



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ
ΠΑΡΑΓΩΓΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΠΛΟΙΩΝ ΚΑΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΔΙΑΔΡΟΜΩΝ ΜΕ
ΤΕΧΝΙΚΕΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ**

ΚΟΜΠΟΤΗΣ ΠΑΝΑΓΙΩΤΗΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΖΑΧΑΡΙΑ ΠΑΡΑΣΚΕΥΗ

ΑΘΗΝΑ, ΦΕΒΡΟΥΑΡΙΟΣ 2023



Η παρούσα διπλωματική εργασία εγκρίθηκε ομόφωνα από την τριμελή εξεταστική επιτροπή, η οποία ορίστηκε από την Γ.Σ. του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής του Πανεπιστημίου Δυτικής Αττικής, σύμφωνα με το νόμο και τον εγκεκριμένο Οδηγό Σπουδών του τμήματος.

Επιβλέπων καθηγητής: Ζαχαρία Παρασκευή

Επιτροπή Αξιολόγησης:

.....
Ζαχαρία Παρασκευή Λελίκου Ελένη Αικατερίνη Νικολάου Γρηγόρης



ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/η κάτωθι υπογεγραμμένος/η **Κομπότης Παναγιώτης** του **Γεωργίου**, με αριθμό μητρώου **47196** φοιτητής/τρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής **Μηχανικών** του Τμήματος **Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής**, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο/Η Δηλών/ούσα



Ευχαριστίες

*Μετά την ολοκλήρωση αυτής της εργασίας,
θα ήθελα να ευχαριστήσω ειλικρινά
την καθηγήτρια κα Ζαχαρία Παρασκευή
για τις συμβουλές και τη βοήθειά της
στην ανάπτυξη της διπλωματικής μου.
Επιπλέον, θέλω να ευχαριστήσω την
οικογένειά μου για την αμέριστη
υποστήριξή της καθ' όλη τη διάρκεια
της προετοιμασίας της εργασίας και
κατά την ολοκλήρωση του πενταετούς
κύκλου σπουδών.*



Πρόλογος

Τα τελευταία χρόνια, έχει γίνει πολλή έρευνα για τον σχεδιασμό διαδρομής και την αποφυγή εμποδίων. Ο σχεδιασμός δρομολογίων είναι ένα ζήτημα με μεγάλη σημασία για τον σχεδιασμό κίνησης των θαλάσσιων σκαφών. Όταν υπάρχουν δυναμικοί παράγοντες στο περιβάλλον, αβεβαιότητες στα δεδομένα των αισθητήρων και πρότυπα ασφαλείας που πρέπει να πληρούνται, αυτή η δραστηριότητα μπορεί να γίνει αρκετά δύσκολη. Το πρόβλημα της συντομότερης διαδρομής παρουσιάζει μεγάλη δημοτικότητα τα τελευταία χρόνια λόγω των πολλών εφαρμογών στον τομέα των μεταφορών. Αυτό σχετίζεται άμεσα με τις εξελίξεις στα Ευφυή Συστήματα Μεταφορών, ειδικά στους τομείς του συστήματος καθοδήγησης διαδρομής εντός οχήματος και του αυτοματοποιημένου συστήματος διανομής οχημάτων σε πραγματικό χρόνο, όπου υπάρχει σαφής απαίτηση για ακρίβεια στις διαδρομές από την αρχή έως τον προορισμό. Ένα αυτοματοποιημένο σύστημα οχημάτων πρέπει να εντοπίζει γρήγορα, σε πραγματικό χρόνο, νέες διαδρομές και δρομολόγια. Οι αλγόριθμοι σχεδιασμού διαδρομής χρησιμοποιούνται προκειμένου να δημιουργηθεί ένα μονοπάτι χωρίς εμπόδια μεταξύ του σημείου εκκίνησης και του τελικού σημείου της διαδρομής και εξετάζουν όλες τις πληροφορίες που είναι διαθέσιμες για μια συγκεκριμένη περιοχή. Η παραγόμενη διαδρομή πρέπει να συμμορφώνεται με έναν αριθμό περιορισμών, όπως οι δυναμικοί περιορισμοί του οχήματος και οι περιβαλλοντικοί περιορισμοί. Στο κομμάτι αυτό προστίθεται και η δημιουργία μοντέλων για την κατηγοριοποίηση των πλοίων με στόχο την παρακολούθησή τους για ασφαλείς διαδρομές.



Περίληψη

Αντικείμενο μελέτης της παρούσας διπλωματικής εργασίας είναι η ανάλυση των δημοφιλέστερων αλγορίθμων δρομολόγησης πλοίων, όσον αφορά την βελτιστοποίηση των διαδρομών καθώς επίσης και την δημιουργία μοντέλου για την ορθή κατηγοριοποίηση διαφορετικών τύπων πλοίων με την χρήση εικόνων. Σε πρώτο στάδιο, πραγματοποιείται αναφορά στις θαλάσσιες μεταφορές και γίνεται ανάλυση των βασικών αλγορίθμων δρομολόγησης. Στην συνέχεια, γίνεται αναφορά στους βασικούς τύπους Μηχανικής Μάθησης και στις βασικότερες εφαρμογές της και κυρίως στην αναγνώριση εικόνας. Η διπλωματική προχωράει στην ανάλυση των βασικών αρχών των Νευρωνικών Δικτύων καθώς επίσης και στην ανάλυση των δικτύων Βαθιάς Μάθησης με αποκορύφωμα το μοντέλο Xception το οποίο χρησιμοποιείται για την εκπαίδευση μοντέλου κατηγοριοποίησης. Κατά τη διαδικασία σχεδιασμού και υλοποίησης του μοντέλου κατηγοριοποίησης γίνεται αναφορά στο περιβάλλον ανάπτυξης και στις βασικές βιβλιοθήκες που χρησιμοποιήθηκαν.

Λέξεις κλειδιά

Αλγόριθμος δρομολόγησης, μηχανική μάθηση, βαθιά μάθηση, νευρωνικό δίκτυο, βέλτιστη διαδρομή, xception, κατηγοριοποίηση εικόνων, Dijkstra, A-star, Visibility graph, Voronoi diagram



Abstract

The object of study of this diploma thesis is the analysis of the most popular ship routing algorithms, regarding the optimization of routes as well as the creation of a model for the correct categorization of different types of ships using images. In a first stage, a reference is made to maritime transport and an analysis of the basic routing algorithms is made. Then, reference is made to the basic types of Machine Learning and its most basic applications, especially image recognition. Diploma proceeds to analyze the basic principles of Neural Networks as well as the analysis of Deep Learning networks culminating in the Xception model which is used to train a classification model. During the planning and implementation process of the categorization model, reference is made to the development environment and the basic libraries used.

Keywords

Routing algorithm, machine learning, deep learning, neural network, optimal path, xception, image categorization, Dijkstra, A-star, Visibility graph, Voronoi diagram



Περιεχόμενα

Πρόλογος	5
Περίληψη	6
Abstract	7
Κεφάλαιο 1 Εισαγωγή	12
1.1 Διάρθρωση.....	13
Κεφάλαιο 2 Αλγόριθμοι δρομολόγησης	18
2.1 Προκλήσεις παρακολούθησης της θαλάσσιας κυκλοφορίας.....	15
2.2 Βασικές έννοιες	17
2.3 Το πρόβλημα της συντομότερης διαδρομής.....	19
2.4 Αλγόριθμοι διαδρομής.....	21
2.4.1 Αλγόριθμος Dijkstra.....	21
2.4.2 Μέθοδος visibility graph.....	25
2.4.3 Διάγραμμα Voronoi.....	26
2.5 Ευρετικοί αλγόριθμοι συντομότερης διαδρομής.....	28
2.5.1 Αλγόριθμος A-star.....	28
2.5.2 Βελτιωμένος αλγόριθμος δυναμικής αναζήτησης A-star.....	31
Κεφάλαιο 3 Μηχανική Μάθηση	34
3.1 Ορισμός Μηχανικής Μάθησης.....	34
3.2 Τύποι Μάθησης.....	35
3.2.1 Μάθηση με επίβλεψη.....	35



