύστημά μας, η αρχιτεκτονική βασίζεται σε δύο αρχιτεκτονικά πανομοιότυπα νευρωνικά δίκτυα, το Online και το Target network, μια προσέγγιση που αποτελεί τον πυρήνα του Double Q-learning. Η διαφορά τους δεν είναι στη δομή, αλλά στον ρόλο που παίζουν και στο πότε ενημερώνονται τα βάρη τους. Το Online δίκτυο είναι το «ενεργό» καθώς μαθαίνει συνεχώς σε κάθε βήμα εκπαίδευσης και είναι αυτό που χρησιμοποιείται για την επιλογή των δράσεων. Αντίθετα, το Target δίκτυο είναι το «σταθερό». Λειτουργεί ως ένα παλαιότερο, αντίγραφο του Online δικτύου και

χρησιμοποιείται για τον υπολογισμό των τιμών στόχου (target Q-values) κατά την εκπαίδευση.

Η εσωτερική αρχιτεκτονική καθενός από αυτά τα δίκτυα αποτελεί ένα συνδυασμό της Dueling αρχιτεκτονικής με βαθιά νευρωνικά δίκτυα LSTM, μια προσέγγιση που έχει αποδειχθεί αποτελεσματική σε χρηματοοικονομικά περιβάλλοντα με σύνθετες αλληλουχίες δεδομένων, όπως περιγράφεται στην διπλωματική του (Azis, 2024).

Η εσωτερική τους δομή φαίνεται στην Εικόνα 11. Σε κάθε ένα από τα δύο αυτά δίκτυα, η πληροφορία που εισέρχεται (το διάνυσμα της κατάστασης) περνάει πρώτα από τρία διαδοχικά επίπεδα LSTM. Τα LSTM δίκτυα είναι ιδανικά για το πρόβλημά μας, καθώς είναι εξειδικευμένα στο να αναγνωρίζουν μοτίβα και μακροπρόθεσμες εξαρτήσεις σε δεδομένα που έχουν χρονική διάσταση. Παρόλο που κατά την εκπαίδευση με experience memory ο πράκτορας μαθαίνει από τυχαίες, μη διαδοχικές εμπειρίες, κάθε μεμονωμένη κατάσταση s περιέχει χρονική πληροφορία. Για παράδειγμα, του παρέχουμε τις αποδόσεις των τελευταίων 5, 10, και 40 λεπτών, καθώς και δείκτες όπως το MACD που υπολογίζονται πάνω σε ένα ιστορικό χρονικό παράθυρο. Τα LSTM λοιπόν, μαθαίνουν να βρίσκουν σύνθετες, μη-γραμμικές σχέσεις και μοτίβα μεταξύ αυτών των χρονικών χαρακτηριστικών. Για παράδειγμα, μαθαίνουν ότι «όταν η βραχυπρόθεσμη τάση είναι ανοδική, αλλά η μακροπρόθεσμη ήταν πτωτική, αυτό μπορεί να σηματοδοτεί μια ευκαιρία αγοράς». Η ιεραρχική δομή των LSTM επιπέδων (128→64→32 νευρώνες) λειτουργεί σαν ένα «φίλτρο» που μαθαίνει σταδιακά να απομονώνει τις πιο σημαντικές, αφηρημένες πληροφορίες (Azis, 2024), επιτρέποντας τη βαθύτερη κατανόηση των δεδομένων και βελτιώνοντας την απόδοση του μοντέλου, όπως υποστηρίζεται και από τη βιβλιογραφία (LeCun et al, 2015) και (Bengio, 2009).

Μετά τα LSTM, η επεξεργασμένη πληροφορία εισέρχεται στην Dueling αρχιτεκτονική. Εδώ, το δίκτυο διαχωρίζεται σε δύο ροές (streams) και μαθαίνει να απαντά σε δύο ερωτήσεις ταυτόχρονα. Την Value stream (V) «πόσο καλή είναι η τρέχουσα κατάσταση της αγοράς, ανεξάρτητα από τη δράση που θα επιλεγεί» και η Advantage stream (A) «ποιο είναι το σχετικό πλεονέκτημα κάθε δράσης (buy, sell, hold) για αυτή τη συγκεκριμένη κατάσταση»; Αυτές οι δύο απαντήσεις συνδυάζονται στο τέλος για να δώσουν την τελική, πιο αξιόπιστη εκτίμηση της Q-τιμής και επιτρέπει στο μοντέλο να αξιολογεί καλύτερα την ποιότητα μιας κατάστασης, οδηγώντας σε πιο ακριβείς και σταθερές αποφάσεις.

```python
class Dueling(nn.Module):
    """Dueling architecture with LSTM layers."""

    def __init__(self, input_dim=STATE_SPACE, output_dim=ACTION_SPACE):
        super(Dueling, self).__init__()
        self.input_dim = input_dim
        self.output_dim = output_dim

        # First LSTM layer with Layer Normalization
        self.lstm1 = nn.LSTM(input_size=input_dim, hidden_size=128)
        self.layer_norm1 = nn.LayerNorm(128)

        # Second LSTM layer with Layer Normalization
        self.lstm2 = nn.LSTM(input_size=128, hidden_size=64)
        self.layer_norm2 = nn.LayerNorm(64)

        # Third LSTM layer with Layer Normalization
        self.lstm3 = nn.LSTM(input_size=64, hidden_size=32)
        self.layer_norm3 = nn.LayerNorm(32)

        # Fully connected layers for state-value (V) and advantage (A) streams
        self.V = nn.Linear( in_features: 32,  out_features: 1) # State-value function
        self.A = nn.Linear( in_features: 32, self.output_dim) # Advantage function

        self.tanh = nn.Tanh()
```

Εικόνα 11: Dueling architecture

```python
def forward(self, state):
    """Forward pass through the network."""

    # Pass state through the first LSTM layer
    lstm1_output, _ = self.lstm1(state)
    x = self.layer_norm1(self.tanh(lstm1_output))

    # Pass through the second LSTM layer
    lstm2_output, _ = self.lstm2(x)
    x = self.layer_norm2(self.tanh(lstm2_output))

    # Pass through the third LSTM layer
    lstm3_output, _ = self.lstm3(x)
    x = self.layer_norm3(self.tanh(lstm3_output))

    # Compute state-value (V) and advantage (A)
    V = self.V(x)
    A = self.A(x)

    # Combine state-value and advantage using the dueling architecture
    x = V + A - A.mean(dim=1, keepdim=True)

    return x
```

Εικόνα 12: Forward method

Η κανονικοποίηση ανά επίπεδο (Layer Normalization) προσφέρει σταθερότητα στην εκπαίδευση, μειώνοντας την εξάρτηση από ακραίες τιμές εισόδου. Εξασφαλίζει ότι οι τιμές των δεδομένων μένουν σε σταθερό εύρος, επιταχύνει τη σύγκλιση και περιορίζει την πιθανότητα αστάθειας κατά τη διάρκεια της μάθησης. Όπως φαίνεται στην μέθοδο forward (βλέπε Εικόνα 12), η κανονικοποίηση εφαρμόζεται στην έξοδο κάθε LSTM επιπέδου (μετά την συνάρτηση ενεργοποίησης tanh) και πριν αυτή τροφοδοτηθεί στο επόμενο επίπεδο του δικτύου. Δεν εφαρμόζεται δηλαδή στην αρχική, ακατέργαστη είσοδο που δέχεται κάθε LSTM για τους εσωτερικούς του υπολογισμούς (π.χ., για τις πύλες του). Αυτή η προσέγγιση επιλέχθηκε διότι τα LSTM επίπεδα έχουν την ικανότητα να διατηρούν εσωτερική κατάσταση (cell state) και να χειρίζονται πληροφορίες σε μεγάλες χρονικές ακολουθίες. Αν εφαρμοζόταν κανονικοποίηση των σημάτων εισόδου πριν από την επεξεργασία τους από τις πύλες του LSTM, θα μπορούσε να διαταράξει την εσωτερική δυναμική του LSTM, επηρεάζοντας την ικανότητά του να μαθαίνει μακροπρόθεσμες εξαρτήσεις. Αντίθετα, η κανονικοποίηση επιπέδου εφαρμόζεται στην έξοδο των LSTM για να διασφαλιστεί ότι οι έξοδοι θα είναι σταθεροποιημένες και να αποφευχθούν προβλήματα όπως οι εκρηκτικές ή φθίνουσες κλίσεις κατά τη διάρκεια της εκπαίδευσης.

2.3 Εργαλεία υλοποίησης

2.3.1 Σχετικά με την γλώσσα υλοποίησης

Η γλώσσα που θα επιλέξουμε για την υλοποίηση ενός Project είναι καθοριστική για διάφορους λόγους. Για την υλοποίηση της διπλωματικής εργασίας χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python. Η Python έχει διάφορα πεδία εφαρμογής. Είναι μία από τις κύριες γλώσσες προγραμματισμού στον τομέα της μηχανικής μάθησης και της τεχνητής νοημοσύνης. Κάποιοι από τους λόγους που την καθιστούν ιδανική σε αυτούς τους τομείς είναι η απλή και φιλική σύνταξή της σε σχέση με άλλες γλώσσες, όπου πολλοί αναφέρουν ότι μοιάζει με την καθομιλουμένη της αγγλικής γλώσσας, η μεγάλη κοινότητα που έχει η Python όπου υπάρχουν απαντήσεις σχεδόν σε κάθε ερώτημα και το ποιο σημαντικό είναι η πληθώρα βιβλιοθηκών και Frameworks όπου ειδικά για μηχανική μάθηση και τεχνητή νοημοσύνη είναι κομβικής σημασίας. Μας εξοικονομεί χρόνο (χρήμα) καθώς επιλέγοντας

κάποια/ες έτοιμες βιβλιοθήκη/ες έχουμε πρόσβαση σε διάφορες λειτουργικότητες χωρίς να χρειάζεται να γράψουμε από το μηδέν τον κώδικα.

2.3.2 Βιβλιοθήκες και πακέτα

Η βιβλιοθήκη pandas

Η pandas είναι μια ισχυρή βιβλιοθήκη για την επεξεργασία και την ανάλυση δεδομένων. Δημιουργεί δεδομένα σε μορφή πίνακα (DataFrame) και προσφέρει εύκολη και αποδοτική διαχείριση δεδομένων, όπως το διάβασμα των δεδομένων, την μετατροπή, ανάλυση χρονικών δεδομένων, φιλτράρισμα, επιλογή και κανονικοποίηση δεδομένων βάση της μέσης τιμής και της τυπικής απόκλισης των δεδομένων εκπαίδευσης. Η pandas επίσης παρέχει λειτουργικότητα για υπολογισμό τεχνικών δεικτών, οι οποίοι παρουσιάστηκαν σε προηγούμενες ενότητες.

Η βιβλιοθήκη numpy

Η numpy είναι βασική βιβλιοθήκη της python για αριθμητικούς υπολογισμούς. Κάποιες από τις λειτουργίες που μας παρέχει είναι η δημιουργία και ο χειρισμός πολυδιάστατων πινάκων δεδομένων, ο υπολογισμός βασικών αλλά και προχωρημένων αριθμητικών πράξεων (απόλυτες τιμές, λογαριθμικές, εκθετικές, τριγωνομετρικές συναρτήσεις κ.α.) όπως επίσης μας παρέχει εργαλεία για υπολογισμό βασικών στατιστικών μέτρων (μέσων όρων, τυπικών αποκλίσεων κ.α.).

Η βιβλιοθήκη scikit-learn.preprocessing

Η scikit-learn.preprocessing χρησιμοποιείται για κανονικοποίηση αλλά και κλιμάκωση των δεδομένων, για παράδειγμα πριν την εισαγωγή τους στο νευρωνικό δίκτυο. Αυτή η προετοιμασία βελτιώνει την απόδοση των αλγορίθμων μηχανικής μάθησης. Υπάρχουν διάφορες επιλογές για κανονικοποίηση των δεδομένων. Κάποιες από τις οποίες είναι:

- Z-score: Η StandardScaler της scikit-learn.preprocessing χρησιμοποιείται για κανονικοποίηση των δεδομένων με την μέθοδο Z-score, όπου μετασχηματίζει τα

δεδομένα ώστε να έχουν μέσο όρο 0 και τυπική απόκλιση 1. Η Z-score δείχνει πόσο απέχει μια τιμή από τον μέσο όρο των δεδομένων, εκφρασμένη σε μονάδες τυπικής απόκλισης (αν η τιμή της Z-score είναι 2, τότε απέχει 2 τυπικές αποκλίσεις πάνω από το μέσο όρο δεδομένων). Με αυτή τη μέθοδο τα δεδομένα ομαδοποιούνται γύρω από τον μέσο όρο και κανονικοποιούνται, ώστε όλα τα δεδομένα να έχουν την ίδια κλίμακα για να είναι συγκρίσιμα. Η κανονικοποίηση εξασφαλίζει ότι οι αριθμητικές τιμές των δεδομένων διατηρούνται σε ένα σταθερό εύρος, βοηθώντας τους αλγόριθμους να συγκλίνουν γρηγορότερα και πιο αξιόπιστα. Αυτό βοηθά στην καλύτερη απόδοση των αλγορίθμων.

Ο τύπος της Z-score είναι:

$$Z = \frac{X - \mu}{\sigma} \tag{2.23}$$

- X: είναι η αρχική τιμή
- μ: ο μέσος όρος των δεδομένων
- σ: η τυπική απόκλιση των δεδομένων.

- MinMax: Η κανονικοποίηση με MinMax, εξασφαλίζει ότι οι τιμές των δεδομένων παραμένουν εντός ενός σταθερού εύρους. Ειδικά σε ένα περιβάλλον συναλλαγών, αυτό είναι ιδιαίτερα χρήσιμο για τη διατήρηση της συνέπειας των τιμών και την αποφυγή ακραίων μεταβολών. Το σταθερό εύρος (συνήθως μεταξύ 0 και 1), δίνει την δυνατότητα συγκρίσεις των δεδομένων μεταξύ τους, διευκολύνοντας την ανάλυση και τη λήψη αποφάσεων. Οι αλγόριθμοι μηχανικής μάθησης μπορούν να συγκλίνουν γρηγορότερα και πιο αξιόπιστα, βελτιώνοντας την απόδοση του μοντέλου.

  Ο τύπος υπολογισμού είναι:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2.24}$$

  - X: η αρχική τιμή
  - X_{min}: Η ελάχιστη τιμή χαρακτηριστικού (High, Low, Close, Volume).
  - X_{max}: Η μέγιστη τιμή χαρακτηριστικού.

Η βιβλιοθήκη Torch

Η torch είναι μια ισχυρή βιβλιοθήκη που μας δίνει την δυνατότητα για δημιουργία και εκπαίδευση νευρωνικών δικτυών. Κάποια από τα βασικά χαρακτηριστικά της είναι:

- Διαχείριση Tensors: Παρέχει εργαλεία για τη δημιουργία και τον χειρισμό πολυδιάστατων πινάκων δεδομένων (tensors), παρόμοια με τα arrays της numpy που αναφέρθηκε παραπάνω.
- Νερωνικά δίκτυα (nn): Περιλαμβάνει κλάσεις και συναρτήσεις για τη δημιουργία και τον χειρισμό νευρωνικών δικτύων.
- Αυτόματη διαφοροποίηση (Automatic differentiation): Η torch υποστηρίζει την αυτόματη διαφορική, που είναι σημαντική για την εκπαίδευση νευρωνικών δικτύων μέσω της οπισθοδιάδοσης(backpropagation). Η αυτόματη διαφοροποίηση επιτρέπει τον αυτόματο υπολογισμό των παραγώγων (gradients) των συναρτήσεων απώλειας (loss function) σε σχέση με τις παραμέτρους του μοντέλου, διευκολύνοντας τη διαδικασία της οπισθοδιάδοσης. Η χρήση συναρτήσεων απώλειας όπως η MSELoss για τον υπολογισμό της διαφοράς μεταξύ των προβλέψεων και των πραγματικών τιμών, σε συνδυασμό με την οπισθοδιάδοση για την ενημέρωση των βαρών, αποτελεί κρίσιμη διαδικασία για την εκπαίδευση και την βελτιστοποίηση του μοντέλου μηχανικής μάθησης.
- Υποστήριξη CUDA: Υποστηρίζει την εκτέλεση σε GPU, κάνοντας την εκπαίδευση των μοντέλων ταχύτερη.
- Βελτιστοποίηση: Μπορούν να χρησιμοποιηθούν διάφοροι αλγόριθμοι για τη βελτιστοποίηση παραμέτρων των νευρωνικών δικτύων, όπως ο Adam optimizer, που υπολογίζει και προσαρμόζει τους ρυθμούς μάθησης για κάθε παράμετρο, παρακολουθώντας τη μέση τιμή (mean) και τη διασπορά (variance) των εκτιμήσεων των gradient, βελτιώνοντας έτσι την ακρίβεια των προσαρμογών. Η χρήση τέτοιων αλγορίθμων εξασφαλίζει αποδοτική και σταθερή εκπαίδευση του νευρωνικού δικτύου και του πράκτορα.

2.3.3 Λογισμικά

IDE Pycharm

Το Pycharm της Jetbrain είναι ένα από τα δημοφιλέστερα IDE για την γλώσσα προγραμματισμού Python. Η χρήση του για την ανάπτυξη προγραμμάτων έχει πολλά πλεονεκτήματα λόγω των πλούσιων χαρακτηριστικών και εργαλείων που προσφέρει εξατομικεύοντας το περιβάλλον του με διάφορους τρόπους για την βελτίωση της συγγραφής κώδικα. Κάποια από τα πλεονεκτήματα χρήσης του Pycharm είναι:

- Η έξυπνη συμπλήρωση κώδικα, βοηθώντας στην ταχύτερη και ακριβέστερη συγγραφή κώδικα μέσω της πρόβλεψης και συμπλήρωσης των συναρτήσεων και των μεταβλητών.
- Η δυνατότητα απασφαλμάτωσης (Debugging) που μας παρέχει, όπου μπορούμε να ορίσουμε σημεία διακοπής (breakpoints), βοηθώντας στον εντοπισμό και την επίλυση προβλημάτων στον κώδικα.
- Η υποστήριξη εικονικού περιβάλλοντος, όπως για παράδειγμα μέσω του Anaconda, εξασφαλίζοντας την αυτονομία κάθε project.

Anaconda

Το Anaconda είναι μια διανομή (distribution) της python, για την επιστήμη των δεδομένων και την ανάλυση δεδομένων. Μας παρέχει ένα περιβάλλον που περιλαμβάνει προ εγκατεστημένα πακέτα και εργαλεία για στατιστική ανάλυση, μηχανική μάθηση και τεχνητή νοημοσύνη, που είναι απαραίτητα για την ανάπτυξη και εκτέλεση προγραμμάτων. Επίσης διευκολύνει την διαδικασία της εγκατάσταση νέων πακέτων που μπορεί να χρειάζονται για την ανάπτυξη του προγράμματος, μέσω του γραφικού περιβάλλοντος (Anaconda Navigator).

Είναι διαθέσιμο για διάφορα υπολογιστικά συστήματα (Windows, Linux, MacOs) δίνοντας στους χρήστες την ελευθερία να εργάζονται σε διαφορετικά υπολογιστικά συστήματα, χωρίς να υπάρχει πρόβλημα συμβατότητας.

Ένα από τα σημαντικότερα πλεονεκτήματα του Anaconda, είναι η δυνατότητα δημιουργίας πολλαπλών περιβαλλόντων, επιτρέποντας την αυτόνομη χρήση πακέτων και βιβλιοθηκών,

χωρίς να υπάρχει πρόβλημα μεταξύ διαφορετικών projects που μπορεί να χρησιμοποιούν διαφορετικές εκδόσεις.

3 Κεφάλαιο

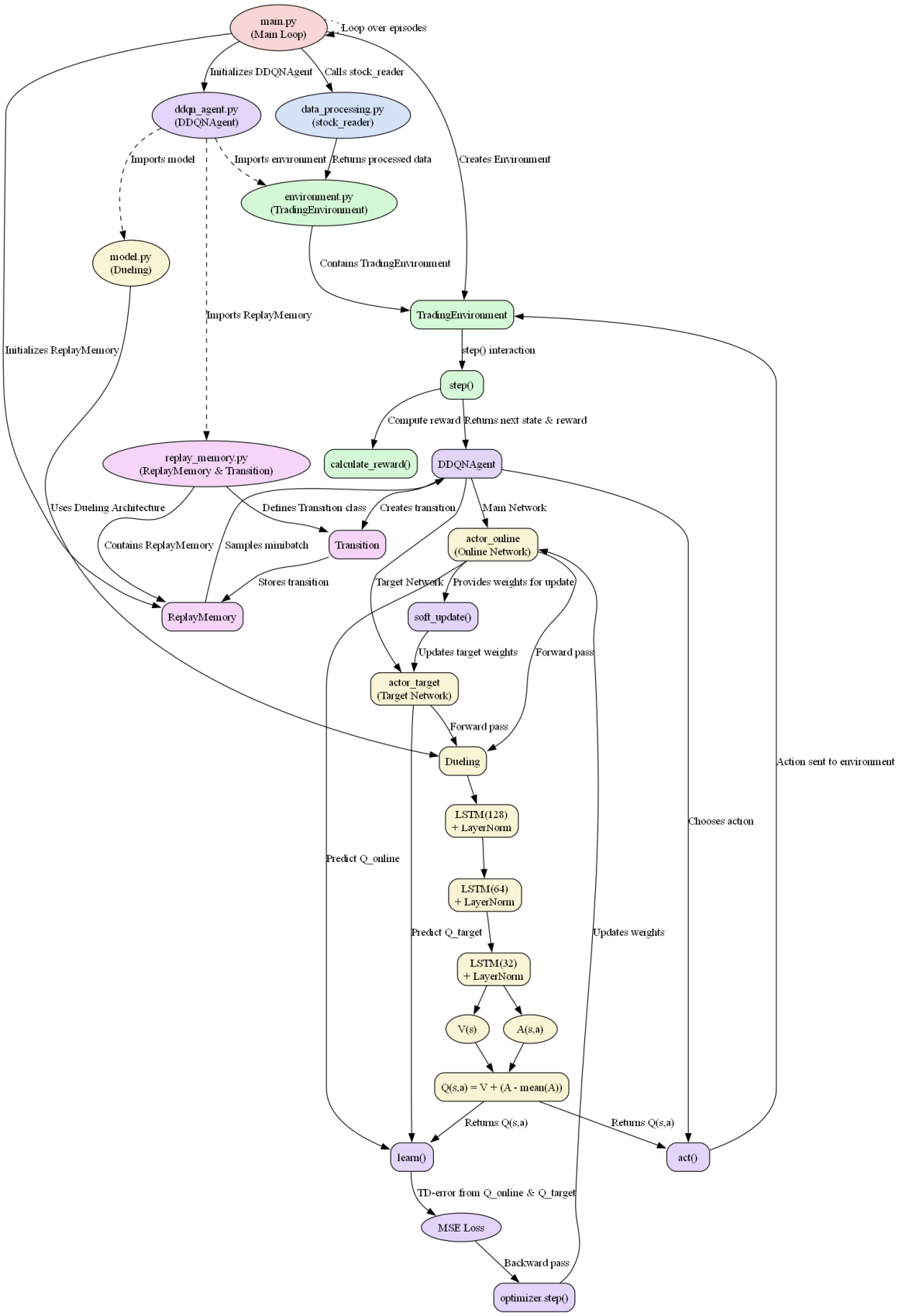
3.1 Υλοποίηση αλγορίθμου

Μετά τον σχεδιασμό του αλγορίθμου, η διαδικασία υλοποίησης αποδείχθηκε ιδιαίτερα απαιτητική, καθώς έπρεπε να ληφθούν υπόψη πολλοί παράγοντες που επηρεάζουν την απόδοσή του. Ένα από τα πιο κρίσιμα σημεία ήταν η διαμόρφωση της συνάρτησης ανταμοιβής, καθώς ακόμα και μικρές αλλαγές στη λογική της μπορούσαν να οδηγήσουν σε εντελώς διαφορετική συμπεριφορά του μοντέλου.

Επιπλέον, η επιλογή κατάλληλων τιμών στις παραμέτρους ήταν καθοριστική για την επιτυχία της εκπαίδευσης. Παρατηρήσαμε ότι η παραμετροποίηση του μοντέλου είναι μια λεπτή διαδικασία, όπου η παραμικρή τροποποίηση μπορεί να επηρεάσει σημαντικά την απόδοση.

3.1.1 Flowchart του συστήματος και ψευδοκώδικας

Πριν προχωρήσουμε στην παρουσίαση του κώδικα, για καλύτερη κατανόηση της συνολική λειτουργίας του συστήματος, κρίθηκε απαραίτητο να αναπαρασταθεί γραφικά η ροή των δεδομένων (βλέπε Εικόνα 13: Stock Trading Agent flowchart) και των κλήσεων μεταξύ των βασικών modules του προγράμματος. Το παρακάτω διάγραμμα ροής απεικονίζει τη «συνεργασία» μεταξύ των επιμέρους αρχείων, κλάσεων και συναρτήσεων του αλγορίθμου που θα παρουσιαστούν μετέπειτα, ώστε να δοθεί μία εικόνα της δομής και του τρόπου λειτουργίας του DDDQN. Επίσης παρατίθεται ο ψευδοκώδικας (βλέπε Ψευδοκώδικας 4) του συστήματος, υλοποιημένος κατά τέτοιο τρόπο ώστε να καταγράφει τις μεθόδους και τις σχέσεις που χρησιμοποιούνται στη διαδικασία μάθησης και λήψης αποφάσεων του προγράμματος.



Εικόνα 13: Stock Trading Agent flowchart DDDQN

1) Initialize Parameters:

Initialize primary network Q_θ and target network $Q_{\theta'}$ with random weights

Set Hyperparameters: learning rate (a), discount factor (γ), target update(τ),epsilon decay (ε_{start}, ε_{end}, ε_{decay}),

Initialize $\varepsilon \leftarrow \varepsilon_{start}$

Define action space A={-1, 1, 0} (short, long, hold)

Load historical stock market data using stock_reader()

2) Training and Trading Loop (for each episode):

3) Initialize date pointer $t_0 \leftarrow$ INITIAL_DATE

4) Training Phase (Intraday)

5) Get initial state $s_t \leftarrow$ env.get_state()

6) While not done:

7) Select action a_t using epsilon-greedy policy via agent.act():

$$a_t = \begin{cases} random\ action & with\ probability\ \varepsilon \\ \arg\max_a Q_\theta(s_t, a) & otherwise \end{cases}$$

8) Execute action a_t, using env.step(a_t):

9) Observe next state s_{t+1} reward $r_t \leftarrow$ env.calculate_reward(), done flag d_t

10) Store transition (s_t, a_t, r_t, $_{st+1}$, d_t)in ReplayMemory D

11)

12) If done:

13) Close all position and calculate portfolio finalize_trade()

14) Learning Step

15) If t > LEARN_AFTER and every LEARN_EVERY steps:

16) Sample mini-batch B ← D.sample()

17) For each (s, a, r, s′, d) ∈ B:

18) Compute target Q-value

$$y = r + \gamma \cdot Q_{target}\left(s', \underset{a'}{argmax} Q_{online}(s', a')\right)$$

19) Compute loss:

$$L = \frac{1}{N}\sum\left(y - Q_{online}(s, a)\right)^2$$

20) Perform gradient descent:

21) Backpropagate and apply optimizer.step() to update θ

22) Update target network parameters using soft_update():

$$\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$$

23) Set $s_t \leftarrow s_{t+1}$

24) Decay $\varepsilon \leftarrow \max(\varepsilon_{end}, \varepsilon * \varepsilon_{decay})$

25) Trading Phase (Next Day)

26) Move to next date t′←t+1

27) Get initial state $s_t \leftarrow$trade_env.get_state()

28) While not done:

29) Select action $a_t \leftarrow$ agent.act(s_t, ε)

```
30)       Execute action a_t
31)       Observe reward r, next state s_t+1, done flag
32)       Accumulate trading performance metrics(PnL, etc.)
33)     If done:
34)      Close all position and calculate portfolio finalize_trade()

35)   If episode ends
36)     Reset train environment: env.reset()
37)     Reset trade environment: trade_env.reset()
38)   Save trained model Q_θ and log metrics
```

Ψευδοκώδικας 4: Stock Trading Agent flowchart DDDQN

3.1.2 Υλοποίηση του Dueling DDQN αλγορίθμου (κώδικας)

Ένα βασικό στοιχείο της υλοποίησης είναι ο τρόπος με τον οποίο τα δεδομένα περνάνε στο μοντέλο. Όπως ήδη αναφέραμε, δεν χρησιμοποιούμε απευθείας τα δεδομένα εισόδου (raw data), αλλά τα μετασχηματίζουμε μέσω τεχνικών δεικτών (technical indicators)(βλέπε Εικόνα 14 και Εικόνα 15). Τα οφέλη αυτής της προσέγγισης τα έχουμε ήδη αναφέρει όπως επίσης και την πληροφορία που παρέχει στο μοντέλο ο κάθε δείκτης.

```python
# Calculate pct_change
for i in self.timeframes:
    if len(self.data) >= i:
        self.data[f"v-{i}"] = self.data['volume'].pct_change(i)
        self.data[f"r-{i}"] = self.data['close'].pct_change(i)
    else:
        self.data[f"v-{i}"] = np.nan
        self.data[f"r-{i}"] = np.nan

# Calculate rolling volatility
for i in [2, 5, 10, 20, 40]:
    if len(self.data) >= i:
        self.data[f'sig-{i}'] = np.log(1 + self.data["r-1"]).rolling(i).std()
    else:
        self.data[f'sig-{i}'] = np.nan

# Calculate Bollinger Bands
self.bollinger_lback = 10
if len(self.data) >= self.bollinger_lback:
    self.data["bollinger"] = self.data["r-1"].ewm(self.bollinger_lback).mean()
    self.data["low_bollinger"] = self.data["bollinger"] - 2 * self.data["r-1"].rolling(self.bollinger_lback).std()
    self.data["high_bollinger"] = self.data["bollinger"] + 2 * self.data["r-1"].rolling(self.bollinger_lback).std()
else:
    self.data["bollinger"] = np.nan
    self.data["low_bollinger"] = np.nan
    self.data["high_bollinger"] = np.nan

# Calculate RSI
self.rsi_lb = 5
if len(self.data) >= self.rsi_lb:
    self.pos_gain = self.data["r-1"].where(self.data["r-1"] > 0, 0).ewm(self.rsi_lb).mean()
    self.neg_gain = self.data["r-1"].where(self.data["r-1"] < 0, 0).ewm(self.rsi_lb).mean()
    self.rs = np.abs(self.pos_gain / self.neg_gain)
    self.data["rsi"] = 100 * self.rs / (1 + self.rs)
else:
    self.data["rsi"] = np.nan

# Calculate MACD
self.data["macd_lmw"] = self.data["r-1"].ewm(span=20, adjust=False).mean()
self.data["macd_smw"] = self.data["r-1"].ewm(span=12, adjust=False).mean()
self.data["macd_bl"] = self.data["r-1"].ewm(span=9, adjust=False).mean()
self.data["macd"] = self.data["macd_smw"] - self.data["macd_lmw"]
self.data["macd_signal"] = self.data["macd"].ewm(span=9, adjust=False).mean()
self.data["macd_histogram"] = self.data["macd"] - self.data["macd_signal"]

# Calculate STC
if len(self.data) >= 9:
    macd_range = self.data["macd"].rolling(window=9).max() - self.data["macd"].rolling(window=9).min()
    self.data["stc"] = 100 * (self.data["macd"] - self.data["macd"].rolling(window=9).min()) / macd_range
    self.data["stc_smoothed"] = self.data["stc"].rolling(window=3).mean()
else:
    self.data["stc"] = np.nan
    self.data["stc_smoothed"] = np.nan
```

Εικόνα 14: Technical indicators 1

```python
# Calculate TR
if len(self.data) > 1:
    self.data['HL'] = self.data['high'] - self.data['low']
    self.data['HC'] = abs(self.data['high'] - self.data['close'].shift(-1))
    self.data['LC'] = abs(self.data['high'] - self.data['close'].shift(-1))
    self.data['TR'] = self.data[['HL', 'HC', 'LC']].max(axis=1)
    self.data.drop( labels: ['HL', 'HC', 'LC'], axis=1, inplace=True)
else:
    self.data['TR'] = np.nan

# Calculate Stochastic Oscillator
if len(self.data) >= 14:
    self.data['Lowest_Low'] = self.data['low'].rolling(window=14).min()
    self.data['Highest_High'] = self.data['high'].rolling(window=14).max()
    self.data['%K'] = ((self.data['close'] - self.data['Lowest_Low']) / (
                self.data['Highest_High'] - self.data['Lowest_Low'])) * 100
    self.data['%D'] = self.data['%K'].rolling(window=3).mean()
else:
    self.data['Lowest_Low'] = np.nan
    self.data['Highest_High'] = np.nan
    self.data['%K'] = np.nan
    self.data['%D'] = np.nan

# Calculate ATR
self.atr_period = 11
if len(self.data) >= self.atr_period:
    self.data['ATR'] = self.data['TR'].rolling(window=self.atr_period).mean()
else:
    self.data['ATR'] = np.nan

# Calculate next_state_return
if len(self.data) > 1:
    self.data['next_state_return'] = self.data['close'].pct_change().shift(-1)
else:
    self.data['next_state_return'] = np.nan

# Exclude non-numeric columns from normalization
columns_to_normalize = self.data.select_dtypes(include=[np.float64, np.int64]).columns
self.data[columns_to_normalize] = self.scaler.fit_transform(self.data[columns_to_normalize])

# Fill NaN values with zeros
self.data.fillna( value: 0, inplace=True)

self.train_days = self.data.loc[self.train_start:self.train_end].copy()
self.trade_days = self.data.loc[self.trade_start:self.trade_end].copy()

# Exclude non-numeric columns from mean and std calculations
numeric_columns = self.train_days.select_dtypes(include=[np.float64, np.int64]).columns

self.train_mean = self.train_days[numeric_columns].mean()
self.train_std = self.train_days[numeric_columns].std()

for column in numeric_columns:
    self.train_days[f"{column}_norm"] = (self.train_days[column] - self.train_mean[column]) / self.train_std[
        column]
    self.trade_days[f"{column}_norm"] = (self.trade_days[column] - self.train_mean[column]) / self.train_std[
        column]
```

Εικόνα 15: Technical indicators 2

Αφού σχεδιάστηκε η αρχιτεκτονική του αλγορίθμου, την οποία αναφέραμε στο προηγούμενο κεφάλαιο, η επόμενη πρόκληση ήταν η υλοποίηση του πράκτορα, ο οποίος είναι υπεύθυνος για τη λήψη αποφάσεων στο περιβάλλον. Η ανάπτυξη του πράκτορα, όπως ήδη έχουμε αναφέρει, βασίστηκε στη μέθοδο Double Deep Q-Network (DDQN)(βλέπε Εικόνα 16).