3.2.2 Μάθηση χωρίς επίβλεψη.....	36
3.2.3 Μάθηση με ενίσχυση.....	37
3.3 Εφαρμογές μηχανικής μάθησης.....	38
3.4 Αναγνώριση εικόνας.....	38
Κεφάλαιο 4 Νευρωνικά δίκτυα.....	39
4.1 Ορισμός Νευρωνικών Δικτύων.....	39
4.2 Ιστορική αναδρομή.....	40
4.3 Δομή βιολογικού νευρώνα.....	41
4.4 Δομή τεχνητού νευρώνα.....	42
4.5 Συναρτήσεις ενεργοποίησης.....	44
4.6 Αρχιτεκτονική δικτύων.....	46
4.6.1 Δίκτυα πρόσθιας τροφοδότησης πολλών στρωμάτων.....	46
4.6.2 Αναδρομικά Νευρωνικά Δίκτυα.....	47
4.7 Ανάπτυξη τεχνητού νευρωνικού δικτύου για πρόβλεψη θαλάσσιας κυκλοφορίας.....	49
Κεφάλαιο 5 Βαθιά μάθηση.....	51
5.1 Ορισμός Βαθιάς Μάθησης.....	51
5.2 Νευρωνικά δίκτυα βαθιάς μάθησης.....	51
5.3 Συνελικτικά νευρωνικά δίκτυα.....	52
5.4 Επίπεδο συνέλιξης.....	53
5.5 Επίπεδα συγκέντρωσης	54
5.6 Μοντέλο Xception.....	56
Κεφάλαιο 6 Περιβάλλον ανάπτυξης και βιβλιοθήκες.....	59
6.1 Anaconda Jupyter Notebook.....	59



6.2 TensorFlow.....	59
6.3 Keras.....	60
Κεφάλαιο 7 Σχεδιασμός και υλοποίηση.....	62
Κεφάλαιο 8 Συμπεράσματα.....	78
Βιβλιογραφία.....	79

Πίνακας εικόνων

Εικόνα 1. Θαλάσσια κυκλοφορία στη Βαλτική Θάλασσα.....	16
Εικόνα 2. Γραμμικά σχέδια.....	18
Εικόνα 3. Η διαδικασία εφαρμογής του αλγορίθμου του Dijkstra.....	24
Εικόνα 4. Γράφημα ορατότητας.....	26
Εικόνα 5. Δημιουργία ενός γενικού σχεδίου ωκεάνιου οικοσυστήματος με βάση το διάγραμμα Νοτοποί.....	27
Εικόνα 6. Το μοντέλο της μάθησης με επίβλεψη.....	35
Εικόνα 7. Το μοντέλο μάθησης χωρίς επίβλεψη.....	37
Εικόνα 8. Το μοντέλο μάθησης με ενίσχυση.....	37
Εικόνα 9. Τα βασικά χαρακτηριστικά ενός βιολογικού νευρώνα.....	41
Εικόνα 10. Διάγραμμα τεχνητού νευρώνα.....	43
Εικόνα 11. Το πολυστρωματικό perceptron.....	47
Εικόνα 12. Σχεδιάγραμμα αναδρομικού νευρωνικού δικτύου.....	48
Εικόνα 13. Τεχνητό νευρωνικό δίκτυο για θαλάσσια κυκλοφορία.....	49
Εικόνα 14. Η αρχιτεκτονική ενός δικτύου CNN.....	52
Εικόνα 15. Πίνακας συνέλιξη.....	54
Εικόνα 16. Παράδειγμα παγκόσμιας μέσης συγκέντρωσης.....	55
Εικόνα 17. Διάγραμμα μοντέλου Xception.....	57
Εικόνα 18. Σύγκριση ImageNet.....	58
Εικόνα 19. Σύμβολο Jupyter Notebook.....	59
Εικόνα 20. Σύμβολο Tensorflow.....	60
Εικόνα 21. Σύμβολο Keras.....	60
Εικόνα 22. Ανάλυση της απόδοσης των μοντέλων ταξινόμησης.....	76

Κεφάλαιο 1

Εισαγωγή

Τα τελευταία χρόνια, η σημασία των θαλάσσιων μεταφορών αυξάνεται ραγδαία. Η εύρεση συγκεκριμένων προτύπων στις κινήσεις των πλοίων είναι ένα από τα πιο κρίσιμα ζητήματα για την πρόβλεψη της θαλάσσιας κυκλοφορίας. Καθώς αναπτύσσονται οι δυνατότητες τεχνητής νοημοσύνης, η ναυτιλιακή βιομηχανία μεταφορών αναζητά τρόπους για να αυξήσει την ασφάλεια, την αποτελεσματικότητα και την κερδοφορία. Τα μη επανδρωμένα θαλάσσια συστήματα εξελίσσονται σε ένα ταχέως μεταβαλλόμενο περιβάλλον. Υπάρχουν πολυάριθμες ερευνητικές προσπάθειες σχετικά με την διαδικασία σχεδιασμού διαδρομής θαλάσσιων οχημάτων. Ο σχεδιασμός της διαδρομής έχει πολλές παγίδες όπως η ασφάλεια, η πολυπλοκότητα και η περιβαλλοντική δυναμική που εμποδίζουν την εμφάνιση αξιόπιστων αυτόνομων συστημάτων. Λόγω τοπογραφικών χαρακτηριστικών και άλλων διαταραχών, ακόμη και τα σύγχρονα εργαλεία πλοήγησης δεν μπορούν να παρέχουν μια πλήρη ναυτική εικόνα σε περιορισμένους ποταμούς. Η ανάπτυξη εξελιγμένων συστημάτων παρακολούθησης και πρόβλεψης για τη θαλάσσια κυκλοφορία γίνεται πιο κρίσιμη ως αποτέλεσμα των ζωνών συμφόρησης πλοίων.

Η δημιουργία μιας δυνητικά καλύτερης διαδρομής μεταξύ δύο καθορισμένων θέσεων σε έναν προκαθορισμένο χώρο αναφέρεται γενικά ως εύρεση διαδρομής. Το ζήτημα της συντομότερης διαδρομής είναι ένας κοινός όρος που χρησιμοποιείται στην επιστήμη των υπολογιστών για να περιγράψει την εύρεση μιας διαδρομής από ένα σημείο Α στο σημείο Β. Πολλοί αλγόριθμοι που ασχολούνται με περιστάσεις όπως η διαδρομή σχεδιασμού ενός ή περισσότερων οχημάτων έχουν δημιουργηθεί για να χειριστούν αυτήν την πρόκληση.

Η υπολογιστική πολυπλοκότητα προκύπτει από έναν πολυδιάστατο χώρο αναζήτησης, από δυναμικές αλλαγές στο περιβάλλον και από τις ανωμαλίες του είναι καθοριστικό στοιχείο πολλών εργασιών κυκλοφοριακού σχεδιασμού. Επομένως, είναι απαραίτητη η ανάπτυξη αποδοτικών μεθόδων κυκλοφοριακού σχεδιασμού.

Στόχος της διπλωματικής εργασίας είναι η ανάλυση των βασικότερων αλγορίθμων εύρεσης συντομότερων διαδρομών και η ανάπτυξη μοντέλου για την ορθή ταξινόμηση πλοίων ανά κατηγορία, μέσω ενός συνόλου δεδομένων εικόνων.

1.1 Διάρθρωση

Η παρούσα διπλωματική εργασία αναπτύσσεται σε οχτώ κεφάλαια, στα οποία δίνονται οι απαραίτητες πληροφορίες για την κάλυψη του θεωρητικού υπόβαθρου και παρουσιάζονται οι σημαντικότεροι αλγόριθμοι δρομολόγησης, ενώ τα δύο τελευταία κεφάλαια αποτελούν το πρακτικό μέρος της εργασίας.

Το δεύτερο κεφάλαιο αφορά τις προκλήσεις της θαλάσσιας κυκλοφορίας και τους σημαντικότερους και δημοφιλέστερους αλγορίθμους δρομολόγησης. Γίνεται μια συνοπτική αναφορά της απαραίτητης ορολογίας και στην συνέχεια αναλύονται οι αλγόριθμοι Dijkstra, Visisbility Graph, Voronoι diagram καθώς και οι ευρετικοί αλγόριθμοι A-star και ο βελτιωμένος αλγόριθμος δυναμικής δρομολόγησης A-star.

Το τρίτο κεφάλαιο αποτελεί το θεωρητικό υπόβαθρο της Μηχανικής Μάθησης, με αναφορά στους τύπους και τις εφαρμογές της και ολοκληρώνεται με εκτενή αναφορά στην αναγνώριση εικόνας.