Ένα κρίσιμο σημείο στην υλοποίηση του Dueling DDQN πράκτορα είναι ότι χρησιμοποιούμε δύο αντίγραφα του ίδιου ακριβώς νευρωνικού δικτύου (Dueling DDQN). Το online network και το target network.

Το online network είναι υπεύθυνο για την επιλογή δράσεων κατά τη διάρκεια της εκπαίδευσης. Ο πράκτορας παρατηρεί την τρέχουσα κατάσταση s. Το online network λαμβάνει ως είσοδο την κατάσταση s και επιστρέφει ένα διάνυσμα που περιέχει τις εκτιμώμενες Q-τιμές για όλες τις πιθανές δράσεις a. Ο πράκτορας επιλέγει τη δράση με την υψηλότερη Q-τιμή (εκμετάλλευση, exploitation) με πιθανότητα $(1-\varepsilon)$ ή επιλέγει τυχαία μια δράση (εξερεύνηση, exploration) με πιθανότητα ε (ε-greedy policy). Αυτή η προσέγγιση βοηθάει το μοντέλο να ανακαλύψει νέες στρατηγικές αντί να βασίζεται αποκλειστικά σε προηγούμενες εμπειρίες.

Μετά την εκτέλεση της δράσης, ο πράκτορας παρατηρεί την νέα κατάσταση s' και την ανταμοιβή r. Για να ενημερωθεί το online network, πρέπει να υπολογιστεί η target Q-τιμή, δηλαδή το αναμενόμενο κέρδος αν ακολουθήσει την καλύτερη στρατηγική από εκείνο το σημείο και μετά. Στην προσέγγιση Double DQN, για τον υπολογισμό αυτής της target Q-τιμής, η επιλογή της καλύτερης μελλοντικής δράσης γίνεται από το online network, ενώ η αξιολόγηση της αξίας αυτής της δράσης γίνεται από το target network. Συγκεκριμένα, η target Q-τιμή υπολογίζεται ως εξής:

$$y = r + \gamma \cdot Q_{target}\left(s', arg\max_{a'} Q_{online}(s', a')\right) \quad (3.1)$$

Το online δίκτυο χρησιμοποιείται για να προσδιορίσει ποια δράση $\left(arg\max_{a'} Q_{online}(s', a')\right)$ θεωρείται η καλύτερη στην επόμενη κατάσταση s'. Στη συνέχεια, το target δίκτυο παρέχει την εκτίμηση της αξίας για αυτή ακριβώς την επιλεγμένη δράση στην κατάσταση s'. Αυτή η target Q-τιμή συγκρίνεται με την Q-τιμή που είχε προβλέψει το online network για την αρχική κατάσταση και δράση $\left(Q_{online}(s, a)\right)$, και το

σφάλμα αυτής της διαφοράς χρησιμοποιείται για να εκπαιδευτεί το online network. Ο διαχωρισμός της επιλογής δράσης από την αξιολόγησή της είναι ο μηχανισμός που βοηθά τον Double DQN να μειώσει το πρόβλημα της υπερεκτίμησης των Q-τιμών.

Η χρήση του target network, είναι μια τεχνική που βελτιώνει τη σταθερότητα της εκπαίδευσης. Αντί το μοντέλο να βασίζεται αποκλειστικά στο online δίκτυο, με την χρήση του target network αποτρέπεται η συνεχής αλλαγή των στόχων εκπαίδευσης, μειώνοντας την αστάθεια και επιταχύνοντας τη σύγκλιση του μοντέλου.

```
class DDQNAgent():
    def __init__(self, actor_net, memory):
        self.actor_online = actor_net(STATE_SPACE, ACTION_SPACE).to(DEVICE)
        self.actor_target = actor_net(STATE_SPACE, ACTION_SPACE).to(DEVICE)
        self.actor_target.load_state_dict(self.actor_online.state_dict())
        self.actor_target.eval()

        # Replay memory for experience replay
        self.memory = memory

        # Loss function and optimizer
        self.actor_criterion = nn.MSELoss()
        self.actor_op = optim.Adam(self.actor_online.parameters(), lr=LR_DQN)

        self.t_step = 0
```

Εικόνα 16: DDQN Agent class

Η διαδικασία εκμάθησης του πράκτορα πραγματοποιείται μέσω της μεθόδου learn (την ξανά παραθέτουμε για πληρότητα) (βλέπε Εικόνα 17), η οποία εκτελείται περιοδικά κατά τη διάρκεια της εκπαίδευσης. Η εκπαίδευση δεν συμβαίνει σε κάθε χρονική στιγμή, αλλά μόνο όταν ο αριθμός των αποθηκευμένων εμπειριών στο Replay Memory ξεπεράσει ένα προκαθορισμένο κατώφλι (LEARN_AFTER), όπως έχει ήδη αναφερθεί. Επιπλέον, η εκπαίδευση πραγματοποιείται ανά τακτά διαστήματα (LEARN_EVERY), ώστε να αποφεύγεται η υπερβολική ενημέρωση του δικτύου, η οποία μπορεί να οδηγήσει σε αστάθεια.

Αρχικά, επιλέγεται ένα τυχαίο δείγμα (batch) από τις αποθηκευμένες εμπειρίες. Τα δεδομένα που εξάγονται μετατρέπονται σε tensors, ώστε να μπορούν να επεξεργαστούν από το νευρωνικό δίκτυο. Στη συνέχεια, χρησιμοποιείται η προσέγγιση του Double DQN για την ενημέρωση των Q-τιμών. Αντί να επιλέγεται η καλύτερη δράση από το target network,

όπως συμβαίνει στο κλασικό DQN που έχουμε αναφέρει, χρησιμοποιείται το online network για την επιλογή της δράσης με τη μεγαλύτερη εκτιμώμενη Q-τιμή (best_actions_online). Στη συνέχεια, το target network εκτιμά την αξία αυτής της δράσης (next_state_values), εξασφαλίζοντας μια πιο αξιόπιστη εκτίμηση της target Q-τιμής.

Στον αλγόριθμό μας, χρησιμοποιήσαμε τη συνάρτηση MSELoss για τη μέτρηση της διαφοράς μεταξύ της target Q-τιμής και της Q-τιμής που έχει προβλέψει το online network. Η απώλεια αυτή ορίζεται ως το μέσο τετραγωνικό σφάλμα, όπου N είναι το μέγεθος του mini-batch από το replay buffer:

$$L = \frac{1}{N}\sum\left(y - Q_{online}(s,a)\right)^2 \qquad (3.2)$$

Παράλληλα, χρησιμοποιούμε τον Adam optimizer για την ενημέρωση των βαρών του δικτύου, μειώνοντας προοδευτικά το σφάλμα κατά τη διάρκεια της εκπαίδευσης, ώστε οι Q-τιμές να γίνονται πιο ακριβείς.

```python
def learn(self):
    if len(self.memory) <= LEARN_AFTER:
        return 0

    # Sample experiences from memory at defined intervals
    if self.t_step > LEARN_AFTER and self.t_step % LEARN_EVERY == 0:

        batch = self.memory.sample()

        # Convert batch to tensors
        states = T.from_numpy(np.vstack([t.States for t in batch])).float().to(DEVICE)
        actions = T.from_numpy(np.vstack([t.Actions for t in batch])).long().to(DEVICE)
        rewards = T.from_numpy(np.vstack([t.Rewards for t in batch])).float().to(DEVICE)
        next_states = T.from_numpy(np.vstack([t.NextStates for t in batch])).float().to(DEVICE)
        dones = T.from_numpy(np.vstack([t.Dones for t in batch])).float().to(DEVICE)

        # Compute target Q-values using Double DQN approach
        best_actions_online = self.actor_online(next_states).argmax(1).unsqueeze(1)
        next_state_values = self.actor_target(next_states).gather(1, best_actions_online)
        y = rewards + (1 - dones) * GAMMA * next_state_values
        state_values = self.actor_online(states).gather(1, actions.long())

        # Compute loss and optimize network
        actor_loss = self.actor_criterion(y, state_values)
        self.actor_op.zero_grad()
        actor_loss.backward()
        self.actor_op.step()

        # Update target network periodically
        if self.t_step % UPDATE_EVERY == 0:
            self.soft_update(self.actor_online, self.actor_target)
```

Εικόνα 17: Learn method

Για να αποφύγουμε την απότομη αλλαγή των Q-τιμών και να διατηρήσουμε τη σταθερότητα στην εκπαίδευση, δεν ενημερώνουμε άμεσα τις παραμέτρους του target network σε κάθε βήμα Αν το αντικαθιστούσαμε πλήρως σε κάθε ενημέρωση, οι στόχοι εκπαίδευσης θα άλλαζαν πολύ γρήγορα, οδηγώντας σε αστάθεια και πιθανή απόκλιση. Αντίθετα, χρησιμοποιούμε την τεχνική soft update (βλέπε Εικόνα 18), όπου σε κάθε βήμα εκπαίδευσης, το target network ενημερώνεται σταδιακά μέσω της σχέσης:

$$\theta_{target} = \tau \cdot \theta_{online} + (1 - \tau) \cdot \theta_{target} \quad (3.3)$$

όπου το τ καθορίζει πόσο γρήγορα προσαρμόζεται το target network στις νέες εκτιμήσεις. Με αυτόν τον τρόπο το target network διατηρεί πληροφορία από προηγούμενες εκτιμήσεις για κάποιο διάστημα, λειτουργώντας ως σταθερό σημείο αναφοράς στον υπολογισμό της απώλειας όπως θα δούμε παρακάτω, μειώνοντας έτσι τη μεταβλητότητα των ενημερώσεων και βοηθά ως προς τη σύγκλιση του αλγορίθμου.

```
def soft_update(self, local_model, target_model, tau=TAU):

    for target_param, local_param in zip(target_model.parameters(), local_model.parameters()):
        target_param.data.copy_(tau * local_param.data + (1.0 - tau) * target_param.data)
```

Εικόνα 18: Soft method

```
def act(self, state, eps=0.):
    """Select an action using epsilon-greedy strategy."""
    self.t_step += 1

    state = T.from_numpy(state).float().to(DEVICE).view(1, 1, -1)

    self.actor_online.eval()
    with T.no_grad():
        actions = self.actor_online(state) # Forward pass
    self.actor_online.train()

    # Epsilon-greedy action selection
    if random.random() > eps:
        act = np.argmax(actions.cpu().data.numpy())
    else:
        act = random.choice(np.arange(ACTION_SPACE))
    return int(act)
```

Εικόνα 19: Act method

Η μέθοδος act (βλέπε Εικόνα 19) επιλέγει την επόμενη δράση που θα εκτελέσει ο πράκτορας, λαμβάνοντας υπόψη την τρέχουσα κατάσταση και το exploration rate ε. Η επιλογή δράσης βασίζεται στην ε-greedy στρατηγική, η οποία επιδιώκει την ισορροπία μεταξύ εξερεύνησης (exploration) και εκμετάλλευσης (exploitation). Αν η τυχαία τιμή είναι μικρότερη από ε, τότε επιλέγεται μια τυχαία δράση (exploration), επιτρέποντας στον πράκτορα να εξερευνήσει νέες στρατηγικές, αντίθετα αν η τυχαία τιμή είναι μεγαλύτερη από ε, τότε επιλέγεται η δράση με τη μέγιστη Q-τιμή (exploitation) που επιστρέφει το online network, ακολουθώντας την πολιτική που θεωρεί βέλτιστη με βάση τις μέχρι τώρα εμπειρίες του. Αυτή η προσέγγιση βοηθάει τον πράκτορα να αποφεύγει τοπικά βέλτιστα σημεία και να ανακαλύπτει στρατηγικές που μπορεί να οδηγήσουν σε υψηλότερες συνολικές ανταμοιβές.

```python
class TradingEnvironment:

    def __init__(self, asset_data, bank=CAPITAL, trans_coef=COST, position_limit=POSITION_LIMIT_COEF, threshold=THRESHOLD, store_flag=1, max_steps=MAX_STEPS):
        # Initializing trading parameters
        self.pnl = bank
        self.portfolio = bank
        self.long_positions = []
        self.short_positions = []
        self.closed_positions = []
        self.position_limit = position_limit
        self.trans_coef = trans_coef
        self.bank = bank
        self.max_steps = max_steps
        self.step_count = 0
        self.prev_act = 0
        self.returns = []
        self.prev_pnl = bank
        self.loss_threshold = threshold
        self.profit_threshold = threshold

        ### data variables
        self.asset_data = asset_data
        self.terminal_idx = len(self.asset_data) - 1

        ### pointers, actions, rewards
        self.pointer = 0
        self.next_return, self.current_state = 0, None
        self.current_act = 0
        self.reward_offset = 0

        assert len(self.asset_data) > 0, "asset_data is empty!"
        assert not self.asset_data.isnull().values.any(), "Attention! There are NaNs in asset_data!"

        if self.pointer >= len(self.asset_data): raise IndexError(
            "The index is outside the bounds of the asset_data DataFrame")

        self.current_price = self.asset_data.iloc[self.pointer, :]['close']
        self.done = False
```

Εικόνα 20: Trading Environment class

Μετά τη σχεδίαση του αλγορίθμου και την σχεδίαση το πράκτορα, το επόμενο βήμα ήταν η ανάπτυξη του περιβάλλοντος (βλέπε Εικόνα 20) και της συνάρτησης ανταμοιβής. Όπως έχουμε αναφέρει, το περιβάλλον αναπαριστά το πλαίσιο μέσα στο οποίο λαμβάνει δράση ο πράκτορας, ενώ η συνάρτηση ανταμοιβής καθορίζει τη λογική αξιολόγησης κάθε ενέργειας. Το περιβάλλον που υλοποιήσαμε προσομοιώνει μια χρηματοοικονομική αγορά. Ο πράκτορας καλείται να πάρει αποφάσεις σε ένα συνεχώς μεταβαλλόμενο περιβάλλον, βασιζόμενο σε οικονομικούς δείκτες και ιστορικά δεδομένα.

Το περιβάλλον περιλαμβάνει βασικές οικονομικές παραμέτρους όπως το αρχικό κεφάλαιο (bank), το κόστος συναλλαγών (trans_coef), το όριο θέσεων (position_limit) και το όριο κέρδους/ζημιάς (threshold). Ορίζεται επίσης ο μέγιστος αριθμός βημάτων (max_steps), που όπως θα δούμε παρακάτω θα είναι ένα από τα κριτήρια ώστε να κλείσει μία ανοιχτή θέση. Η λογική που αναπτύξαμε επιτρέπει την διατήρηση μέχρι ενός συγκεκριμένου αριθμού (π.χ 3) ταυτόχρονων ανοιχτών θέσεων, οι οποίες δεν περιορίζονται ως προς πόσες long ή short θέσεις θα είναι ταυτόχρονα ανοιχτές, ή ότι πρέπει να κλείσει μία long για να ανοίξει μία short ή αντίστροφα, αλλά αποθηκεύονται σε λίστες και όταν θα πληρούν τα κριτήρια τότε θα κλείνουν. Κατά την αρχικοποίηση, διαμορφώνονται μεταβλητές όπως, PnL για συνολικά κέρδη/ζημιές, portfolio για την τρέχουσα αξία του χαρτοφυλακίου, long_position και short_position για καταγραφή των ανοιχτών θέσεων αγοράς και πώλησης αντίστοιχα, closed_positions κλειστές θέσεις, pointer όπου είναι δείκτης που δείχνει στην ιστορική σειρά των δεδομένων και την μεταβλητή done ώστε να δηλώνει αν το επεισόδιο έχει τερματίσει.

```python
def get_state(self):
    state = []
    observation = ['r-1', 'r-2', 'r-5', 'r-10', 'r-20', 'r-40',
                   'v-1', 'v-2', 'v-5', 'v-10', 'v-20', 'v-40',
                   'sig-2', 'sig-5', 'sig-10', 'sig-20', 'sig-40',
                   'bollinger', 'low_bollinger', 'high_bollinger',
                   'rsi', 'macd_lmw', 'macd_smw', 'macd_bl', 'macd',
                   'macd_signal', 'macd_histogram', 'stc', 'stc_smoothed',
                   'TR', '%K', '%D', 'ATR']

    observation = [obs + '_norm' for obs in observation]

    long_positions_sign = sum([1 if pos.size > 0 else 0 for pos in self.long_positions])
    short_positions_sign = sum([1 if pos.size < 0 else 0 for pos in self.short_positions])

    port_state = [
        self.pnl / self.bank,
        self.portfolio / self.pnl,
        (long_positions_sign - short_positions_sign) * self.current_price / self.bank,
        self.prev_act
    ]

    for column in observation:
        value = self.asset_data.loc[self.asset_data.index[self.pointer], column]
        if isinstance(value, (float, int)) and pd.isna(value):
            print(f"Warning: NaN detected in {column}, replacing with 0.")
            value = 0
        if isinstance(value, pd.Series):
            value = value.iloc[0]
        state.append(value)

    state.extend(port_state)

    next_ret = self.asset_data['next_state_return'].iloc[self.pointer]

    return next_ret, state
```

Εικόνα 21: Get state method

Η get_state (βλέπε Εικόνα 21) επιστρέφει την τρέχουσα κατάσταση του περιβάλλοντος, καθώς και την αναμενόμενη απόδοση του επόμενους βήματος. Η κατάσταση περιλαμβάνει τεχνικούς δείκτες όπως (RSI, MACD κτλ), στοιχεία θέσεων που δείχνουν τη δραστηριότητα στις long και short θέσεις, καθώς και οικονομικά δεδομένα που σχετίζονται με το PnL και την αξία του χαρτοφυλακίου. Η get_state είναι καθοριστικής σημασίας για τη διασύνδεση του περιβάλλοντος με τον αλγόριθμο ενισχυτικής μάθησης, παρέχοντας πληροφορίες που απαιτείται για την αξιολόγηση της κατάσταση και τη λήψη αποφάσεων.

```python
def step(self, action):
    # Execute a trading step
    self.loss_threshold = max(self.portfolio * 0.1, THRESHOLD)
    self.profit_threshold = max(self.portfolio * 0.2, THRESHOLD)

    self.current_act = action
    assert self.pointer < len(self.asset_data), "Pointer out of bounds!"
    self.current_price = self.asset_data.iloc[self.pointer, :]['close']

    # Calculate the reward for the current action
    self.current_reward = self.calculate_reward()

    assert not np.isnan(self.current_reward), f"Reward is NaN on step {self.step_count}"

    # Update pointer, step count, and fetch the next state
    self.pointer += 1
    self.step_count += 1
    self.next_return, self.current_state = self.get_state()
    self.done = self.check_terminal()

    # Store the previous action for consistency
    self.prev_act = self.current_act

    # Terminate the episode if portfolio drops below a safety threshold
    if self.portfolio < 0.7 * self.bank:
        print(f"Portfolio dropped below 70% of initial capital: {self.portfolio}")
        self.done = True

    # Store data if the storage flag is enabled
    if not self.done and self.store_flag:
        self.store["action_store"].append(self.current_act)
        self.store["reward_store"].append(self.current_reward)
        self.store["close_price"].append(self.current_price)
        info = self.store
    else:
        info = None

    # Finalize all remaining positions if the episode is done
    if self.done:
        self.reward_offset = 0
        for position in self.long_positions[:]:
            self.finalize_trade(position)
        for position in self.short_positions[:]:
            self.finalize_trade(position)

        self.long_positions = []
        self.short_positions = []
        self.store["position"].append(0)
        self.store["pnl"].append(self.pnl)
        self.store["portfolio"].append(self.portfolio)

        return self.current_state, self.current_reward, self.done, info
```

Εικόνα 22: Step method

Σε κάθε βήμα (βλέπε Εικόνα 22) λαμβάνονται οι απαραίτητες πληροφορίες από την get_state και ο πράκτορας εκτελεί μία δράση (action), αγορά, πώληση ή αναμονή. Η ανταμοιβή υπολογίζεται μέσω τη (calculate_reward), λαμβάνοντας υπόψη την απόδοση της αγοράς και τη στρατηγική του πράκτορα. Στη συνέχεια η κατάσταση ενημερώνεται, συμπεριλαμβανομένης της τρέχουσας τιμής της μετοχής (current_price). Εξετάζονται τα όρια ασφαλείας, ενώ παράλληλα αξιολογείται αν η προσομοίωση έχει φτάσει στο τέλος, είτε λόγω χρονικού περιορισμού (terminal_idx) (βλέπε Εικόνα 23), είτε επειδή η αξία του χαρτοφυλακίου έχει μειωθεί κάτω από το 70% του αρχικού κεφαλαίου.

Η step αποτελεί τον βασικό «μηχανισμό» που κινεί την προσομοίωση, καθορίζοντας την εξέλιξη των συναλλαγών με βάση τις αποφάσεις του πράκτορα και τις μεταβολές στην αγορά.

```
def check_terminal(self):
    return self.pointer == self.terminal_idx
```

Εικόνα 23: Terminal method

```
class Position:
    def __init__(self, entry_price, size):
        self.entry_price = entry_price
        self.size = size
        self.steps = 0

    def increment_steps(self):
        self.steps += 1
```

Εικόνα 24: Position Class

Η κλάση Position (βλέπε Εικόνα 24) , παρέχει βασικές πληροφορίες κάθε θέσης όπως την τιμή που είχε η μετοχή και το μέγεθος των μετοχών που δεσμεύτηκαν (entry price, size) την στιγμή που άνοιξε η θέση.

```
def open_position(self, entry_price, size):
    # Create and return a new Position object
    return Position(entry_price, size)
```

Εικόνα 25: Open position method

Η μέθοδος open_position (βλέπε Εικόνα 25) διαχειρίζεται τη δημιουργία μιας νέας θέσης όταν ο πράκτορας λαμβάνει δράση. Αφού υπολογιστεί το μέγεθος της εντολής και η τρέχουσα τιμή της αγοράς μέσω των υπόλοιπων συναρτήσεων του περιβάλλοντος, η open position επιστρέφει ένα αντικείμενο Position, το οποίο αποθηκεύεται στις αντίστοιχες λίστες θέσεων (long_position ή short_position). Έτσι κάθε θέση παρακολουθείται για τη διάρκειά της και χρησιμοποιείται για τον υπολογισμό κερδών/ζημιών στην συνέχεια.

```
def finalize_trade(self, position):
    # Finalize trade and update portfolio
    entry_price, size = position.entry_price, position.size
    profit_or_loss = size * (self.current_price - entry_price) - abs(size) * entry_price * self.trans_coef
    self.portfolio += profit_or_loss

    # Ensure position is removed from active lists
    if position in self.long_positions:
        self.long_positions.remove(position)
    elif position in self.short_positions:
        self.short_positions.remove(position)

    # Add position to closed positions for tracking
    self.closed_positions.append(position)
```

Εικόνα 26: Finalize trade method

Η μέθοδος finalize_trade (βλέπε Εικόνα 26) αναλαμβάνει το κλείσιμο μιας ανοιχτής θέσης (αγοράς ή πώλησης) και την επικαιροποίηση της οικονομικής κατάστασης του χαρτοφυλακίου. Υπολογίζονται τα συνολικά κέρδη/ζημιές (profit_or_loss) για τη θέση βάση της διαφοράς τιμών (τιμή εισόδου – τρέχουσα τιμή) και του κόστους συναλλαγής. Η θέση αφαιρείται από τις ενεργές λίστες (long ή short) και προστίθεται στη λίστα των κλειστών θέσεων (closed_positions). Η μέθοδος εξασφαλίζει την καταγραφή και την ανάλυση των συναλλαγών του πράκτορα, αποτελώντας κρίσιμο σημείο για τη διαχείριση του χαρτοφυλακίου.

```python
def choose_position_to_close(self, positions, position_type):
    if not positions:
        return None

    for pos in positions:
        if pos.steps >= self.max_steps:
            return pos, position_type

    for pos in positions:
        profit_loss = self.calculate_reward_for_position(pos)
        if profit_loss < -self.loss_threshold or profit_loss > self.profit_threshold:
            return pos, position_type

    min_reward_position = min(positions, key=lambda x: self.calculate_reward_for_position(x))

    return min_reward_position, position_type
```

Εικόνα 27: Choose position to close method

Η choose_position_to_close (βλέπε Εικόνα 27) επιλέγει ποια θέση θα κλείσει όταν χρειάζεται, με βάση διάφορα κριτήρια όπως η διάρκεια, η απώλεια ή το κέρδος. Παρέχει τη λογική για την επιλογή μιας θέσης για κλείσιμο όταν υπάρχει ανάγκη περιορισμού ανοιχτών θέσεων. Εξετάζει πρώτα τις θέσεις που έχουν φτάσει στο μέγιστο αριθμό βημάτων, ακολουθούμενες από θέσεις με μεγάλες ζημιές ή κέρδη. Εάν δεν ισχύει καμία από αυτές τις συνθήκες, επιλέγεται η θέση με το ελάχιστο κέρδος.

```python
def calculate_reward_for_position(self, position):
    # Calculate reward for a given position
    entry_price, size = position.entry_price, position.size
    current_value = size * self.current_price
    entry_value = size * entry_price
    trans_cost = abs(size) * entry_price * self.trans_coef
    reward = (current_value - entry_value) - trans_cost
    assert not np.isnan(reward), f"NaN reward in position: {position}"

    return reward
```

Εικόνα 28: Calculate reward for position method

Η calculate_reward_for_position (βλέπε Εικόνα 28) υπολογίζει το κέρδος ή τη ζημιά μιας ανοιχτής θέσης, λαμβάνοντας υπόψη την τρέχουσα τιμή της αγοράς και το μέγεθος της

θέσης. Ο υπολογισμός βασίζεται στη διαφορά μεταξύ της τρέχουσας αξίας της θέσης και της αρχικής τιμής εισόδου, αφαιρώντας το κόστος της συναλλαγής. Αν η θέση είναι long, το κέρδος αυξάνεται όταν η τιμή ανεβαίνει, ενώ αν είναι short, το κέρδος αυξάνεται όταν η τιμή πέφτει.

```
def calculate_reward(self):  1 usage
    risk_factor = 1.0 + (self.pnl / (abs(self.pnl) + 1e-8)) * 0.001
    self.order_size = max(1.0, self.portfolio * 0.0001 * risk_factor)
    investment = self.order_size * self.current_price
    trans_cost = investment * self.trans_coef
    total_cost = investment + trans_cost

    reward = 0
    reward_offset = 0
    trade = False
    total_positions = len(self.long_positions) + len(self.short_positions)

    for pos in self.long_positions + self.short_positions:
        if pos.steps >= self.max_steps:
            self.finalize_trade(pos)

    if total_positions >= self.position_limit:
        all_positions = self.long_positions + self.short_positions
        position_to_close, position_type = self.choose_position_to_close(all_positions,  position_type: None)

        if position_to_close:
            self.finalize_trade(position_to_close)

    if len(self.long_positions) + len(self.short_positions) < self.position_limit:
        if self.current_act == 1:
            trade = True
            new_position = self.open_position(self.current_price, self.order_size)
            self.long_positions.append(new_position)
            self.portfolio -= total_cost
        elif self.current_act == -1:
            trade = True
            new_position = self.open_position(self.current_price, -self.order_size)
            self.short_positions.append(new_position)
            self.portfolio += investment - trans_cost
        else:
            if self.current_act == self.prev_act:
                reward_offset += -0.1
```

Εικόνα 29: Calculate reward 1

```
for position in self.long_positions + self.short_positions:
    position.increment_steps()

self.store["trade"].append(trade)

# Recalculate PnL including only active positions
active_positions_pnl = sum(
    self.calculate_reward_for_position(pos) for pos in self.long_positions + self.short_positions)
self.pnl = self.portfolio + active_positions_pnl

if self.current_act != self.prev_act:
    reward = (self.pnl - self.prev_pnl) / max(abs(self.prev_pnl), 1e-8)

if self.step_count > 0:
    current_return = (self.pnl - self.prev_pnl) / max(abs(self.prev_pnl), 1e-8)
    self.returns.append(current_return)

self.next_return = np.clip(np.nan_to_num(self.next_return, nan=0.0), -1, a_max: 1)

if reward == 0:
    reward = 100 * self.next_return * self.current_act

reward += reward_offset
if reward < 0:
    reward *= NEG_MUL

sortino_ratio = np.nan_to_num(self.calculate_sortino_ratio(), nan=0.0)
drawdown = np.nan_to_num(self.calculate_drawdown(), nan=0.0)
combined_reward = ALPHA * reward + BETA * sortino_ratio - GAMMA * drawdown
reward = max(-10, min(combined_reward, 10))

self.store["position"].append((self.long_positions, self.short_positions))
self.store["pnl"].append(self.pnl)
self.store["portfolio"].append(self.portfolio)

self.prev_pnl = self.pnl

return reward
```

Εικόνα 30: Calculate reward 2

Όπως έχουμε αναφέρει η calculate_reward (βλέπε Εικόνα 29 και Εικόνα 30) είναι από τις πιο κρίσιμες μεθόδους του συστήματος, καθώς καθορίζει την ανταμοιβή του πράκτορα και κατ' επέκταση, τη στρατηγική που θα μάθει να ακολουθεί. Η σωστή διαμόρφωση είναι ζωτικής σημασίας, διότι επηρεάζει τον τρόπο με τον οποίο ο πράκτορας αξιολογεί τις κινήσεις του στην αγορά. Αν η ανταμοιβή είναι λανθασμένα σχεδιασμένη, μπορεί να το οδηγήσει σε μη ρεαλιστικά ή υπερβολικά ριψοκίνδυνες στρατηγικές.

Ο υπολογισμός της ανταμοιβής βασίζεται στο συνολικό PnL, με ενσωματωμένους παράγοντες όπως ο συντελεστής ρίσκου, το μέγεθος της θέσης και το μέγιστο drawdown. Αν ο πράκτορας επιτυγχάνει υψηλά κέρδη αλλά με τεράστιες διακυμάνσεις, το σύστημα πρέπει να τον αποθαρρύνει από υπερβολικά ριψοκίνδυνες κινήσεις, καθώς αυτές μπορεί να οδηγήσουν σε σημαντικές απώλειες σε περιόδους μεταβλητότητας.

Κατά τη διάρκεια της συναλλαγής, η calculate_reward υπολογίζει δυναμικά το μέγεθος της παραγγελίας με βάση την τρέχουσα αξία του χαρτοφυλακίου, διασφαλίζοντας ότι ο πράκτορας δεν ανοίγει υπερβολικά μεγάλες θέσεις που θα τον εξέθεταν σε αδικαιολόγητο ρίσκο. Επιπλέον, εξετάζει αν υπάρχουν θέσεις που έχουν φτάσει το μέγιστο επιτρεπτό όριο και αν χρειάζεται να κλείσουν. Αν η πράξη είναι long, το κόστος συναλλαγής αφαιρείται άμεσα από το χαρτοφυλάκιο, ενώ αν είναι short, προστίθεται το ποσό πώλησης στο χαρτοφυλάκιο, το οποίο όπως έχουμε αναφέρει, δημιουργεί μα μελλοντική υποχρέωση να ξανά αγοράσει τη μετοχή, εκθέτοντας τον σε κίνδυνο αν η τιμή ανέβει. Το αποτέλεσμα εξαρτάται από τον αν ο πράκτορας μπορεί να κλείσει τη θέση σε χαμηλότερη τιμή από αυτήν που την πούλησε. Οι ανοιχτές θέσεις ενημερώνονται και καταγράφονται, ώστε να διατηρείται η συνοχή των δεδομένων και να διασφαλίζεται ότι ο πράκτορας λαμβάνει υπόψη του το πραγματικό ρίσκο των ενεργειών του.

Για να αξιολογήσει καλύτερα την απόδοση της στρατηγικής, η calculate_reward χρησιμοποιεί δείκτες διαχείρισης ρίσκου, όπως ο Sortino ratio (βλέπε Εικόνα 31) και το μέγιστο drawdown (βλέπε Εικόνα 32), σε αντίθεση με τον Sharpe ratio που λαμβάνει υπόψη όλη τη μεταβλητότητα, ο Sortino ratio επικεντρώνεται αποκλειστικά στις αρνητικές αποδόσεις, «τιμωρώντας» τις περιόδους κατά τις οποίες το χαρτοφυλάκιο είχε χαμηλή απόδοση. Με αυτόν τον τρόπο, η ανταμοιβή επηρεάζεται αρνητικά αν ο πράκτορας επιδεικνύει έντονες διακυμάνσεις στα κέρδη του, ενθαρρύνοντας τη σταθερότητα και αποτρέποντας υπερβολικά ριψοκίνδυνες αποφάσεις. Παράλληλα, το μέγιστο drawdown μετράει τη μεγαλύτερη πτώση αξίας του χαρτοφυλακίου από την υψηλότερη κορυφή του. Αν ο πράκτορας εμφανίζει μεγάλες απώλειες πριν ανακάμψει, αυτό υποδηλώνει υψηλή μεταβλητότητα και πιθανό αυξημένο ρίσκο, μειώνοντας την ανταμοιβή του ώστε να αποτρέπεται η ανάληψη υπερβολικού κινδύνου.