Το τέταρτο κεφάλαιο αποτελεί εισαγωγή στα Νευρωνικά Δίκτυα ξεκινώντας, για λόγους πληρότητας, από τον ορισμό και την ιστορική τους αναδρομή. Στην συνέχεια αναλύεται η δομή του βιολογικού και του τεχνητού νευρώνα και παρουσιάζονται οι συναρτήσεις ενεργοποίησης. Το κεφάλαιο ολοκληρώνεται με αναφορά στην αρχιτεκτονική των δικτύων.

Στο πέμπτο κεφάλαιο γίνεται εισαγωγή στην Βαθιά Μάθηση και στα νευρωνικά δίκτυα βαθιάς μάθησης. Πραγματοποιείται ανάλυση των επιπέδων συνέλιξης και συγκέντρωσης και το κεφάλαιο ολοκληρώνεται με εκτενή αναφορά στο βαθύ συνελικτικό νευρωνικό μοντέλο Xception το οποίο χρησιμοποιήθηκε κατά την εκπαίδευση του μοντέλου κατηγοριοποίησης.

Στο έκτο κεφάλαιο παρουσιάζεται το Jupyter Notebook το οποίο αποτελεί το περιβάλλον ανάπτυξης του πειραματικού μέρους και οι βιβλιοθήκες Tensorflow και Keras.

Το έβδομο κεφάλαιο αποτελεί την πειραματική μελέτη της εργασίας κατά την οποία χρησιμοποιούνται δύο σύνολα δεδομένων, ένα για εκπαίδευση και άλλο ένα για πρόβλεψη του μοντέλου σε τυχαίες εικόνες.

Στο όγδοο κεφάλαιο συνοψίζονται τα συμπεράσματα που προκύπτουν από την υλοποίηση και την εφαρμογή του μοντέλου κατηγοριοποίησης.

Κεφάλαιο 2

Αλγόριθμοι δρομολόγησης

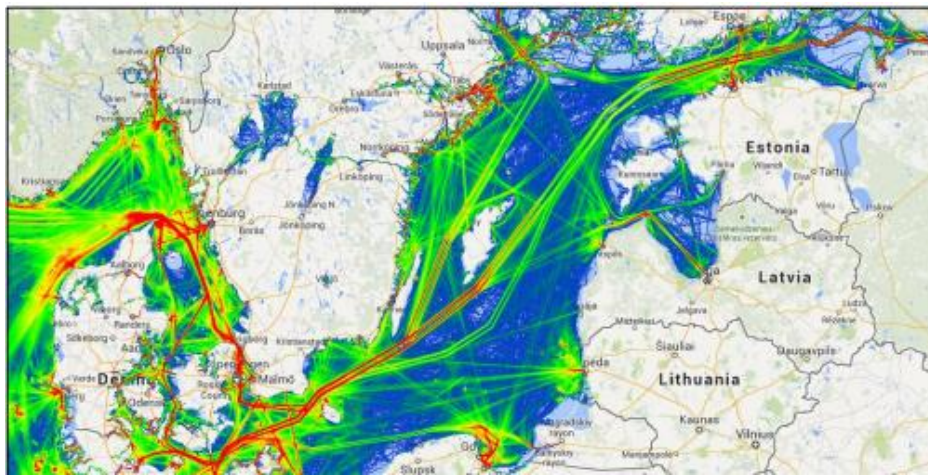
2.1 Προκλήσεις παρακολούθησης της θαλάσσιας κυκλοφορίας

Η αξία των θαλάσσιων μεταφορών έχει αυξηθεί σημαντικά τα τελευταία χρόνια. Ένα από τα πιο σημαντικά προβλήματα στην πρόβλεψη της θαλάσσιας κυκλοφορίας είναι ο εντοπισμός συγκεκριμένων προτύπων στις κινήσεις των πλοίων. Ακόμη και τα σύγχρονα βοηθήματα πλοήγησης δεν μπόρεσαν να δώσουν μια πλήρη ναυτική εικόνα σε περιορισμένους ποταμούς λόγω τοπογραφικών παραγόντων και άλλων διαταραχών.

Η ανάγκη για αυξανόμενες προκλήσεις ασφάλειας στη θαλάσσια κυκλοφορία καθιστά πιο κρίσιμη την πρόβλεψη της θαλάσσιας κυκλοφορίας. Η ανάπτυξη εξελιγμένων συστημάτων παρακολούθησης και πρόβλεψης για τη θαλάσσια κυκλοφορία γίνεται πιο κρίσιμη ως αποτέλεσμα των ζωνών συμφόρησης πλοίων.

Η πρωταρχική πρόκληση ενός τέτοιου ευφυούς συστήματος παρακολούθησης και πρόβλεψης θαλάσσιας κυκλοφορίας είναι να προβλέψει την πορεία του πλοίου καθ' όλη τη διάρκεια του ταξιδιού του. Για το σκοπό αυτό μπορούν να χρησιμοποιηθούν πληροφορίες από το αυτόματο σύστημα αναγνώρισης του σκάφους. Διαφορετικά, αυτό το σύστημα έχει πρόσβαση μόνο σε κρίσιμα δεδομένα σχετικά με το λιμάνι προορισμού για την πρόβλεψη διαδρομής. Επομένως, όλες οι περιοχές στροφής της διαδρομής πρέπει να προβλεφθούν προκειμένου να μειωθεί σημαντικά ο κίνδυνος των θαλάσσιων μεταφορών.

Οι υπηρεσίες κυκλοφορίας πλοίων διαχειρίζονται τη θαλάσσια κυκλοφορία. Εξασφαλίζουν τη διέλευση του σκάφους χρησιμοποιώντας τεχνικές διαχωρισμού της κυκλοφορίας και ειδοποιούν τα διερχόμενα σκάφη για πιθανές συγκρούσεις και ατυχήματα. Πολλές συσκευές θαλάσσιας επιτήρησης, συμπεριλαμβανομένων των ραντάρ, των αυτόματων βοηθημάτων σχεδίασης, των συστημάτων αυτόματης αναγνώρισης και των συστημάτων αναγνώρισης και παρακολούθησης μεγάλης εμβέλειας, περιλαμβάνονται στις συμβατικές υπηρεσίες κυκλοφορίας πλοίων για το σκοπό αυτό.



Εικόνα 1. Θαλάσσια κυκλοφορία στη Βαλτική Θάλασσα[20]

Τα συστήματα θαλάσσιας επιτήρησης αντιμετωπίζουν και μεγάλες προκλήσεις όπως:

- Τεράστια περιοχή επιτήρησης
- Μεγάλος αριθμός στόχων
- Παρατηρήσεις πολλαπλών στόχων
- Θορυβώδες σήμα και γεωγραφικές δομές
- Διάδοση ραδιοκυμάτων

Συνήθως, αυτές οι περιοχές βρίσκονται σε λιμάνια, στενά και κανάλια. Παρακάτω προσδιορίζονται τα αίτια της συμφόρησης της θαλάσσιας κυκλοφορίας:

- Η πλοήγηση των πλοίων είναι ασταθής
- Υπερπλήρη στενά και κανάλια
- Τα πλοία πλέουν αργά και συχνά ανεβοκατεβάζουν ταχύτητα

Με τη χρήση τεχνητών νευρωνικών δικτύων, αυτές οι δυσκολίες θα μπορούσαν να επιλυθούν. Η αυτοματοποίηση της διαχείρισης της θαλάσσιας κυκλοφορίας εξαρτάται ουσιαστικά και αναντικατάστατα από την περαιτέρω πρόοδο των ευφυών συστημάτων παρακολούθησης και πρόβλεψης της θαλάσσιας κυκλοφορίας. Επιπλέον, τα ευφυή συστήματα θαλάσσιας επιτήρησης μπορούν να εκτιμήσουν τη ροή της κυκλοφορίας, να αξιολογήσουν τη συγχώνευση πληροφοριών και να ανιχνεύσουν επικίνδυνες περιστάσεις. Είναι ζωτικής σημασίας για μελλοντικές καινοτομίες, αποτελεσματική πλοήγηση και ασφάλεια κυκλοφορίας.

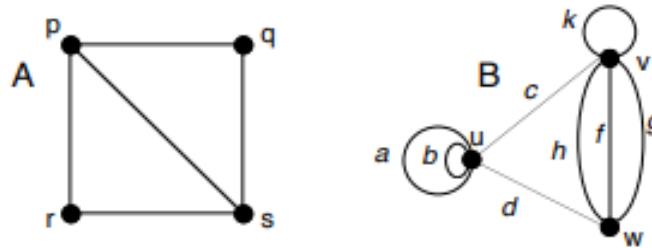
2.2 Βασικές έννοιες

Δίκτυο

Ένα δίκτυο ορίζεται ως μια συλλογή κόμβων και διακλαδώσεων με τον τύπο $G = (V, A)$, όπου V είναι η συλλογή των κόμβων και A είναι η συλλογή των διακλαδώσεων.

Γράφος

Ένας γράφος $G = (V, E)$ είναι μια μαθηματική δομή που αποτελείται από δύο πεπερασμένα σύνολα V και E . Τα στοιχεία του V ονομάζονται κορυφές (ή κόμβοι) και τα στοιχεία του E ονομάζονται ακμές. Κάθε άκρη έχει ένα σύνολο από μία ή δύο κορυφές που συνδέονται με αυτήν και ονομάζονται τελικά σημεία του.



Εικόνα 2. Γραμμικά σχέδια[27]

Τα σύνολα κορυφών και ακμών του γραφήματος A δίνονται από

$$V_A = \{p, q, r, s\} \text{ και } E_A = \{pq, pr, ps, rs, qs\}.$$

Τα σύνολα κορυφών και ακμών του γραφήματος B δίνονται από

$$V_B = \{u, v, w\} \text{ και } E_B = \{a, b, c, d, f, g, h, k\}.$$

Μονοπάτι

Σε ένα μη κατευθυνόμενο δίκτυο, μια διαδρομή είναι μια συλλογή από κοντινούς κλάδους και κόμβους. Οι διαδρομές σε ένα κατευθυνόμενο δίκτυο έχουν επίσης διευθύνσεις. Μια σειρά γειτονικών κόμβων ή διακλαδώσεων μπορεί να χρησιμοποιηθεί για να υποδείξει μια διαδρομή. Ένα μονοπάτι είναι κλειδί εάν κάθε κόμβος εμφανίζεται μόνο μία φορά στην ακολουθία και εάν κάθε κλάδος εμφανίζεται μόνο μία φορά.

2.3 Το πρόβλημα της συντομότερης διαδρομής

Ένα δίκτυο οδικής κυκλοφορίας αντιπροσωπεύεται από ένα δίγραμμα $G=(N, A)$ που αποτελείται από ένα σύνολο κόμβων N και ένα σύνολο τόξων A . Ένας σύνδεσμος $a = (i, j) \in A$ κατευθύνεται από τον κόμβο i στον κόμβο j και έχει ένα σχετικό γενικευμένο κόστος c_{ij} . Το γενικευμένο κόστος, το οποίο τυπικά εκφράζεται σε όρους χρόνου διαδρομής ζεύξης, μήκος ζεύξης, χρεώσεις κ.λπ., δείχνει την αντίσταση ενός μεμονωμένου οχήματος που διέρχεται από αυτήν τη ζεύξη. Μια διαδρομή είναι μια διαδοχική λίστα συνδέσμων που συνδέει μια προέλευση (o) και έναν προορισμό (d). Ο συνολικός χρόνος ταξιδιού της διαδρομής είναι το άθροισμα των χρόνων ταξιδιού σε κάθε μεμονωμένο σύνδεσμο. Η εύρεση της διαδρομής που χρειάζεται συνολικά τον λιγότερο χρόνο για να φτάσουμε από τον κόμβο προέλευσης στον κόμβο προορισμού είναι το πρόβλημα.

Η πλειονότητα της έρευνας σε αυτόν τον τομέα αφορούσε τη δημιουργία ολοένα και πιο αποτελεσματικών βέλτιστων αλγορίθμων για την αντιμετώπιση του ζητήματος λόγω της υπολογιστικής τους ικανότητας προσέλκυσης. Η συντριπτική πλειοψηφία των καλύτερων αλγορίθμων συντομότερης διαδρομής είναι βασικά η θεωρία δυναμικού προγραμματισμού που εφαρμόζεται στο πρόβλημα εύρεσης της συντομότερης διαδρομής σε ένα γράφημα. Από τον κόμβο προέλευσης (ή τον κόμβο προορισμού) στον κόμβο-στόχο, χρησιμοποιείται μια αναδρομική διαδικασία λήψης αποφάσεων για τον προσδιορισμό της συντομότερης διαδρομής (ή κόμβου προέλευσης). Η πλειονότητα των αλγορίθμων για την εύρεση της συντομότερης διαδρομής ακολουθούν μια κοινή διαδικασία.

Ορισμένες σημειώσεις εισάγονται για να εξηγήσουν αυτή τη διαδικασία. Το $L(i)$ είναι ο ορισμός του κόστους διαδρομής, το οποίο είναι επίσης γνωστό ως "ετικέτα" του κόμβου, από τον κόμβο προέλευσης σε έναν συγκεκριμένο κόμβο i . Το $P(i)$ σημαίνει τον προηγούμενο σύνδεσμο στη συντομότερη διαδρομή προς τον κόμβο i και το P είναι η λίστα που καταγράφει τους προηγούμενους συνδέσμους στο δέντρο της συντομότερης διαδρομής σε κάθε κόμβο. Το σύνολο κατάλληλων κόμβων σάρωσης ή Q , διατηρεί τους κόμβους που θα εξεταστούν σε όλη τη διαδικασία αναζήτησης.

Υποθέτοντας ότι ο αλγόριθμος ξεκινά από τον κόμβο προέλευσης, τα ακόλουθα είναι τα τυπικά βήματα των αλγορίθμων της συντομότερης διαδρομής:

Αρχικοποίηση: Set $i = 0$; $L(i) = 0$; $L(j) = \infty \forall j = i$; $P(i) = \text{NULL}$.

Καθορίστε το σύνολο κόμβων που πληρούν τις προϋποθέσεις
σάρωσης $Q = \{i\}$;

Επιλογή κόμβου: Επιλέξτε και αφαιρέστε έναν κόμβο (i) από το Q .

Επέκταση κόμβου: Σαρώστε κάθε σύνδεσμο που προέρχεται από τον κόμβο i .

For each link $a = (i, j)$

If

$L(i) + ca < L(j)$

then

$L(j) = L(i) + ca$; $P(j) = i$

Insert node j into Q

Κανόνας διακοπής: If $Q = \emptyset$ then STOP.

otherwise: go to step 2.

Οι κύριες διαφορές μεταξύ των διαφόρων τεχνικών έγκεινται στη δομή δεδομένων που χρησιμοποιείται για τη δημιουργία του συνόλου των κατάλληλων κόμβων σάρωσης και στη μέθοδο που χρησιμοποιείται για τον εντοπισμό και την επιλογή των κόμβων για εξέταση.

2.4 Αλγόριθμοι διαδρομής

Οι πιο συχνά χρησιμοποιούμενοι αλγόριθμοι εύρεσης μονοπατιών σε έναν δισδιάστατο κόσμο είναι ο αλγόριθμος Dijkstra και ο A*. Αν και αυτοί οι δύο αλγόριθμοι είναι αναμφισβήτητα οι πιο δημοφιλείς, υπάρχουν πολλοί ακόμη που χρησιμοποιούνται για εύρεση μονοπατιού. Κάποιοι από αυτούς είναι οι αλγόριθμοι Visibility graph και Voronoi diagram.

Η χρήση αλγορίθμων εύρεσης μονοπατιών απαιτεί περαιτέρω την εξέταση ακόμη περισσότερων παραγόντων, όπως το σύστημα πρόβλεψης κίνησης ενός πλοίου και το μοντέλο υδροδυναμικής ελιγμών. Θα πρέπει να τονιστεί ότι ένα μη επανδρωμένο σκάφος, για το οποίο η τεχνολογία ερευνάται και αναπτύσσεται, δεν μπορεί να εφαρμόσει τυφλά το αποτέλεσμα ενός μαθηματικού αλγορίθμου, ειδικά εάν δεν έχει την απαραίτητη κατανόηση για έναν αξιωματικό ναυσιπλοΐας.

2.4.1 Αλγόριθμος Dijkstra

Οι αλγόριθμοι συντομότερης διαδρομής συνήθως εκμεταλλεύονται την ιδιότητα ότι μια συντομότερη διαδρομή μεταξύ δύο κορυφών περιέχει άλλες συντομότερες διαδρομές εντός. Αυτή η ιδιότητα βέλτιστης υποδομής είναι χαρακτηριστικό της δυνατότητας εφαρμογής τόσο του δυναμικού προγραμματισμού όσο και του άπληστου. Στην πραγματικότητα, ο αλγόριθμος Dijkstra είναι ένας άπληστος αλγόριθμος, ο οποίος βρίσκει τα συντομότερα μονοπάτια μεταξύ όλων των ζευγών κορυφών. Μόνο όταν όλα τα βάρη τόξων είναι μη αρνητικά, θα είναι ακριβής ο αλγόριθμος Dijkstra για τον προσδιορισμό των συντομότερων μονοπατιών προς όλους τους κόμβους από έναν μόνο κόμβο πηγής σε ένα δίκτυο.