Παρόλο που η calculate_reward δεν αλλάζει δυναμικά κατά τη διάρκεια της συναλλαγής, αξιολογεί τις κινήσεις του πράκτορα αφού αυτές έχουν εκτελεστεί, λαμβάνοντας υπόψη τα αποτελέσματα των ενεργειών του. Η ανταμοιβή δεν εξαρτάται μόνο από την άμεση κίνηση

της τιμής, αλλά διαμορφώνεται κυρίως από τη συνολική απόδοση του χαρτοφυλακίου, ενσωματώνοντας πληροφορίες που σχετίζονται με το κέρδος και τη διαχείριση ρίσκου.

```
def calculate_sortino_ratio(self):

    negative_returns = [r for r in self.returns if r < 0]

    if len(negative_returns) == 0:
        return 0

    if len(self.returns) > 1:
        returns = np.array(self.returns)
        downside_returns = returns[returns < 0]
        downside_deviation = np.std(downside_returns) if len(downside_returns) > 0 else 0
        mean_return = np.mean(returns)
        risk_free_rate = 0

        if downside_deviation != 0:
            sortino_ratio = (mean_return - risk_free_rate) / downside_deviation
        else:
            sortino_ratio = 0

        return sortino_ratio
    else:
        return 0
```

Εικόνα 31:Sortino method

```
def calculate_drawdown(self):
    if not self.store["portfolio"]:
        return 0

    max_portfolio_value = np.maximum.accumulate(self.store["portfolio"])
    current_portfolio_value = self.store["portfolio"][-1]

    drawdown = (max_portfolio_value[-1] - current_portfolio_value) / (max_portfolio_value[-1] + 1e-6)
    return drawdown
```

Εικόνα 32: Drawdown method

```python
def reset(self):
    # Reset environment to initial state
    self.pnl = self.bank
    self.portfolio = self.bank
    self.loss_threshold = THRESHOLD
    self.profit_threshold = THRESHOLD
    self.long_positions = []
    self.short_positions = []
    self.closed_positions = []
    self.reward = 0
    self.reward_offset = 0
    self.returns = []
    self.prev_pnl = self.bank

    self.pointer = 0
    self.next_return, self.current_state = self.get_state()
    self.current_act = 0

    if self.pointer >= len(self.asset_data): raise IndexError(
        "The index is outside the bounds of the asset_data DataFrame")

    self.current_price = self.asset_data.iloc[self.pointer, :]['close']
    self.done = False
    self.step_count = 0
    self.prev_act = 0

    return self.current_state
```

Εικόνα 33: Reset method

Η reset (βλέπε Εικόνα 33) επαναφέρει το σύστημα στην αρχική κατάσταση μετά από κάθε επεισόδιο. Αυτό είναι απαραίτητο τόσο για την έναρξη μιας νέας προσομοίωσης όσο και για τη σωστή λειτουργία της διαδικασίας εκπαίδευσης του πράκτορα. Η επαναφορά διασφαλίζει ότι κάθε επεισόδιο ξεκινά με τις ίδιες αρχικές συνθήκες, επιτρέποντας τη συγκρισιμότητα μεταξύ των δοκιμών και την αξιόπιστη αξιολόγηση των στρατηγικών συναλλαγών.

4 ΚΕΦΑΛΑΙΟ

4.1 Αποτελέσματα και αξιολόγηση Intraday συναλλαγών με ενισχυτική μάθηση

Σε αυτό το κεφάλαιο παρουσιάζονται τα αποτελέσματα της εφαρμογής του trading agent για intraday συναλλαγές, χρησιμοποιώντας τον αλγόριθμο Dueling Double Deep Q-Network (DDDQN), καθώς και την αξιολόγηση της απόδοσή του σε σύγκριση με άλλες στρατηγικές.

Η ανάλυση των αποτελεσμάτων έγινε με βάση δεδομένα του δείκτη S&P 500 από το Kaggle, το οποίο παρέχει πληθώρα dataset όχι μόνο για μετοχές και χρηματοοικονομικά. Στο παρακάτω link θα βρείτε το dataset που χρησιμοποιήθηκε https://www.kaggle.com/datasets/nickdl/snp-500-intraday-data. Στο dataset που χρησιμοποιήθηκε τα δεδομένα παρουσιάζουν τιμές ανά λεπτό, γεγονός που καθιστά τη μελέτη ιδανική για intraday συναλλαγές. Όπου υπήρχε έλλειψη δεδομένων, αυτά αντικαταστάθηκαν με τα επόμενα διαθέσιμα, εναλλακτικά ο αλγόριθμος τα κενά δεδομένα θα τα συμπλήρωνε με μηδενικές τιμές, μια πρακτική που είναι σχεδιασμένος να διαχειρίζεται.

Ο πλήρης κώδικας του αλγορίθμου μας είναι διαθέσιμος στο παρακάτω link : https://github.com/MrV0/TradingAgent

4.1.1 Εκπαίδευση και δοκιμή αλγορίθμου

Η διαδικασία εκπαίδευσης και δοκιμής του πράκτορα ενισχυτικής μάθησης για ενδοημερήσιες (intraday) συναλλαγές είναι μία εξαιρετικά χρονοβόρα και απαιτητική διαδικασία, καθώς πρέπει να βρεθούν και να αναλυθούν οι κατάλληλοι δείκτες για την διαμόρφωση του observation space. Χρησιμοποιήθηκαν δείκτες όπως Bollinger Bands, RSI, MACD και άλλα εργαλεία τεχνικής ανάλυσης που συμβάλλουν στον προσδιορισμό βέλτιστων αποφάσεων trading, όπου σε προηγούμενο κεφάλαιο έχουμε αναφέρει την χρησιμότητά τους. Παράλληλα η επιλογή των υπερπαραμέτρων για την εκπαίδευση του μοντέλου Dueling Double DQN ήταν ιδιαίτερα κρίσιμη. Οι τελικές τιμές των

υπερπαραμέτρων προσαρμόστηκαν βάση πειραματισμών για να επιτευχθούν βέλτιστα αποτελέσματα (βλέπε Πίνακας 3: First Config και Πίνακας 4: Final Config).

First Config Table	
Signals	Volume returns for [1,2,5,10,20,40] Price returns for [1,2,5,10,20,40], Volatility for [2,5,10,20,40], Bollinger, Low Bollinger, High Bollinger, RSI, MACD, STC, TR, ATR, %K, %D
Portfolio	P&L/Bank, Portfolio/P&L, Pos*Price/Bank, Previous Action
Actions	-1, 0 , 1
Observation Space	len(SIGNALS)(27) + len(Portfolio)(4)
Action Space	len(Actions)(3)
TAU (Soft update parameter)	1e-3
START	1.0
END	0.1
DECAY	0.9
GAMMA	0.9995
MEM LEN	10000
MEM THRESH	500
BATCH SIZE	200
LEARNING RATE	5e-4
TRANS COST	3e-4
BANK	100000
NEG MULT	2

Πίνακας 3: First Config

Final Config Table	
Signals	Volume returns for [1,2,5,10,20,40] Price returns

	for [1,2,5,10,20,40], Volatility for [2,5,10,20,40], Bollinger, Low Bollinger, High Bollinger, RSI, MACD LMW, MACD SMW, MACD BL, MACD, MACD Signal, MACD Histogram, STC, STC Smoothed, TR, ATR, %K, %D
Portfolio	P&L/Bank, Portfolio/P&L, Pos*Price/Bank, Previous Action
Actions	-1, 0 , 1
Observation Space	len(SIGNALS)(33) + len(Portfolio)(4)
Action Space	len(Actions)(3)
TAU (Soft update parameter)	1e-3
START	0.9
END	0.05
DECAY	0.99
GAMMA	0.8
MEM LEN	10000
MEM THRESH	500
BATCH SIZE	200
LEARNING RATE	1e-4
TRANS COST	3e-4
BANK	100000
NEG MULT	1.2

Πίνακας 4: Final Config

Η αρχική διαμόρφωση (βλέπε Πίνακας 3) όπως αναφέρθηκε περιλάμβανε συνδυασμό τεχνιτών δεικτών και μετασχηματισμών τιμή και όγκου. Ωστόσο, κατά την διάρκεια της εκπαίδευση παρατηρήθηκε ότι, παρότι ο πράκτορας φαινόταν να μαθαίνει, η απόδοσή του εμφάνιζε αστάθεια και περιορισμένη δυνατότητα γενίκευσης σε νέα δεδομένα. Αυτό οδήγησε σε τροποποιήσεις τόσο στο χώρο παρατήρησης (observation space) όσο και στις υπερπαραμέτρους, καταλήγοντας στην τελική διαμόρφωση (βλέπε Πίνακας 4: Final Config).

Συγκεκριμένα, ο χώρος παρατήρησης επεκτάθηκε από 27 σε 33 σήματα. Οι προσθήκες περιλάμβαναν επιμέρους συνιστώσες του MACD (MACD line, Signl line, Histogram) καθώς και εξομάλυνση του δείκτη STC (Smoothed STC), ώστε να προκύπτουν καθαρότερα σήματα σε δεδομένα με θόρυβο. Οι προσθήκες αυτές οδήγησαν σε μεγαλύτερη σταθερότητα του πράκτορα και σε ταχύτερη σύγκλιση, καθώς ενίσχυσαν την ικανότητα διάκρισης ανοδικών και πτωτικών σημάτων.

Αναφορικά με τις υπερπαραμέτρους, έγιναν προσεκτικές αναπροσαρμογές βάση πειραμάτων και παρατηρήσεων της συμπεριφοράς του πράκτορα.

Η παράμετρος TAU (Soft update parameter), υπεύθυνη για το ρυθμό ενημέρωσης του target δικτύου, διατηρήθηκε στην τιμή 1e-3, καθώς φάνηκε να προσφέρει καλή ισορροπία μεταξύ ταχύτητας σύγκλισης και σταθερότητας. Υψηλότερες τιμές, όπως 1e-2, προκαλούσαν μεγάλες διακυμάνσεις ενώ χαμηλότερες καθυστερούσαν την εκπαίδευση.

Η παράμετρος GAMMA (Discount factor) μειώθηκε από 0.9995 σε 0.8. Η αρχική υψηλή τιμή οδηγούσε τον πράκτορα να υπερεκτίμα μελλοντικές αποδόσεις (διατηρώντας θέσεις για πιθανή κερδοφορία), υιοθετώντας δυνητικά στρατηγικές με αυξημένο ρίσκο. Με τη χαμηλότερη τιμή, η εκπαίδευση έγινε πιο σταθερή και ο πράκτορας εστίασε περισσότερο σε βραχυπρόθεσμες αποδόσεις, κάτι που κρίνεται καταλληλότερο για ενδοημερήσιες συναλλαγές.

Το exploration schedule άλλαξε σημαντικά. Η αρχική προσέγγιση (START=1.0, END=0.1, DECAY=0.9) οδηγούσε σε πολύ γρήγορη μετάβαση από την φάση εξερεύνηση (exploration) στην εκμετάλλευση (exploitation). Η τελική διαμόρφωση (START=0.9, END=0.05, DECAY=0.99) επέτρεψε στον πράκτορα περισσότερο χρόνο εξερεύνησης στην αρχή της εκπαίδευσης και πιο ομαλή μείωση του epsilon, οδηγώντας σε πιο σταθερές στρατηγικές.

Το learning rate μειώθηκε από 5e-4 σε 1e-4. Η αρχική υψηλή τιμή προκαλούσε αστάθειες στην εκπαίδευση (στην σύγκλιση των νευρωνικών δικτύων), ειδικά σε ακραίες ανταμοιβές, (απότομων μεταβολών τιμών ή μεγάλα drawdowns). Η μείωση οδήγησε σε πιο ομαλή και σταθερή εκπαίδευση, μειώνοντας τον κίνδυνο υπερεκπαίδευσης (overfitting).

Τέλος, η ποινή για αρνητικά reward (negative multiplier) μειώθηκε από 2 σε 1.2. Η αρχική τιμή οδηγούσε σε υπερσυντηρητική συμπεριφορά του πράκτορα. Η μείωση της ποινής

επέτρεψε στον πράκτορα να αξιολογεί πιο ρεαλιστικά καταστάσεις με πιθανές προσωρινές ζημιές, οι οποίες όμως θα μπορούσαν να οδηγήσουν σε μελλοντική κερδοφορία.

	Train	Trade
1	04/22/2025	04/23/2025
2	04/23/2025	04/24/2025
3	04/24/2025	04/25/2025
4	04/25/2025	04/28/2025
5	04/28/2025	04/29/2025

Πίνακας 5: Ημερομηνίες εκπαίδευσης/δοκιμής (ενδεικτικό παράδειγμα κυλιόμενου παραθύρου ενός επεισοδίου)

Για την εύρεση της κατάλληλης διαμόρφωσης και την εκπαίδευση του πράκτορα, ακολουθήθηκε μία δομημένη διαδικασία, η οποία οργανώθηκε σε επεισόδια. Κάθε επεισόδιο περιλάμβανε έναν κύκλο ημερήσιας εκπαίδευσης και δοκιμής, υλοποιώντας μια προσέγγιση κυλιόμενου παραθύρου (Rolling Window) για την επεξεργασία δεδομένων.

Η συνολική διαδικασία είναι η εξής:

1. Εκκίνηση Επεισοδίου: Στην αρχή κάθε επεισοδίου:
    - Η ημερομηνία έναρξης της περιόδου εκπαίδευσης επαναφέρεται σε μια προκαθορισμένη αρχική ιστορική ημερομηνία. Αυτό διασφαλίζει ότι κάθε επεισόδιο επανεπεξεργάζεται την ίδια ακολουθία ημερομηνιών, επιτρέποντας στον πράκτορα να βελτιώνει τη στρατηγική του πάνω σε μια συνεπή βάση δεδομένων, καθώς διατηρεί τη μνήμη εμπειριών και τα βάρη των νευρωνικών δικτύων από τα προηγούμενα επεισόδια.
    - Τα ιστορικά δεδομένα για το σύνολο των ημερών που θα επεξεργαστεί το επεισόδιο (καλύπτοντας έναν καθορισμένο αριθμό ημερών εκπαίδευσης και τις αντίστοιχες ημέρες δοκιμής) φορτώνονται μία φορά.
2. Κύκλος ημερήσιας εκπαίδευσης και δοκιμής (Rolling window): Εντός κάθε επεισοδίου, εκτελείται επαναληπτικά για έναν προκαθορισμένο αριθμό ημερών

(στην περίπτωσή μας 5 ημέρες) ένας κύκλος που υλοποιεί τη λογική του κυλιόμενου παραθύρου. Σε κάθε επανάληψη αυτού του κύκλου, ο πράκτορας εκπαιδεύεται στα δεδομένα μιας ημέρας και αμέσως μετά δοκιμάζεται στα δεδομένα της επόμενης εργάσιμης ημέρας. Το παράθυρο αυτό «κυλάει» μία ημέρα μπροστά σε κάθε επανάληψη (βλέπε Πίνακας 5).

- Φάση εκπαίδευσης για την τρέχουσα ημέρα του παραθύρου:
  - Τα δεδομένα της συγκεκριμένης ημέρας εκπαίδευσης ανατίθενται στο περιβάλλον εκπαίδευσης.
  - Ο πράκτορας λαμβάνει την αρχική κατάσταση της ημέρας.
  - Σε έναν ενδοημερήσιο βρόχο, ο πράκτορας επιλέγει μια δράση βασισμένη στην τρέχουσα κατάσταση και την πολιτική εξερεύνησης-εκμετάλλευσης (ε-greedy), όπου η πιθανότητα τυχαίας εξερεύνησης μειώνεται σταδιακά. Η δράση εκτελείται στο περιβάλλον εκπαίδευσης, το οποίο επιστρέφει την επόμενη κατάσταση, την ανταμοιβή και μια ένδειξη ολοκλήρωσης της ημέρας. Η παρατηρούμενη μετάβαση (κατάσταση, δράση, ανταμοιβή, επόμενη κατάσταση, ολοκλήρωση) αποθηκεύεται στη μνήμη επανάληψης. Στην συνέχεια, υπό την προϋπόθεση ότι η μνήμη έχει συλλέξει επαρκή αριθμό εμπειριών και έχει παρέλθει ένας προκαθορισμένος αριθμός βημάτων από την προηγούμενη ενημέρωση, εκτελείται ένα βήμα μάθησης. Κατά το βήμα αυτό, ένα τυχαίο δείγμα εμπειριών (mini-batch) αντλείται από τη μνήμη και χρησιμοποιείται για την ενημέρωση των βαρών του κυρίου νευρωνικού δικτύου. Το δίκτυο στόχος ενημερώνεται επίσης περιοδικά με πιο αργό ρυθμό, μέσω της τεχνικής soft update. Η κατάσταση του πράκτορα ενημερώνεται με την επόμενη κατάσταση και η ημερήσια απόδοση συσσωρεύεται.
  - Ο ενδοημερήσιος βρόχος τερματίζει στο τέλος των ωρών συναλλαγών της ημέρας (16:00 για τον S&P 500). Ο πράκτορας ξεκινά κάθε ημέρα με τις πληροφορίες του τρέχοντος χαρτοφυλακίου και στο τέλος κάθε ημέρας όλες οι ανοιχτές θέσεις κλείνουν, καταγράφοντας τα ημερήσια κέρδη ή τις ζημιές.
  - Ο παράγοντας εξερεύνησης (epsilon) μειώνεται σύμφωνα με ένα προκαθορισμένο σχήμα σταδιακής μείωσης (decay schedule).

- Φάση Δοκιμής για την επόμενη εργάσιμη ημέρα του παραθύρου:
  - Τα δεδομένα της επόμενης εργάσιμη ημέρας ανατίθενται στο περιβάλλον δοκιμής.
  - Ο πράκτορας, με τα ενημερωμένα βάρη από την αμέσως προηγούμενη φάση εκπαίδευσης, επιλέγει δράσεις χρησιμοποιώντας την τρέχουσα (ήδη μειωμένη) τιμή του παράγοντα εξερεύνησης. Οι δράσεις εκτελούνται στο περιβάλλον δοκιμής. Σε αυτή τη φάση, δεν πραγματοποιείται μάθηση ούτε αποθήκευσης εμπειριών στη μνήμη. Σκοπός είναι η αμερόληπτη αξιολόγηση της τρέχουσας στρατηγικής του πράκτορα σε μη παρατηρημένα (out of sample) δεδομένα την επόμενη ημέρας.
  - Ο ενδοημερήσιος βρόχος τερματίζει στο τέλος της ημέρας και οι ανοιχτές θέσεις κλείνουν όπως και στο στάδιο της εκπαίδευσης.
- Προετοιμασία για την επόμενη ημέρα του επεισοδίου (Μετακίνηση του κυλιόμενου παραθύρου): Η ημερομηνία αναφοράς για την εκπαίδευσης προωθείται στην επόμενη εργάσιμη ημέρα, μετακινώντας ουσιαστικά το κυλιόμενο παράθυρο εκπαίδευσης/δοκιμής προς τα εμπρός για την επόμενη επανάληψη του κύκλου.