Ο αλγόριθμος Dijkstra χρησιμοποιείται για την εύρεση της συντομότερης διαδρομής μεταξύ οποιωνδήποτε δύο διασταυρώσεων. Χρησιμοποιεί τον άπληστο βρόχο για να βρει την ελάχιστη απόσταση.

Ακολουθούν τα βήματα για την εφαρμογή του αλγόριθμου του Dijkstra:

1. Εισάγουμε σε κάθε κόμβο μια τιμή απόστασης. Ρυθμίζουμε στο μηδέν για τον αρχικό μας κόμβο και στο άπειρο για όλους τους άλλους κόμβους.
2. Επισημαίνουμε όλους τους κόμβους ως μη επισκέψιμους. Ορίζουμε τον αρχικό κόμβο ως τρέχον.
3. Εξετάζουμε κάθε έναν από τους ανεξερεύνητους γείτονες του τρέχοντος κόμβου και προσδιορίζουμε την απόσταση μεταξύ τους (από τον αρχικό κόμβο). Για παράδειγμα, εάν η απόσταση μεταξύ του τρέχοντος κόμβου (A) και της ακμής που τον ενώνει με έναν άλλο κόμβο (B) είναι 2, η απόσταση μεταξύ B και A θα είναι $6 + 2 = 8$. Αντικαθιστούμε την απόσταση εάν είναι μικρότερη από αυτό που είχε καταγραφεί προηγουμένως (άπειρο στην αρχή, μηδέν στον πρώτο κόμβο).
4. Όταν τελειώσουμε με την εξέταση όλων των γειτόνων του τρέχοντος κόμβου τον χαρακτηρίζουμε ως επισκέψιμο. Ένας κόμβος που ελέγξαμε δεν θα ελεγχθεί ξανά. Η απόσταση που καταγράφεται τώρα είναι τελική και ελάχιστη.
5. Ορίζουμε τον μη επισκέψιμο κόμβο με τη μικρότερη απόσταση (από τον αρχικό κόμβο) ως τον επόμενο "τρέχοντα κόμβο" και συνεχίζουμε από το τρίτο βήμα.

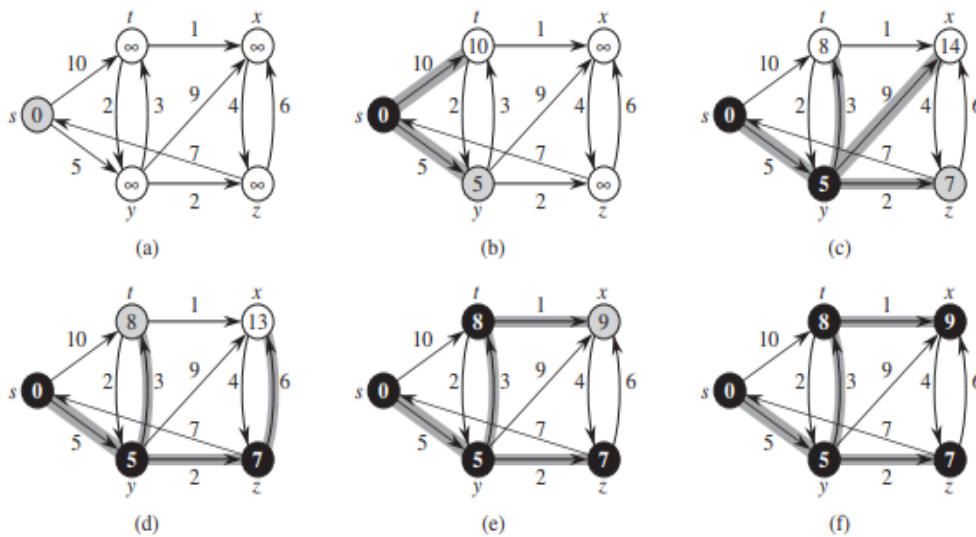
Προκειμένου να διατηρηθούν τα τελικά βάρη της συντομότερης διαδρομής από την πηγή s , ο αλγόριθμος του Dijkstra διατηρεί ένα σύνολο κορυφών S . Ο αλγόριθμος επιλέγει επανειλημμένα την κορυφή $u \in V - S$ με την εκτίμηση της συντομότερης διαδρομής, προσθέτει το u στο S , χαλαρώνει όλες τις ακμές και αφήνει το u . Στη μέθοδο που ακολουθεί, χρησιμοποιούμε μια ουρά ελάχιστης προτεραιότητας κορυφής Q με τις τιμές d να χρησιμεύουν ως κλειδιά.

DIJKSTRA

```
1 INITIALIZE-SINGLE-SOURCE( $G, s$ )
2  $S = \emptyset$ 
3  $Q = G.V$ 
4 while  $Q \neq \emptyset$ 
5      $u = \text{EXTRACT-MIN}(Q)$ 
6      $S = S \cup \{u\}$ 
7     for each vertex  $v \in G.Adj(u)$ 
8         RELAX( $u, v, w$ )
```

Η γραμμή 2 αρχικοποιεί το σύνολο S στο κενό σύνολο αφού η γραμμή 1 αρχικοποιεί το d και τις τιμές ως συνήθως. Ο βρόχος while των γραμμών 4–8 διατηρεί το αμετάβλητο ότι $Q = V - S$ στην αρχή κάθε επανάληψης. Η ουρά ελάχιστης προτεραιότητας Q αρχικοποιείται στη γραμμή 3 για να περιλαμβάνει όλες τις κορυφές στο V . Επειδή $S = \emptyset$ σε εκείνο το σημείο, η αναλλοίωτη ισχύει μετά τη γραμμή 3. Οι γραμμές 5 και 6 του βρόχου while των γραμμών 4 έως 8 διατηρούν την αμετάβλητη παίρνοντας μια κορυφή u από $Q = V - S$ και προσθέτοντάς την στο σύνολο S .

Η εκτίμηση της συντομότερης διαδρομής κάθε κορυφής στο $V - S$ είναι συνεπώς η ελάχιστη για την κορυφή u . Εάν μπορούμε να κάνουμε το συντομότερο μονοπάτι προς το v που βρέθηκε μέχρι τώρα, διανύοντας καλύτερα το u , οι γραμμές 7-8 χαλαρώνουν κάθε άκρο(u, v), ενημερώνοντας την εκτίμηση $v.d$ και τον προκατόχο v .



Εικόνα 3. Η διαδικασία εφαρμογής του αλγορίθμου του Dijkstra[7]

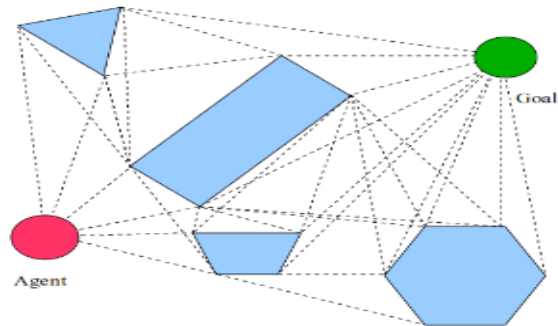
Η αριστερή κορυφή είναι η πηγή s . Εντός των κορυφών βρίσκονται οι εκτιμήσεις της συντομότερης διαδρομής, ενώ οι έγχρωμες ακμές αντιπροσωπεύουν προηγούμενες τιμές. Οι λευκές κορυφές βρίσκονται στην ουρά ελάχιστης προτεραιότητας $Q = V - S$, ενώ οι μαύρες κορυφές βρίσκονται στο σύνολο S . (a) Η περίπτωση λίγο πριν από την αρχική επανάληψη του βρόχου while στις γραμμές 4–8. Η κορυφή u στη γραμμή 5 είναι η έγχρωμη κορυφή με τη χαμηλότερη τιμή d . Η κατάσταση μετά από κάθε επανάληψη του βρόχου while εμφανίζεται στο (b)–(f). Στη γραμμή 5 της επόμενης επανάληψης, η σκουρόχρωμη κορυφή σε κάθε τμήμα επιλέγεται για να χρησιμεύσει ως κορυφή u . Οι τελικές τιμές αντιπροσωπεύονται από τις τιμές d και τις προκατόχους που φαίνονται στην ενότητα (f).