3. Τέλος Επεισοδίου:
   - Μετά την ολοκλήρωση όλων των ημερήσιων κύκλων του κυλιόμενου παραθύρου εντός ενός επεισοδίου, τα περιβάλλοντα εκπαίδευσης και δοκιμής επαναφέρονται στην αρχικής τους κατάσταση. Η συνολική διαδικασία, που περιλαμβάνει πολλαπλά επεισόδια με επαναλαμβανόμενους κύκλους κυλιόμενου παραθύρου, επαναλαμβάνεται για έναν προκαθορισμένο συνολικό αριθμό επεισοδίων (στην περίπτωσή μας 5 επεισόδια).

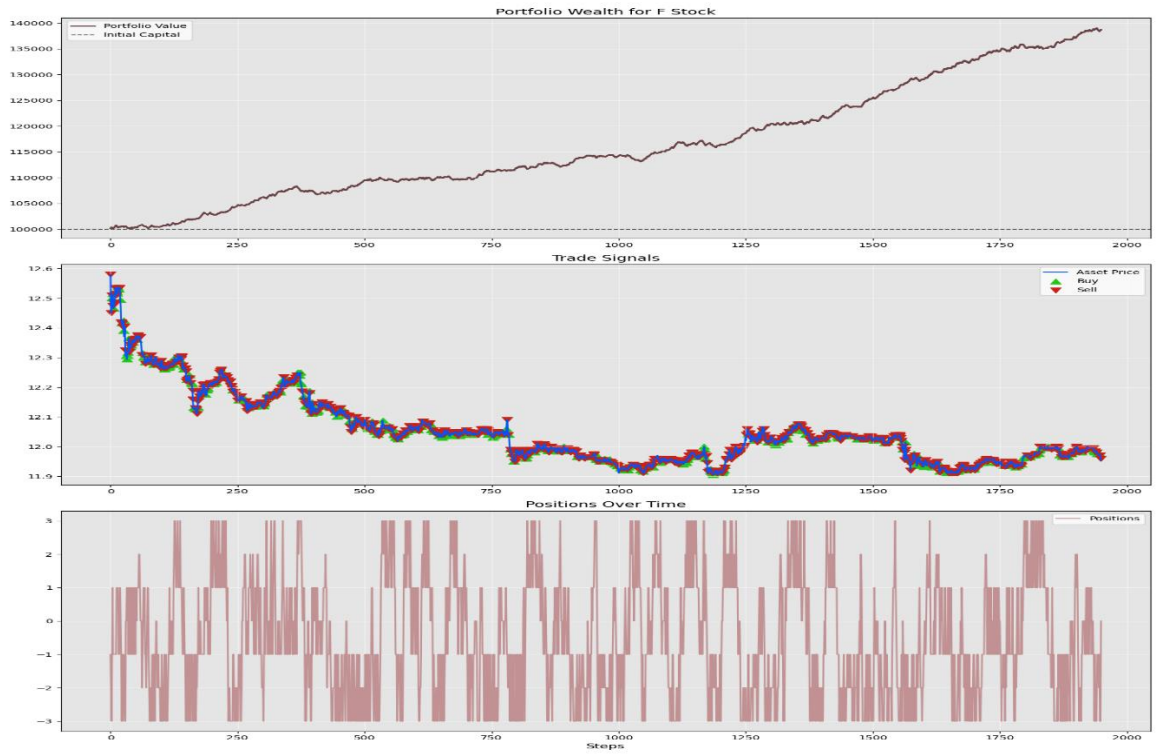
Κατά τη διάρκεια της εκπαίδευσης, ο πράκτορας προσαρμόζει τη στρατηγική του χρησιμοποιώντας τα διαθέσιμα δεδομένα αγοράς και χαρτοφυλακίου εντός του τρέχοντος παραθύρου εκπαίδευσης, ενώ κατά τη φάση της δοκιμής αξιολογούνται οι αποφάσεις του στις συνθήκες της αγοράς της επόμενης ημέρας. Η προσεκτική επιλογή των υπερπαραμέτρων, όπως ρυθμός ενημέρωσης του δικτύου στόχου (TAU), η στρατηγική

μείωσης του παράγοντα εξερεύνησης και η συχνότητα ενημέρωσης των δικτύων, ήταν καθοριστικής για την επιτυχή προσαρμογή του μοντέλου στις διακυμάνσεις της αγοράς. Μέσα από αυτή τη επαναληπτική διαδικασία, που αξιοποιεί την τεχνική του κυλιόμενου παραθύρου για την ενημέρωση των δεδομένων εκπαίδευσης και δοκιμής, η τελική διαμόρφωση του πράκτορα κατέληξε σε μια προσέγγιση trading που στόχευε σε σταθερότητα και αποτελεσματικότητα, βελτιώνοντας τόσο τη συνοχή της στρατηγικής του όσο και την ικανότητα του να μεγιστοποιεί τις αποδόσεις έναντι άλλων στρατηγικών όπως θα δούμε σε επόμενο κεφάλαιο.

4.1.2 Αποτελέσματα πράκτορα

Τα αποτελέσματα του πράκτορα παρουσιάζονται για τέσσερις μετοχές του δείκτη S&P 500, οι οποίες επιλέχθηκαν για να παρατηρηθεί η συμπεριφορά του σε διαφορετικές συνθήκες αγοράς. Οι μετοχές που χρησιμοποιήθηκαν για την ανάλυση των αποτελεσμάτων είναι, Ford Motor Company (F), Advanced Micro Devices, Inc (AMD), AES Corporation (AES) και Chesapeake Energy Corporation (CHK). Για κάθε μετοχή, τα ακόλουθα διαγράμματα (Εικόνες 34-37) απεικονίζουν την απόδοση και τη συναλλακτική δραστηριότητα του πράκτορα κατά τη διάρκεια του τελευταίου επεισοδίου δοκιμής (trading episode), το οποίο αποτελείται από περίπου 2000 χρονικά βήματα (steps) που αντιστοιχούν σε ενδοημερήσιες συναλλαγές. Κάθε διάγραμμα περιλαμβάνει τρία υπό-γραφήματα.

1. Εξέλιξη πλούτου χαρτοφυλακίου (Portfolio wealth): Δείχνει την αξία του χαρτοφυλακίου σε σχέση με το αρχικό κεφάλαιο.
2. Σήματα συναλλαγών (Trade signals): Απεικονίζει την τιμή της μετοχής (Asset Price) και τις εντολές (Buy - πράσινο τρίγωνο προς τα πάνω) και πώλησης (Sell – κόκκινο τρίγωνο προς τα κάτω) που εκτελεί ο πράκτορας.
3. Θέσεις διαχρονικά (Position over time): Δείχνει την καθαρή θέση του πράκτορα (long/short) σε κάθε χρονικό βήμα. Οι θετικές τιμές αντιστοιχούν σε long και οι αρνητικές τιμές σε short θέσεις, με το μέγιστο πλήθος ταυτόχρονων θέσεων να περιορίζεται στις +/- 3 θέσεις όπως έχουμε αναφέρει.

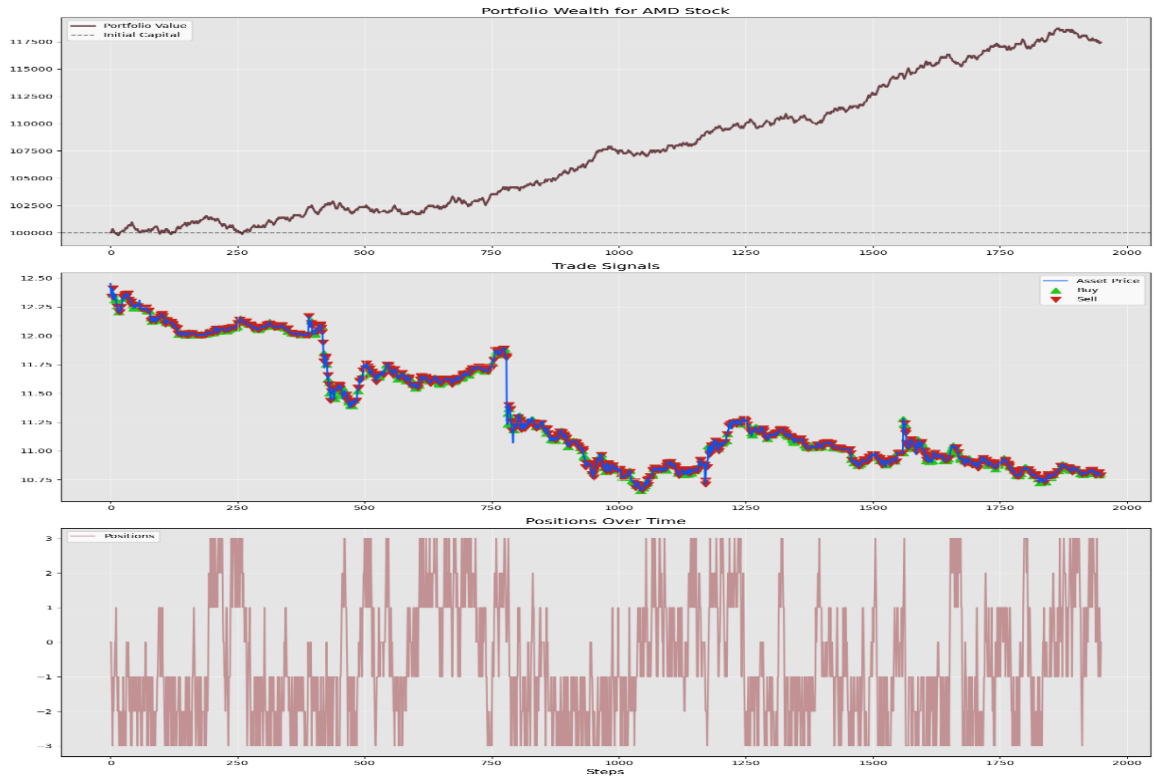


Εικόνα 34: Ford Motor Company stock

Στην μετοχή F (Εικόνα 34), το χαρτοφυλάκιο επιδεικνύει την πιο εντυπωσιακή αύξηση πλούτου από τις τέσσερις μετοχές. Η καμπύλη είναι σταθερά ανοδική και με σχετικά χαμηλή μεταβλητότητα, υποδηλώνοντας μια πολύ επιτυχημένη εφαρμογή της στρατηγικής.

Παρατηρώντας τα σήματα συναλλαγών, η τιμή της μετοχής φαίνεται να ξεκινά με μια πτωτική τάση, κατά της διάρκεια της οποίας ο πράκτορας φαίνεται να πραγματοποιεί εντολές πώλησης και το γράφημα των θέσεων επιβεβαιώνει ότι διατηρεί συχνά short θέσεις, υποδηλώνοντας ότι εκμεταλλεύεται την πτώση της τιμής για τη δημιουργία κέρδους, καθώς το χαρτοφυλάκιό του αυξάνεται. Κοντά στα 750 βήματα η τιμή της F φαίνεται να σταθεροποιείται με ελαφρές ανοδικές τάσεις σε περιόδους, παραμένοντας όμως σε στενό εύρος τιμών. Σε αυτή τη δεύτερη φάση, ο πράκτορας συνεχίζει την ενεργή δραστηριότητα, με τα σήματα συναλλαγών να δείχνουν τόσο αγορές όσο και πωλήσεις. Το γράφημα των θέσεων δείχνει μια συνεχή εναλλαγή μεταξύ long, short και ουδέτερων θέσεων, χωρίς σαφή παρατεταμένη κυριαρχία μιας μόνο κατεύθυνσης, βεβαιώνοντας την προσπάθεια του πράκτορα να εκμεταλλευτεί τις μικρές διακυμάνσεις.

Η συνεχής αύξηση του χαρτοφυλακίου, ακόμα και όταν η τιμή της μετοχής δεν ακολουθεί μια ισχυρή, μονοσήμαντη τάση στο δεύτερο μισό της περιόδου, δείχνει της προσαρμοστικότητα της στρατηγικής. Η περίπτωση της μετοχής F αναδεικνύει την ικανότητα του πράκτορα να είναι κερδοφόρος εκμεταλλευόμενος τόσο τις ομαλές διακυμάνσεις, όσο και τις πτωτικές.



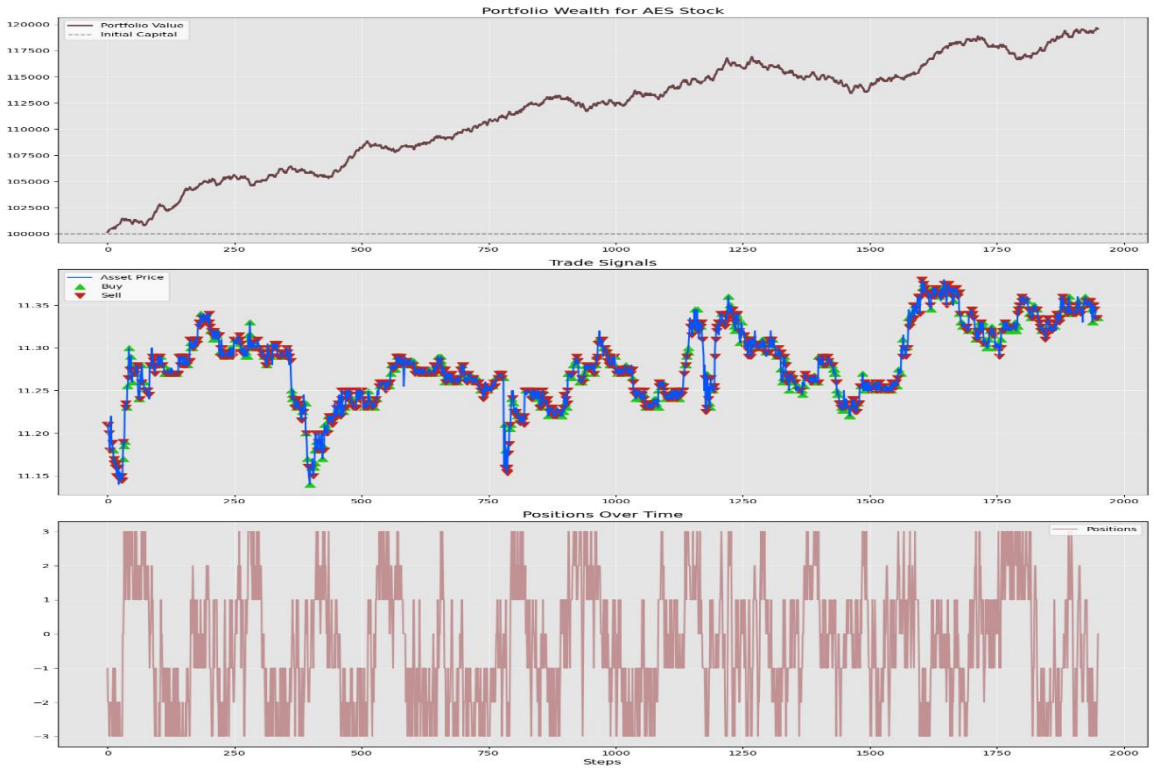
Εικόνα 35: Advanced Micro Devices, Inc stock

Στην περίπτωση της μετοχής AMD (Εικόνα 35), το χαρτοφυλάκιο εμφανίζει επίσης μια ισχυρή ανοδική πορεία, φτάνοντας σε αρκετά υψηλά επίπεδα από το αρχικό κεφάλαιο. Ωστόσο, η καμπύλη φαίνεται να παρουσιάζει ελαφρώς μεγαλύτερη μεταβλητότητα σε σύγκριση με της F, κάτι που μπορεί να αντικατοπτρίζει ενδεχομένως την μεγαλύτερη μεταβλητότητας της ίδιας της μετοχής AMD.

Τα σήματα συναλλαγών υποδεικνύουν ότι ο πράκτορας αντιδρά ενεργά στις κινήσεις της τιμής που όπως φαίνεται υπάρχει αρκετή διακύμανση. Είναι αξιοσημείωτο ότι κατά τη

διάρκεια πτώσεων της τιμής, το χαρτοφυλάκιο διατηρεί την αξία του ή συνεχίζει να αυξάνεται ελαφρώς, υποδηλώνοντας επιτυχή διαχείριση.

Το γράφημα θέσεων επιβεβαιώνει την ευελιξία του πράκτορα, καθώς είναι διατεθειμένος να κρατήσει τόσο short όσο και long θέσεις, συχνά φτάνοντας το μέγιστο επιτρεπτό όριο των +/-3 θέσεων. Η στρατηγική του φαίνεται να προσαρμόζεται αποτελεσματικά στις εναλλαγές της αγοράς, επιδιώκοντας κέρδη ανεξαρτήτου τάσης. Η ικανότητά του να διατηρεί ή να αυξάνει την κερδοφορία του ακόμη και κατά τη διάρκεια σημαντικών πτώσεων της τιμής μετοχής, επιβεβαιώνει την προσαρμοστικότητά του.