2.4.2 Μέθοδος *visibility graph*

Το θεμελιώδες δόγμα της τεχνικής *Visibility Graph* είναι ότι κάθε εμπόδιο, ανεξαρτήτως σχήματος, στην περιοχή όπου ταξιδεύει ένα όχημα θεωρείται ότι αποτελείται από πολύγωνα ή πολύεδρα. Η μέθοδος καθορίζει αποτελεσματικά την καλύτερη πορεία κίνησης του υποκειμένου στο επίπεδο όπου βρίσκονται τα εμπόδια. Ο συνδυασμός κορυφών των οποίων η πρόσπτωση καθορίζεται από την απαίτηση "ορατότητας" έχει ως αποτέλεσμα τη διαμόρφωση της διαδρομής. Κατά τη σύνδεση των κορυφών του οχήματος, των εμποδίων και των σημείων στόχου πρέπει να χρησιμοποιούνται αυτές οι πληροφορίες, διασφαλίζοντας ότι δεν διασχίζουν το ένα το άλλο συνδετικές γραμμές ή εμπόδια.

Το γράφημα ορατότητας κατασκευάζεται με αναδρομικό τρόπο, ξεκινώντας από την αρχική κορυφή. Η αξιολόγηση του άκρου, που είναι η αρχή του κλάδου, και του κλάδου που θα κατευθυνθεί σε αυτόν, αθροίζονται όταν σχηματίζεται για πρώτη φορά η κορυφή που είναι το τέλος του κλάδου. Δεδομένου ότι όλοι αυτοί οι δρόμοι είναι ορατές ευθείες γραμμές που δημιουργήθηκαν με την υπόθεση ότι δεν θα συναντηθούν εμπόδια, είναι όλοι πιθανοί δρόμοι στο περιβάλλον, εφόσον μπορούν να οδηγήσουν από το ένα σημείο στο άλλο. Η ιδανική διαδρομή για αναζήτηση είναι αυτή με τη μικρότερη απόσταση. Αυτή η προσέγγιση καταλήγει σε μια διαδρομή που διασχίζει την περίμετρο του εμποδίου. Επειδή θα είναι πολύ κοντά στο εμπόδιο όταν το περάσει, το όχημα θα στερείται ευελιξίας. Αυτό ισχύει ιδιαίτερα όταν αντιμετωπίζει το εμπόδιο σε ένα τόξο και πρέπει επίσης να κάνει ελιγμούς σε κοντινή απόσταση από αυτό. Το ποσοστό ανοχής σφαλμάτων θα είναι χαμηλό όταν οι πληροφορίες θέσης του εμποδίου δεν κατανοηθούν σωστά ή αλλάξει η κατάσταση κίνησής του.

Η δυνατότητα της μεθόδου γραφήματος ορατότητας να χρησιμοποιείται σε δυναμικά περιβάλλοντα είναι ένα από τα πλεονεκτήματά της.



Εικόνα 4. Γράφημα ορατότητας[26]

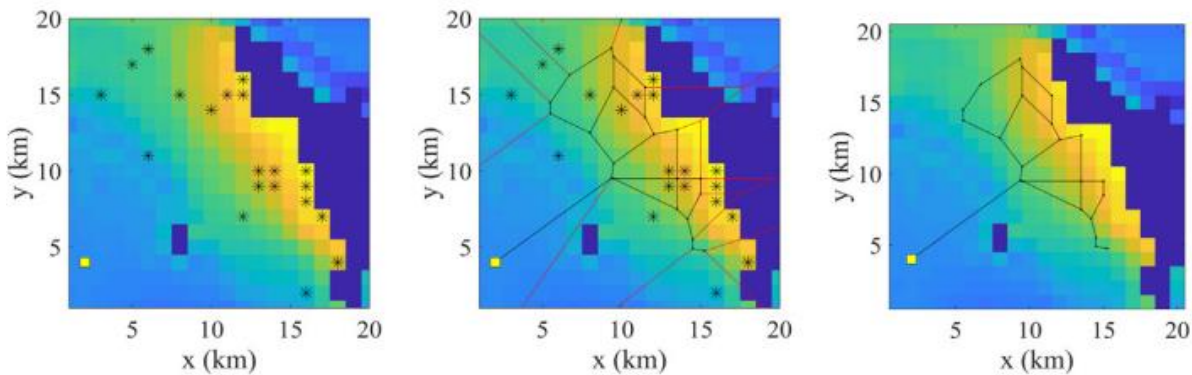
2.4.3 Διάγραμμα Voronoi

Το διάγραμμα Voronoi είναι ένα εργαλείο που χρησιμοποιείται για τον διαχωρισμό ενός γεωμετρικού χώρου σε έναν αριθμό περιοχών με βάση τον τρόπο κατανομής των αντικειμένων σε αυτήν την περιοχή. Πολλοί επιστήμονες μπόρεσαν να χρησιμοποιήσουν αυτό το όργανο σε ποικίλους τομείς και εφαρμογές λόγω αυτής της περιεκτικής μεθοδολογίας. Η ικανότητα του διαγράμματος Voronoi να δημιουργεί έναν οδικό χάρτη με άκρες που είναι σε μέγιστη απόσταση από τα σημεία γεννήτριας το καθιστά πολύτιμο σε εφαρμογές όπως ο σχεδιασμός διαδρομής και κυρίως ο σχεδιασμός διαδρομής για θαλάσσια οχήματα.

Προκειμένου να χωριστεί ο χάρτης περιβάλλοντος ωκεανού σε έναν οδικό χάρτη με ακμές, χρησιμοποιείται ως εργαλείο ένα διάγραμμα Voronoi. Οι συντεταγμένες $pg=\{p1, p2, . . . , pm\}$ χρησιμοποιούνται για τη δημιουργία του διαγράμματος Voronoi. Οι θέσεις παραγωγής στον χάρτη των ωκεανών σημειώνονται με pm . Πρακτικά, όσο περισσότερα άκρα Voronoi υπάρχουν σε περιοχές υψηλού επιστημονικού ενδιαφέροντος, τόσο πιο πολύτιμα δεδομένα θα συγκεντρωθούν.

Οι κορυφές γεννήτριας του διαγράμματος Voronoi χαρακτηρίζονται ως οι κορυφές των εμποδίων. Το διάγραμμα Voronoi περιορίζεται στην περιοχή ενδιαφέροντος από έναν αριθμό τυχαίων κορυφών που δημιουργούνται στα άκρα του σεναρίου χωρίς εμπόδια.

Τα υποψήφια σημεία διαδρομής βρίσκονται στα ακραία σημεία των άκρων του διαγράμματος Voronoί. Η διαδρομή που αντιπροσωπεύεται από τα σημεία είναι αυτή που προσφέρει ο οδικός χάρτης ως η συντομότερη. Το επανασχεδιασμένο διάγραμμα προκύπτει από την εξάλειψη των άκρων και των κορυφών Voronoί που είναι πολύ κοντά στα εμπόδια ή που παραβιάζουν τους καθορισμένους περιορισμούς απόστασης. Μετά τον προσδιορισμό των σημείων γεννήτριας pg, το πεδίο του ωκεάνιου περιβάλλοντος θα διαιρεθεί χρησιμοποιώντας τη μέθοδο διαχωρισμού Voronoί προκειμένου να παρασχεθεί ένας οδικός χάρτης για την επακόλουθη βελτιστοποίηση. Οι ακμές στο διάγραμμα Voronoί σχηματίζονται από τις κάθετες διχοτόμους κάθε σημείου γεννήτριας. Ένα τμήμα των άκρων του Voronoί εκτείνεται πέρα από τα όρια. Για να δημιουργηθεί ένας οδικός χάρτης χωρίς συγκρούσεις, αυτές οι μη πρακτικές άκρες του Voronoί θα αποκοπούν.



Εικόνα 5. Δημιουργία ενός γενικού σχεδίου ωκεάνιου οικοσυστήματος με βάση το διάγραμμα Voronoί[14]

Στην πρώτη εικόνα ο χάρτης ωκεανού δείχνει έναν μαύρο αστερίσκο για τα σημεία γεννήτριας (PG). Τα κίτρινα τετράγωνα αντιπροσωπεύουν την αρχική θέση του αυτόνομου θαλάσσιου οχήματος. Στα βαθύτερα μπλε τμήματα βρίσκονται τα στατικά εμπόδια. Στην δεύτερη εικόνα παρουσιάζεται το αρχικό διάγραμμα Voronoί. Τα πιθανά μονοπάτια συνδέονται μεταξύ τους με μια μαύρη γραμμή. Τα ανέφικτα μονοπάτια υποδεικνύονται από μια κόκκινη γραμμή στις άκρες των voronoί που περιφέρονται γύρω από εμπόδια ή έξω από το ωκεάνιο πεδίο. Στην Τρίτη εικόνα απεικονίζονται τα πιθανά μονοπάτια τα οποία συνδέονται μεταξύ τους με μια μαύρη γραμμή.

2.5 Ευρετικοί αλγόριθμοι συντομότερης διαδρομής

Σε γνήσια δίκτυα κυκλοφορίας, οι καλύτερες μέθοδοι συντομότερης διαδρομής είναι συχνά υπερβολικά υπολογιστικά απαιτητικές για εφαρμογές one-to-one σε πραγματικό χρόνο. Αυτή η "αναποτελεσματικότητα" προκύπτει από τους αλγόριθμους που χρησιμοποιούν "μη ενημερωτικές" προσεγγίσεις εξωτερικής αναζήτησης χωρίς να χρησιμοποιούν προηγούμενη γνώση των θέσεων των κόμβων προέλευσης και προορισμού, της σύνθεσης της διαδρομής και της τοπολογίας του δικτύου. Για παράδειγμα, εάν ο κόμβος προέλευσης βρισκόταν στο κέντρο της πόλης και ο κόμβος προορισμού βρισκόταν στον νότο, οι βέλτιστες τεχνικές αναζήτησης θα ήταν εξίσου πιθανό να αναζητήσουν την ελάχιστη διαδρομή βόρεια του κόμβου προέλευσης όπως θα αναζητούσαν νότια του κόμβου προέλευσης.

Η χρήση περισσότερων δεδομένων στη διαδικασία αναζήτησης θα αυξήσει την αποτελεσματικότητα των αλγορίθμων. Οι ερευνητές στον τομέα της τεχνητής νοημοσύνης (AI) κατάλαβαν πολύ γρήγορα αυτό το τελευταίο σημείο και παρουσιάστηκε ένας αριθμός ευρετικών που στόχευαν να χρησιμοποιήσουν διαφορετικές πηγές πρόσθετης γνώσης για να μειώσουν τις προσπάθειες αναζήτησης.

Οι ευρετικές στρατηγικές αναζήτησης μπορούν να ταξινομηθούν σε τέσσερις στρατηγικές: περιορισμός της περιοχής αναζήτησης, αποσύνθεση του προβλήματος αναζήτησης, περιορισμός των συνδέσμων που αναζητήθηκαν και ορισμένοι συνδυασμοί των παραπάνω.

2.5.1 Αλγόριθμος A-star(A^*)

Ο αλγόριθμος A-star προτάθηκε για την εύρεση διαδρομής ελάχιστου κόστους από το σημείο εκκίνησης μέχρι το τελικό σημείο. Λόγω της πληρότητας, της βελτιστοποίησης και της αποτελεσματικότητάς του, ο αλγόριθμος αναζήτησης διαδρομής A^* είναι ο πιο συχνά χρησιμοποιούμενος αλγόριθμος εύρεσης μονοπατιών.

Η λογική δημιουργίας του αλγορίθμου A^* είναι παρόμοια με αυτή του Dijkstra. Σε κάθε ανοιχτό κόμβο δίνεται ένα βάρος από το A^* που είναι ίσο με το βάρος της ακμής που συνδέεται με αυτόν συν την πλησιέστερη απόσταση από αυτόν τον κόμβο στο τελείωμα. Ο τυπικός αλγόριθμος A-Star είναι μια κοινή ευρετική τεχνική αναζήτησης για τον σχεδιασμό διαδρομής. Η ευρετική καθορίζει αυτή την κατά προσέγγιση απόσταση, η οποία είναι η συντομότερη διαδρομή πεπερασμένης ακολουθίας που μπορεί να υπάρξει μεταξύ του σημείου αρχής και του τέλους.

Ως αποτέλεσμα, αφού βρει μια σύντομη διαδρομή, μπορεί να απορρίψει μεγαλύτερα μονοπάτια. Είναι ένας αλγόριθμος που βασίζεται σε χάρτη πλέγματος. Οι σχεδιαστές πλοίων χρησιμοποιούν συνήθως εργαλεία όπως ηλεκτρονικούς χάρτες και ναυτικό ραντάρ για να συλλέξουν πληροφορίες σχετικά με τις θέσεις των εμποδίων στην άμεση περιοχή και στη συνέχεια να μετατρέψουν αυτά τα δεδομένα σε χάρτη πλέγματος. Το σημείο έναρξης και το σημείο τερματισμού με τις πληροφορίες εμποδίων στον παραγόμενο χάρτη πλέγματος, επιλέγονται για να εκτελέσουν τον αλγόριθμο A^* .

Η δομή του κόμβου διαδρομής (S), η OpenList και η CloseList είναι τα τρία κύρια στοιχεία του αλγορίθμου A-Star. Οι συντεταγμένες $S(i, j)$ του κόμβου διαδρομής, $S'(I', j')$ του γονικού κόμβου και $F(S)$, το ευρετικό κόστος απόστασης του κόμβου διαδρομής, περιέχονται όλες στη δομή του κόμβου. Η τοπική ευρετική συνάρτηση κόστους απόστασης δηλώνεται ως:

$$F(S) = G(S) + H(S),$$

όπου το $G(S)$ υποδηλώνει το κόστος της διαδρομής μιας απόστασης από ένα σημείο εκκίνησης και το $H(S)$ το κόστος της διαδρομής μιας απόστασης από ένα σημείο σε ένα τελικό σημείο. Η ευρετική συνάρτηση είναι επίσης γνωστή ως συνάρτηση Μανχάταν, διαγώνια συνάρτηση ή συνάρτηση Ευκλείδειας απόστασης. Για το $H(S)$ η ακρίβεια και ο ρυθμός αναζήτησης του αλγορίθμου A-star επηρεάζονται από πολλά ευρετικά.

Τα ακόλουθα είναι τα κύρια βήματα του αλγορίθμου A-Star:

1. Μια `OpenList`, `CloseList` και `PathList` δημιουργούνται κατά την προετοιμασία και ο κόμβος `Start` προστίθεται στην `OpenList`.
2. Όταν το `OpenList` δεν είναι κενό, επιλέγεται το σημείο που διασχίζεται με τη μικρότερη τιμή F , αφαιρείται από το `OpenList`, προστίθεται στο `CloseList` και λαμβάνεται ως το τρέχον σημείο.
3. Για να μάθετε εάν ο παρακείμενος εφικτός κόμβος βρίσκεται σε `OpenList`, διασχίστε τον γειτονικό εφικτό κόμβο που αντιστοιχεί στην παρούσα θέση. Σε περίπτωση που δεν είναι, προσθέστε το στην `OpenList`, συμπεριλάβετε την τρέχουσα θέση στη λίστα διαδρομής και καθορίστε το κόστος της ευρετικής απόστασης. Υπολογίστε την τιμή G του κόμβου εάν βρίσκεται ήδη στην `OpenList`. Διαφορετικά, προσθέστε το τρέχον σημείο στο `PathList` και ενημερώστε το ευρετικό κόστος απόστασης του κόμβου.
4. Έως ότου ο τρέχων κόμβος γίνει ο τελικός κόμβος, η επανάληψη προχωρά στους επόμενους κόμβους.

Το `OpenList` κρατά τον κόμβο διαδρομής που θα επιλεγεί και το `CloseList` καταγράφει τον κόμβο διαδρομής που έχει αναζητηθεί. Όταν το τελικό σημείο εμφανίζεται στην `openList`, υποδεικνύοντας ότι έχει ανακαλυφθεί η εφικτή συντομότερη διαδρομή, ο αλγόριθμος A-Star έχει ολοκληρωθεί. Όταν η `OpenList` είναι άδεια ή το τρέχον σημείο βρίσκεται στην `CloseList`, ο αλγόριθμος A-Star διακόπτεται. Σημαίνει ότι δεν υπάρχει δρόμος προς τα εμπρός και η συντομότερη διαδρομή δεν είναι εφικτή.

2.5.2 Βελτιωμένος αλγόριθμος δυναμικής αναζήτησης A-star

Ο παραδοσιακός αλγόριθμος A-star μπορεί μόνο να αποφύγει στατικά εμπόδια. Η αποφυγή δυναμικών εμποδίων δεν σημαίνει ότι η τροχιά του αντικειμένου και οι τροχιές των δυναμικών εμποδίων δεν μπορούν να διασταυρωθούν, δηλαδή να φτάσουν στην ίδια τοποθεσία σε διαφορετικές χρονικές στιγμές. Εάν στο παραγόμενο μονοπάτι δοθεί το χαρακτηριστικό του χρόνου, η αποφυγή σύγκρουσης με δυναμικά εμπόδια μπορεί να πραγματοποιηθεί εγκαίρως. Επομένως, ακόμη και αν το πλοίο θεωρείται ότι κινείται με σταθερή ταχύτητα, είναι σημαντικό να κατασκευαστεί μια διαδρομή που να μπορεί να αποφύγει δυναμικά εμπόδια. Δεδομένου ότι ο αλγόριθμος A-star βασίζει την αναζήτησή του στον τρέχοντα κόμβο P, το μόνο που απομένει είναι να καθοριστεί ο χρόνος Tc που απαιτείται για να πάει από το σημείο εκκίνησης στον τρέχοντα κόμβο P.

Ακολουθεί ο ψευδοκώδικας του βελτιωμένου αλγορίθμου δυναμικής αναζήτησης A-star για αποφυγή συγκρούσεων.

Mark P [*start*] as *openlist*.

2: **while** *openlist* \neq *empty* do

3: Select the node P [i] from the *openlist* whose value of evaluation function F (P [i]) is smallest.

4: Mark P [i] as *closelist*.

5: **if** P [i] = P [*end*] **then**

6: **return** "path is found".

7: **else**

8: $T_c = f$ *indparent* (P [i] , P [*start*]))

- 9: Sampling the trajectory of the n known dynamic obstacles to get its Tc position $Dn[Tc]$.
- 10: **if** $Dn[Tc]$ does not belong to *openlist* **then**
- 11: Mark $Dn[Tc]$ as *closelist*.
- 12: **end if**
- 13: Select the successor node $Pi[j]$ around the node $P [i]$, and calculate $F(Pi[j])$.
- 14: **if** $Pi[j]$ belongs to *obstacle* or *closelist* node then
- 15: continue;
- 16: **end if**
- 17: Mark $Pi[j]$ as *openlist*.
- 18: **if** $Pi[j]$ belongs to *openlist* and $F(Pi[j]) < F(Pm [j])$ when $P [m]$ was marked as *closelist* **then**
- 19: Set parent node of $P [j]$ as $P [i]$, $F (P [i]) = F(Pi[j])$.
- 20: **end if**
- 21: **if** $Dn[Tc]$ does not belong to *openlist* then
- 22: Remove $Dn[Tc]$ from *closelist*.
- 23: **end if**
- 24: **end if**
- 25: **end while**
- 26: **return** "the path cannot be found".

Είναι σημαντικό να σημειωθεί ότι το T_c μπορεί να είναι το ίδιο για διάφορους κόμβους, αλλά για τον ίδιο κόμβο, το αντίστοιχο T_c θα αλλάξει όταν αλλάξει ο γονικός του κόμβος. Μετά τη δειγματοληψία της τροχιάς του δυναμικού εμποδίου, για να λάβουμε τη T_c θέση του $Dn[T_c]$, αυτά τα δυναμικά εμπόδια μπορούν να θεωρηθούν ως τα γνωστά στατικά εμπόδια όπως επισημαίνεται στην closelist. Οι διαδρομές των εμποδίων εισάγονται μόνο στην closelist.

Προκειμένου να αποφευχθούν οι τροχιές αυτών των δυναμικών εμποδίων και όχι οι διαδρομές τους, είναι επίσης απαραίτητο να αφαιρούνται οι κόμβοι τους μετά από κάθε αναζήτηση.

Στο τέλος κάθε βρόχου, είναι σημαντικό να ελέγξουμε εάν το σημείο συντεταγμένων $Dn [T_c]$ βρίσκεται ήδη στην openlist πριν προσθέσουμε ή αφαιρέσουμε στόχους δυναμικών εμποδίων από τη closelist. Ο στόχος του δυναμικού εμποδίου μπορεί να αγνοηθεί εάν είναι στην openlist. Διαφορετικά, θα οδηγήσει στο πρόβλημα της επαναλαμβανόμενης αναζήτησης.

Κεφάλαιο 3

Μηχανική Μάθηση

3.1 Ορισμός Μηχανικής Μάθησης

Η επιστήμη των υπολογιστών και η τεχνητή νοημοσύνη έχουν και οι δύο ένα πεδίο που ονομάζεται μηχανική μάθηση. Ασχολείται με τη χρήση δεδομένων και τη δημιουργία αλγορίθμων που ενισχύουν τη λειτουργικότητα ενός συστήματος. Πιο συγκεκριμένα, αφού αφομοιώσει όλα τα δεδομένα του συστήματος, ο αλγόριθμος είναι σε θέση να εντοπίσει τις τάσεις και να κάνει προβλέψεις. Μεγάλοι όγκοι δεδομένων μπορούν να υποστούν επεξεργασία από μοντέλα μηχανικής μάθησης και μπορούν να προσαρμοστούν με επιτυχία στις αλλαγές στα δεδομένα εισόδου του συστήματος.

Στον αλγόριθμο δίνεται επαρκής αριθμός δειγμάτων για να επιτρέψει την εκπαίδευση προκειμένου να βελτιωθεί. Ο αλγόριθμος είναι τυπικά επαναληπτικός, πράγμα που σημαίνει ότι κατά τη διάρκεια των μαθησιακών εποχών εξετάζει τις περιπτώσεις πολλές φορές, επομένως η πρόοδος είναι τυπικά σταδιακή.

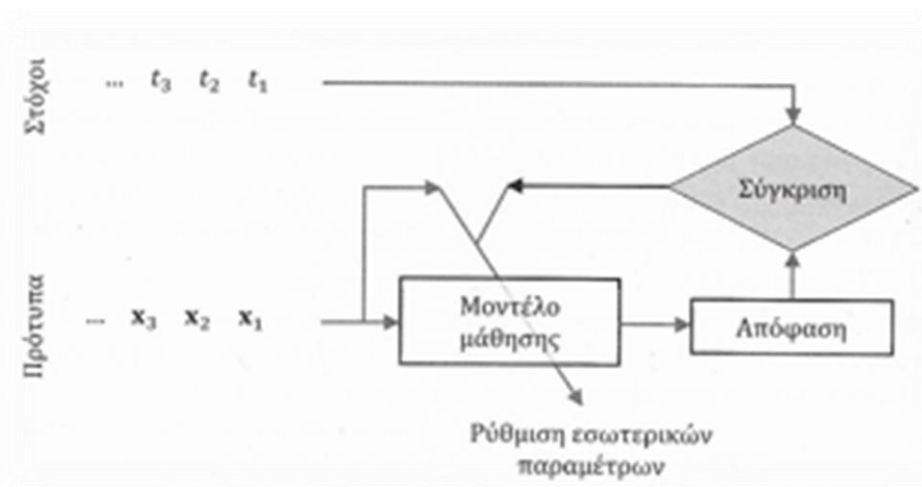
Ο βασικός στόχος της μηχανικής μάθησης είναι να επιτρέψει στο σύστημα να παράγει ακριβείς εκτιμήσεις σχετικά με τα δεδομένα που συναντά για πρώτη φορά.

3.2 Τύποι Μάθησης

Η μάθηση με επίβλεψη, η μάθηση χωρίς επίβλεψη και η ενισχυτική μάθηση είναι οι τρεις θεμελιώδεις τύποι μάθησης. Η καλύτερη από τις τρεις επιλογές επιλέγεται με βάση τον τρόπο αντιμετώπισης ενός προβλήματος.

3.2.1 Μάθηση με επίβλεψη

Η μάθηση με επίβλεψη επιλέγεται για προβλήματα όπου στόχος είναι η πρόβλεψη συγκεκριμένου αποτελέσματος. Σε αυτόν τον τύπο μάθησης το μοντέλο χρησιμοποιεί μια σειρά από πρότυπα εισόδου x_1, x_2, x_3, \dots , μαζί με τους αντίστοιχους στόχους t_1, t_2, t_3, \dots . Τα πρότυπα είναι συνήθως διανύσματα σε έναν χώρο n διαστάσεων. Το διάνυσμα x_i περιέχει