Εικόνα 36: AES Corporation stock

Για την μετοχή της AES (Εικόνα 36), η εξέλιξη του χαρτοφυλακίου μας, δείχνει μια σαφώς ανοδική τάση σε όλη τη διάρκεια του επεισοδίου, με την αξία του να υπερβαίνει σημαντικά το αρχικό κεφάλαιο. Παρά τις εμφανείς διακυμάνσεις στην τιμή της μετοχής (μεσαίο γράφημα), η αύξηση της αξίας του χαρτοφυλακίου παραμένει σχετικά σταθερή και ανοδική, υποδηλώνοντας ότι ο πράκτορας καταφέρνει να διαχειριστεί αποτελεσματικά την

μεταβλητότητα της αγοράς και να μην επηρεάζεται αρνητικά από τις πτωτικές κινήσεις της τιμής.

Το γράφημα των συναλλαγών δείχνει έναν δραστήριο πράκτορα που προσαρμόζει συνεχώς τη στρατηγική του. Η τιμή μετοχής παρουσιάζει πολλαπλές ανοδικές και πτωτικές φάσεις σε όλη την περίοδο. Η πυκνότητα των σημάτων υποδηλώνει μια στρατηγική που στοχεύει στην εκμετάλλευση των πολλαπλών ευκαιριών που παρουσιάζονται από τις διακυμάνσεις της τιμής.

Το γράφημα θέσεων επιβεβαιώνει αυτή τη δυναμική συμπεριφορά. Ο πράκτορας συχνά κυμαίνεται στα όρια των θέσεων. Παρατηρούνται επίσης σύντομες περίοδοι όπου ο πράκτορας υιοθετεί μια πιο ουδέτερη στάση. Αυτή η συνεχής προσαρμογή των θέσεων, υποδηλώνει ότι η ικανότητα του πράκτορα να επιτυγχάνει κέρδη, γίνεται από την επιτυχή αναγνώριση και εκμετάλλευση των βραχυπρόθεσμων και μεσοπρόθεσμων διακυμάνσεων της τιμής μετοχής.

Εικόνα 37: Chesapeake Energy Corporation stock

Για τη μετοχή CHK (Εικόνα 37), η απόδοση του χαρτοφυλακίου είναι και πάλι θετική, με την αξία του να αυξάνεται σταθερά πάνω από το αρχικό κεφάλαιο σχεδόν σε όλη την διάρκεια της περιόδου δοκιμής. Η συνολική αύξηση του χαρτοφυλακίου είναι πιο συγκρατημένη σε σχέση με τις υπόλοιπες μετοχές, παρουσιάζοντας φάσεις πιο αργής ανάπτυξης ή σχετικής στασιμότητας, οι οποίες ενδέχεται να αντιστοιχούν σε περιόδους αυξημένης αβεβαιότητας.

Το γράφημα των σημάτων συναλλαγών αποκαλύπτει μια εξαιρετικά δυναμική αγορά, με έντονες και συχνές διακυμάνσεις στην τιμή της. Παρατηρούνται απότομες ανοδικές εξάρσεις και γενικά περιόδους έντονης μεταβλητότητας. Ο πράκτορας έχει υψηλή δραστηριότητα, πραγματοποιώντας συχνές εντολές αγοράς και πώλησης. Η στρατηγική του φαίνεται προσανατολισμένη στην εκμετάλλευση αυτών των συχνών μεταβολών της τιμής.

Το γράφημα των θέσεων επιβεβαιώνει την ενεργή και ευέλικτη διαχείριση από την πλευρά του πράκτορα. Παρατηρούνται εναλλαγές των θέσεων, με τον πράκτορα να αξιοποιεί συχνά το μέγιστο επιτρεπτό όριο και των δύο κατευθύνσεων. Υπάρχουν επίσης και λίγες περίοδοι όπου ο πράκτορας παραμένει ουδέτερος. Η ικανότητα του πράκτορα να παράγει συνολικά θετικό αποτέλεσμα σε μια μετοχή με τόσο έντονες διακυμάνσεις τιμών, δείχνει την προσαρμοστικότητά του σε ένα απαιτητικό περιβάλλον ενδοημερήσιων συναλλαγών.

Παρατηρήσεις στη συμπεριφορά του πράκτορα DDDQN

Από την ανάλυση των αποτελεσμάτων του πράκτορα Dueling Double DQN στις τέσσερις εξεταζόμενες μετοχές, προκύπτουν ορισμένες ενδιαφέρουσες παρατηρήσεις σχετικά με τη συμπεριφορά και την απόδοσή του υπό διαφορετικές συνθήκες αγοράς:

- Ισχυρές τάσεις (ανοδικές ή πτωτικές): Ο πράκτορας φαίνεται να επιτυγχάνει εξαιρετική κερδοφορία όταν η τιμή της μετοχής ακολουθεί μια σχετικά σαφή παρατεταμένη τάση. Χαρακτηριστικό παράδειγμα είναι η μετοχή F, όπου κατά την αρχική πτωτική φάση, ο πράκτορας κατάφερε να την εκμεταλλευτεί αποτελεσματικά συμβάλλοντας σημαντικά στην αύξηση του χαρτοφυλακίου. Αυτό υποδηλώνει ότι ο πράκτορας είναι ικανός να αναγνωρίζει και να ακολουθεί τις κυρίαρχες τάσεις της αγοράς.

- Αγορές με έντονες διακυμάνσεις (Volatility): Ο πράκτορας επιδεικνύει επίσης καλή απόδοση και προσαρμοστικότητα σε αγορές που χαρακτηρίζονται από έντονες αλλά συχνές αυξομειώσεις τιμών, όπως παρατηρήθηκε στις περιπτώσεις των μετοχών AMD και AES. Η ικανότητά του να πραγματοποιεί συχνές συναλλαγές και να εναλλάσσει γρήγορα μεταξύ long και short θέσεων του επιτρέπει να εκμεταλλεύεται αυτές τις βραχυπρόθεσμες κινήσεις.
- Πλάγιες κινήσεις / Σταθερές τιμές (Sideways Market / Range-Bound): Η απόδοση του πράκτορα φαίνεται να είναι πιο συγκρατημένη, σε περιόδους όπου η τιμή της μετοχής κινείται κυρίως πλάγια, χωρίς σαφή κατεύθυνση και με μικρό εύρος διακύμανσης. Η περίπτωση της μετοχής CHK, η οποία παρουσίασε και τέτοιες φάσεις, οδήγησε σε μικρότερη αύξηση του χαρτοφυλακίου συγκριτικά με μετοχές με πιο έντονες τάσεις. Αυτό είναι αναμενόμενο, καθώς στρατηγικές που βασίζονται στην εκμετάλλευση τάσεων ή σημαντικών διακυμάνσεων έχουν λιγότερες ευκαιρίες σε τέτοιες συνθήκες αγοράς.

Συνολικά, από την ανάλυση των παραπάνω διαγραμμάτων και τις συγκεντρωτικές παρατηρήσεις, προκύπτει ότι ο πράκτορας Dueling Double DQN, παράγει θετικές αποδόσεις σε διαφορετικές μετοχές υπό διαφορετικές συνθήκες αγοράς κατά τη διάρκεια των ενδοημερήσιων συναλλαγών. Η στρατηγική του προσαρμόζεται αναλόγως, επιτυγχάνοντας κέρδη, ανεξαρτήτως της γενικής κατεύθυνσης της τιμής μετοχής, είτε αυτή είναι ανοδική, πτωτική, είτε παρουσιάζει έντονες διακυμάνσεις.

4.1.3 Συγκριτική αξιολόγηση στρατηγικών

Για την αξιολόγηση της αποτελεσματικότητας της στρατηγικής Dueling Double DQN, πραγματοποιήθηκε συγκριτική ανάλυση. Εκτός από τη σύγκριση με τρεις σημαντικές παραλλαγές του αλγορίθμου Deep Q-Network, ενσωματώθηκαν και δύο βασικές, πιο απλοϊκές στρατηγικές ως σημεία αναφοράς (baselines), η Buy & Hold και η Moving Average 5 τιμών.

Για να διασφαλιστεί μια δίκαιη σύγκριση για τις DQN-based μεθόδους, όλοι οι αντίστοιχοι πράκτορες υλοποιήθηκαν χρησιμοποιώντας την ίδια βασική αρχιτεκτονική νευρωνικού δικτύου, αποτελούμενη από τρία LSTM στρώματα με τις ίδιες διαστάσεις, όπως έχει

αναφερθεί σε προηγούμενο κεφάλαιο. Οι διαφορές μεταξύ των πρακτόρων βρίσκονται αποκλειστικά στην εφαρμογή ή μη της Dueling αρχιτεκτονικής και του μηχανισμού Double Q-learning.

- Dueling DQN: Χρησιμοποιεί την αρχιτεκτονική Dueling για τον διαχωρισμό της εκτίμησης της ροής αξίας (Value Stream) και τη ροή πλεονεκτήματος (Advantage Stream), αλλά χωρίς τον μηχανισμό Double Q-learning.
- Double DQN (Double DQN): Ενσωματώνει τον μηχανισμό Double Q-learning για την αντιμετώπιση του προβλήματος της υπερεκτίμησης των Q-τιμών (overestimation bias), αλλά χρησιμοποιεί την κλασική αρχιτεκτονική DQN.
- Βασικό DQN (DQN): Η αρχική υλοποίησης του Deep Q-Network, η οποία είναι επιρρεπής στο overestimation bias και δεν διαχωρίζει τις εκτιμήσεις value και advantage.
- Naive strategy (Buy and Hold): Μία απλοϊκή στρατηγική όπου ο πράκτορας αγοράζει τη μετοχή στην αρχή της περιόδου δοκιμής και διατηρεί τη θέση αυτή αμετάβλητη μέχρι το τέλος της περιόδου, οπότε και υπολογίζεται η τελική αξία του χαρτοφυλακίου.
- Moving Average (MA-5) strategy: Μια στρατηγική crossover που βασίζεται στον απλό κινητό μέσο όρο των τελευταίων 5 τιμών κλεισίματος. Αποφάσεις αγοράς (long) λαμβάνονται όταν η τρέχουσα τιμή διασχίζει τον MA-5 προς τα πάνω και πώληση (κλείσιμο long) όταν διασχίζει προς τα κάτω.

Η αξιολόγηση όλων των στρατηγικών βασίστηκε στα τελικά κέρδη/ζημιές του χαρτοφυλακίου, ξεκινώντας με το ίδιο αρχικό ποσό. Για τις DQN-based μεθόδους, η αποδόσεις αφορούν το τελευταίο επεισόδιο συναλλαγών μετά την ολοκλήρωση της εκπαίδευσης. Οι στρατηγικές Buy & Hold και MA-5 αξιολογήθηκαν πάνω στην ίδια ακριβώς περίοδο δεδομένων δοκιμής του τελευταίου επεισοδίου των DQN-based μεθόδων, για να διασφαλιστεί η άμεση συγκρισιμότητα, δεδομένου ότι δεν περιλαμβάνουν στάδιο μάθησης. Τα αποτελέσματα αυτής της αξιολόγησης παρουσιάζονται αναλυτικά στον Πίνακας 6.

Strategies / Stocks	Dueling Double DQN	Dueling DQN	Double DQN	DQN	Buy & Hold	MA - 5
F	138.694	135.987	92.393	88.537	95.968	98.745
AMD	117.504	116.992	100.126	98.410	86.656	96.979
AES	119.057	119.001	111.021	109.165	101.099	99.370
CHK	106.832	105.993	102.985	102.798	105.834	97.831

Πίνακας 6: Τελικά κέρδη/ζημιές χαρτοφυλακίου για κάθε στρατηγική ανά μετοχή

Όπως προκύπτει από τον Πίνακας 6, η στρατηγική Dueling Double DQN έχει σταθερά τα υψηλότερα κέρδη σε όλες τις εξεταζόμενες μετοχές. Αυτό υπογραμμίζει τα οφέλη που προκύπτουν από τον συνδυασμό των δύο βασικών βελτιώσεων του αρχικού DQN.

Συγκριτικά με τις απλούστερες στρατηγικές Buy & Hold και MA-5, ο DDDQN κατάφερε σημαντικά υψηλότερα κέρδη σχεδόν στο σύνολο των περιπτώσεων. Η στρατηγική MA-5 κατέγραψε ζημιές και στις τέσσερις μετοχές, με τις μεγαλύτερες απώλειες να παρατηρούνται στις CHK και AMD. Αυτό πιθανώς οφείλεται σε μη σαφείς τάσεις της αγοράς και της επίπτωσης των συχνών συναλλαγών και του κόστους. Η στρατηγική Buy & Hold είχε μικτά αποτελέσματα, καθώς ήταν κερδοφόρα για τις μετοχές AES και CHK, ενώ για τις περιπτώσεις των F και AMD, όπου οι μετοχές είχαν συνολικά πτωτική τάση κατά την περίοδο δοκιμής, ήταν ζημιογόνες. Τα αποτελέσματα της Buy & Hold ήταν αναμενόμενα, διότι η απόδοσή της εξαρτάται άμεσα από τη συνολική κατεύθυνση που έχει η τιμή της μετοχής κατά την εξεταζόμενη περίοδο. Αυτή η σύγκριση αναδεικνύει την ικανότητα του DDDQN να προσαρμόζεται και να επιτυγχάνει θετικές αποδόσεις σε ποικίλες συνθήκες, ακόμα και όταν απλούστερες μέθοδοι αποτυγχάνουν ή είχαν χαμηλότερη απόδοση.

Παρακάτω αναλύονται οι διαφορές των αποτελεσμάτων μεταξύ των DQN-based μεθόδων όπως επίσης και οι πιθανοί λόγοι αυτών των διαφορών.

- Dueling Double DQN vs Dueling DQN: Η DDDQN υπερτερεί, έστω και οριακά σε ορισμένες περιπτώσεις. Η Dueling DQN από μόνη της προσφέρει σημαντική βελτίωση σε σχέση με τις non-Dueling αρχιτεκτονικές, γεγονός που αποδίδεται στην ύπαρξη δύο ξεχωριστών ροών (streams) στο δίκτυό της, του value stream, που εκτιμά την αξία μιας κατάστασης (V(s)) ανεξάρτητα από την επίδραση συγκεκριμένων δράσεων, και του advantage stream που εκτιμά το σχετικό πλεονέκτημα (A(s, a)) κάθε δράσης. Αυτός ο διαχωρισμός επιτρέπει στο δίκτυο να μάθει ποιες καταστάσεις είναι «καλές ή κακές», χωρίς να χρειάζεται να αξιολογεί κάθε δράση σε κάθε κατάσταση. Ωστόσο, η Dueling DQN εξακολουθεί να είναι ευάλωτη στο πρόβλημα της υπερεκτίμησης των Q-τιμών, καθώς ο υπολογισμός της τιμής-στόχου βασίζεται στην ίδια αρχιτεκτονική που επιλέγει και αξιολογεί την καλύτερη επόμενη δράση. Η προσθήκη του μηχανισμού Double Q-learning στην DDDQN μετριάζει το πρόβλημα, καθώς η επιλογή της καλύτερης δράσης στην επόμενη κατάσταση γίνεται από το online δίκτυο, ενώ η αξιολόγησή της γίνεται από το target δίκτυο. Αυτή η αποσύνδεση οδηγεί σε πιο συντηρητικές και όπως φαίνεται πιο ακριβείς εκτιμήσεις των Q-τιμών, επιτρέποντας στον πράκτορα DDDQN να λαμβάνει ελαφρώς καλύτερες αποφάσεις και να επιτυγχάνει υψηλότερα κέρδη.
- Dueling Double DQN vs Double DQN: Η υπεροχή της DDDQN έναντι της Double DQN είναι πιο έντονη. Η Double DQN, αντιμετωπίζοντας το πρόβλημα της υπερεκτίμησης, έχει υψηλότερα κέρδη από το βασικό DQN. Ωστόσο, χωρίς την αρχιτεκτονική Dueling, ο πράκτορας Double DQN ενδέχεται να δυσκολεύεται να διακρίνει την αξία μιας κατάστασης από το πλεονέκτημα μιας συγκεκριμένης δράσης σε αυτήν. Σε περιβάλλοντα όπως οι χρηματοοικονομικές αγορές, όπου πολλές δράσεις μπορεί να έχουν παρόμοια (ή μηδαμινή) επίδραση σε ορισμένες καταστάσεις, η ικανότητα της Dueling αρχιτεκτονικής να μαθαίνει το V(s) πιο αποτελεσματικά φαίνεται να προσφέρει ένα σημαντικό πλεονέκτημα. Ο πράκτορας DDDQN, συνδυάζοντας την σταθερότητα του Double Q-learning με την αρχιτεκτονική Dueling, μπορεί να επιλέξει πιο αποτελεσματικά καταστάσεις ώστε να μεγιστοποιήσει το κέρδος.
- Dueling DQN vs Double DQN: Είναι ενδιαφέρον ότι σε ορισμένες μετοχές (F, AMD), η Dueling DQN αποδίδει καλύτερα από την Double DQN, ενώ σε άλλες (AES, CHK) οι διαφορές είναι μικρότερες (η Double DQN πλησιάζει περισσότερο).

Αυτό μπορεί να υποδηλώνει ότι για ορισμένα χαρακτηριστικά μετοχών, η βελτιωμένη αναπαράσταση που προσφέρει η Dueling αρχιτεκτονική είναι πιο κρίσιμη από την αντιμετώπιση του overestimation bias από μόνη της, ή το αντίστροφο. Ωστόσο, ο συνδυασμός και των δύο στον DDDQN φαίνεται να καλύπτει τις αδυναμίες και των δύο μεμονωμένων προσεγγίσεων.

- Βασικό DQN: Όπως αναμενόταν, το βασικό DQN επιτυγχάνει σταθερά τα χαμηλότερα αποτελέσματα. Αυτό είναι συνεπές με την εκτενή βιβλιογραφία που αναδεικνύει το πρόβλημα του overestimation bias, όπου το DQN τείνει να υπερεκτιμά τις Q-τιμές, οδηγώντας σε μη βέλτιστες πολιτικές. Επιπλέον, η έλλειψη του διαχωρισμού μεταξύ value και advantage που προσφέρει η Dueling αρχιτεκτονική περιορίζει την ικανότητά του να γενικεύει αποτελεσματικά σε διάφορες καταστάσεις και δράσεις.

Τα ευρήματα αυτά συνάδουν με τη γενικότερη βιβλιογραφία που εξετάζει τις βελτιώσεις των αλγορίθμων DQN. Η εργασίας του (Sewak, 2019), αναλύει συγκριτικά αυτές τις αρχιτεκτονικές και καταδεικνύει τα οφέλη που προσφέρουν το Double DQN στην αντιμετώπιση του overestimation bias και το Dueling DQN στην καλύτερη εκτίμηση των συναρτήσεων αξίας (V(s)) και πλεονεκτήματος (A(s, a)). Η υπεροχή του συνδυασμού αυτών των τεχνικών, όπως παρατηρείται στην παρούσα διπλωματική με τον DDDQN, επιβεβαιώνει τη θεωρητική βάση και τις παρατηρήσεις άλλων ερευνητών σχετικά με την αυξημένη απόδοση και σταθερότητα που μπορούν να επιτύχουν αυτές οι προηγμένες αρχιτεκτονικές ενισχυτικής μάθησης.

4.2 Συμπεράσματα

Στην παρούσα διπλωματική εργασία, διερευνήθηκε η εφαρμογή της ενισχυτικής μάθησης στην αυτοματοποίηση της αγοραπωλησίας μετοχών σε intraday συναλλαγές για τον δείκτη S&P 500, βασιζόμενη στον αλγόριθμο Dueling Double Deep Q-Network. Μέσα από δοκιμές και συγκριτική αξιολόγηση απλών στρατηγικών όπως επίσης με παραλλαγές του Deep Q- Network (Dueling DQN, Double DQN και απλό DQN) όπως φαίνεται στον Πίνακας 6, διαπιστώθηκε ότι η χρήση του Dueling Double DQN επέδειξε ανώτερη αποδοτικότητα, καταγράφοντας σταθερά τα υψηλότερα κέρδη σε όλες τις μετοχές που

δοκιμάστηκαν. Ο διαχωρισμός της state value και του advantage μέσω της αρχιτεκτονικής Dueling συνέβαλε ουσιαστικά στη βελτίωση της επιλογής δράσεων, ενώ η ενσωμάτωση της λογικής Double Q-learning μείωσε τις υπερεκτιμήσεις στις Q-τιμές, οδηγώντας σε πιο αξιόπιστη μάθηση.

Τα πειράματα με την προσέγγιση rolling window, όπου ο πράκτορας διατηρεί τη γνώση του (βάρη δικτύου) και τη συσσωρευμένη αξία του χαρτοφυλακίου μεταξύ των ημερών, ανέδειξαν την ικανότητα του να επιτυγχάνει σταθερή κερδοφορία σε μια ρεαλιστική προσομοίωση intraday trading. Η χρήση πολυάριθμων τεχνικών δεικτών για την προετοιμασία του observation space και ο σχεδιασμός όσο το δυνατόν ολοκληρωμένου περιβάλλοντος προσομοίωσης, στο οποίο λαμβάνονται υπόψη τα κόστη συναλλαγών και οι δυναμικές συνθήκες αγοράς, παρείχαν το σταθερό θεμέλιο για την εφαρμογή και αξιολόγηση του μοντέλου.

Παρά τα ενθαρρυντικά ευρήματα, η διπλωματική εργασία, ανέδειξε σημαντικούς περιορισμούς. Το υψηλό υπολογιστικό κόστος, η ευαισθησία στις ρυθμίσεις των υπερπαραμέτρων και οι απλουστευμένες υποθέσεις του περιβάλλοντος, περιορίζουν την πλήρη ρεαλιστικότητα της προσομοίωσης και την εφαρμογή σε πολύ δυναμικά περιβάλλοντα. Συνολικά τα αποτελέσματα της σύγκρισης όπως παρατίθενται στον Πίνακας 6, επιβεβαιώνουν ότι το Dueling Double DQN υπερτερεί εμφανώς των άλλων παραλλαγών, επαληθεύοντας τη δυνητική ικανότητα του συστήματος να βελτιστοποιεί τις στρατηγικές trading.

Η παρούσα εργασία συμβάλει τόσο σε θεωρητικό επίπεδο, αποδεικνύοντας την αποτελεσματικότητα των τεχνικών ενισχυτικής μάθησης στις χρηματοοικονομικές συναλλαγές, όσο και σε πρακτικό, ανοίγοντας νέους δρόμους για περαιτέρω έρευνα και ανάπτυξη αυτοματοποιημένων συστημάτων trading.

4.3 Μελλοντικές επεκτάσεις του αλγορίθμου

Η παρούσα διπλωματική εργασία ανοίγει πολλές προοπτικές για μελλοντικές βελτιώσεις και εφαρμογές στον χώρο της αυτόματης συναλλαγής. Μία σημαντική κατεύθυνση είναι η ενσωμάτωση τεχνιτών ανάλυσης συναισθήματος από οικονομικά ειδησεογραφικά feeds, που θα εμπλουτίζουν τα state representations και θα επιτρέψουν στον πράκτορα να

ανταποκρίνεται ακόμα καλύτερα στις μεταβολές της αγοράς, ιδιαίτερα σε περιβάλλοντα υψηλής μεταβλητότητας. Επιπλέον, η διερεύνηση πολύ-πρακτόρων, είτε σε συνεργατικό είτε σε ανταγωνιστικό πλαίσιο, αποτελεί ακόμη μία ενδιαφέρουσα κατεύθυνση, καθώς θα μπορούσαν να ανταλλάσσουν πληροφορίες και να βελτιώνουν συλλογικά τις στρατηγικές τους μέσα σε ένα κοινό περιβάλλον.

Μια άλλη προοπτική είναι η περαιτέρω βελτιστοποίηση του πλαισίου κόστους συναλλαγών. Η ενσωμάτωση μεταβλητών όπως ο όγκος και η ρευστότητα στην διαμόρφωση του κόστους μπορεί να οδηγήσει σε πιο προσαρμοσμένες στρατηγικές που ελαχιστοποιούν τα συνολικά έξοδα συναλλαγών και να αυξήσουν την κερδοφορία. Επιπλέον, η μετάβαση από το back-testing σε εφαρμογές σε πραγματικό χρόνο, με χρήση streaming δεδομένων και online learning, θα επιτρέψει στο σύστημα να λειτουργεί σε «ζωντανές» συνθήκες αγοράς, προσφέροντας δυναμική προσαρμογή.

Οι παραπάνω μελλοντικές επεκτάσεις δεν αποσκοπούν μόνο στη βελτίωση της αποτελεσματικότητας του αλγορίθμου, αλλά και στην ενίσχυση της αξιοπιστίας και της γενικότητάς του, ώστε να μπορεί να εφαρμοστεί σε ευρύτερα χρηματοοικονομικά περιβάλλοντα και πραγματικές συνθήκες αγοράς. Οι προτεινόμενες μελλοντικές έρευνες ανοίγουν το πεδίο για την ανάπτυξη ακόμα πιο ολοκληρωμένων και αποδοτικών συστημάτων trading, που θα προσφέρουν πρακτικά οφέλη τόσο σε επενδυτές όσο και σε χρηματοοικονομικούς οργανισμούς.

Βιβλιογραφία

Ακολουθούν οι βιβλιογραφικές αναφορές (πηγές) της Εργασίας.

Andey, H. (2022, June 15). Deep Reinforcement Learning for Trading Cryptocurrencies. Retrieved from https://medium.com/coinmonks/deep-reinforcement-learning-for-trading-cryptocurrencies-5b5502b1ece1

Avellaneda, M., & Stoikov, S. (2008). High-frequency trading in a limit order book. Quantitative Finance, 8(3), 217–224. https://doi.org/10.1080/14697680701381228

Azis, S. (2024). Αλγοριθμικές Συναλλαγές: Μια προσέγγιση ενισχυτικής μάθησης (Thesis). University of Patras, Patras, Greece. Retrieved from https://nemertes.library.upatras.gr/items/225e03c7-3192-41f8-bfd6-a0526cbe7e24

Bengio, Y. (2009). Learning deep architectures for AI. Foundations and Trends® in Machine Learning, 2(1), 1–127. https://doi.org/10.1561/2200000006

Bluefin. (2023, May 22). Understanding PnL: Meaning, calculation, and key metrics. Bluefin.io. https://bluefin.io/blog/understanding-pnl-meaning-calculation-and-key-metrics

Bollinger, J. (2001). Bollinger on Bollinger Bands. http://ci.nii.ac.jp/ncid/BA59933253

Deng, Y., Bao, F., Kong, Y., Ren, Z., & Dai, Q. (2017). Deep direct reinforcement learning for financial signal representation and trading. IEEE Transactions on Neural Networks and Learning Systems, 28(3), 653–664. https://doi.org/10.1109/TNNLS.2016.2522401

Glavic, M., & Saric, A. T. (2017). Reinforcement learning for energy management in buildings: A survey. Energy and Buildings, 139, 1-12

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

Hado, V. H., Guez, A., & Silver, D. (2015). Deep Reinforcement Learning with Double Q-learning. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1509.06461

Hasselt, H. V. (2010). “Double Q-learning. Neural Information Processing Systems”, 23, 2613–2621

Huang, Y., Zhou, C., Zhang, L., & Lu, X. (2024). A Self-Rewarding Mechanism in Deep Reinforcement learning for trading strategy optimization. Mathematics, 12(24), 4020. https://doi.org/10.3390/math12244020

Hyndman, R.J.(2011). Cyclic and seasonal time series. https://robjhyndman.com/hyndsight/cyclicts/

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. Journal of Artificial Intelligence Research, 4, 237-285. doi:10.1613/jair.301

Karthikeyan, A., & Priyakumar, U. D. (2021). Artificial intelligence: machine learning for chemical sciences. Journal of Chemical Sciences, 134(1). https://doi.org/10.1007/s12039-021-01995-2

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org.10.1038/nature14539

Li, Shengbo (2023). Reinforcement Learning for Sequential Decision and Optimal Control (First ed.). Springer Verlag, Singapore. pp. 1–460. doi:10.1007/978-981-19-7784-8. ISBN 978-9-811-97783-1. S2CID 257928563.

Li, Y., Ni, P., & Chang, V. (2019). Application of deep reinforcement learning in stock trading strategies and stock forecasting. Computing, 102(6), 1305–1322. https://doi.org/10.1007/s00607-019-00773-w

Li, Y., Ni, P., & Chang, V. (2019b). Application of deep reinforcement learning in stock trading strategies and stock forecasting. Computing, 102(6), 1305–1322. https://doi.org/10.1007/s00607-019-00773-w

Li, Z., Yu, H., Xu, J., Liu, J., & Mo, Y. (2023). Stock Market Analysis and Prediction using LSTM: A case study on technology stocks. Deleted Journal, 1–6. https://doi.org/10.62836/iaet.v2i1.162

Liang, Z., Chen, W., Zhu, Y., Jiang, J., Li, Z., & Li, Z. (2018). Adversarial deep reinforcement learning in portfolio management. arXiv preprint arXiv:1808.09940. https://doi.org/10.48550/arXiv.1808.09940

Mahmoodzadeh, Z., Wu, K., Droguett, E. L., & Mosleh, A. (2020). Condition-Based Maintenance with Reinforcement Learning for Dry Gas Pipeline Subject to Internal Corrosion. Sensors, 20(19), 5708. https://doi.org/10.3390/s20195708

Matiisen, Tambet (December 19, 2015). "Demystifying Deep Reinforcement Learning". neuro.cs.ut.ee. Computational Neuroscience Lab. Retrieved 2018-04-06

Melo, Francisco S. "Convergence of Q-learning: a simple proof"

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1312.5602

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533. doi:10.1038/nature14236

Moody, J., Wu, L., Liao, Y., & Saffell, M. (1998). Performance functions and reinforcement learning for trading systems and portfolios. Journal of Forecasting, 17(5–6), 441–470. https://doi.org/10.1002/(sici)1099-131x(1998090)17:5/6

Murphy, J. J. (1999). Technical analysis of the financial markets: a comprehensive guide to trading methods and applications. Choice Reviews Online, 36(07), 36–4016. https://doi.org/10.5860/choice.36-4016

RavenProtocol. (2017, December 4). Everything you need to know about neural networks. Medium. Retrieved from https://medium.com/ravenprotocol/everything-you-need-to-know-about-neural-networks-6fcc7a15cb4

Rodinos, G., Nousi, P., Passalis, N., & Tefas, A. (2023). A sharpe ratio based reward scheme in deep reinforcement learning for financial trading. In IFIP advances in information and communication technology (pp. 15–23). https://doi.org/10.1007/978-3-031-34111-3_2

Rollinger, T. N., & Hoffman, S. T. (2023). Sortino: A 'Sharper' Ratio. Red Rock Capital. https://www.cmegroup.com/education/files/rr-sortino-a-sharper-ratio.pdf

Sami, H., Kazi, A., & Rozario, P. (2023). Determining the best activation functions for predicting stock prices in different (Stock exchanges) through multivariable time series

forecasting of LSTM. Australian Journal of Engineering and Innovative Technology, 63–71. https://doi.org/10.34104/ajeit.023.063071

Sanghavi, M., & Benedict, S. M. (2024). Unlocking investment insights: Exploring standard deviation and coefficient of variation in stock analysis. International Journal of Creative Research Thoughts (IJCRT), 12(4), 311-317. Retrieved from https://www.ijcrt.org

Schaff, D. (2008, February 15). Releasing the Code to the Schaff Trend Cycle. Retrieved from https://studylib.net/doc/26257935/releasing-the-code-to-the-schaff-trend-cycle

Sewak, M. (2019). Deep Q Network (DQN), double DQN, and dueling DQN. In Springer eBooks (pp. 95–108). https://doi.org/10.1007/978-981-13-8285-7_8

Shavandi, A., & Khedmati, M. (2022). A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets. Expert Systems With Applications, 208, 118124. https://doi.org/10.1016/j.eswa.2022.118124

Silver, D., Singh, S., Precup, D., & Sutton, R. S. (2021). Reward is enough. Artificial Intelligence, 299, 103535. doi:10.1016/j.artint.2021.103535

Sutton, R., & Barto, A. (1998). Reinforcement Learning: An Introduction. IEEE Transactions on Neural Networks, 9(5), 1054. https://doi.org/10.1109/tnn.1998.712192

Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning, second edition: An Introduction. MIT Press

Théate, T., & Ernst, D. (2021). An application of deep reinforcement learning to algorithmic trading. Expert Systems With Applications, 173, 114632. https://doi.org/10.1016/j.eswa.2021.114632

Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. Artificial Intelligence Review, 53(8), 5929–5955. https://doi.org/10.1007/s10462-020-09838-1

Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., & De Freitas, N. (2015). Dueling network architectures for deep reinforcement learning. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1511.06581

Watkins, C. J. C. H. (1989). Learning from delayed rewards (Doctoral dissertation, University of Cambridge)

Wikipedia contributors. (n.d.). MACD. In Wikipedia. Retrieved December 15, 2024, from https://en.wikipedia.org/wiki/MACD

Wikipedia contributors. (n.d.). Moving average. In Wikipedia. Retrieved December 15, 2024, from https://en.wikipedia.org/wiki/Moving_average

Wikipedia contributors. (n.d.). Relative change. In Wikipedia. Retrieved December 14, 2024, from https://en.wikipedia.org/wiki/Relative_change

Wikipedia contributors. (n.d.). Relative strength index. In Wikipedia. Retrieved December 16, 2024, from https://en.wikipedia.org/wiki/Relative_strength_index

Wikipedia contributors. (n.d.). Stochastic oscillator. In Wikipedia. Retrieved December 17, 2024, from https://en.wikipedia.org/wiki/Stochastic_oscillator

Wilder, J. W. (1978). New concepts in technical trading systems. https://agris.fao.org/agris-search/search.do?recordID=US201300554903

Xu, Y. (2021). The impact of AI on algorithmic trading. Investopedia. https://www.investopedia.com/the-impact-of-ai-on-algorithmic-trading-5185469

Zhang, Y., Zhou, Z., & Zhang, D. (2020). Stock market prediction with multi-agent reinforcement learning. IEEE Access, 8, 9298-9305. https://doi.org/10.1109/ACCESS.2020.2964682

Υπεύθυνη Δήλωση Συγγραφέα:

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν.1599/1986, η παρούσα εργασία αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης.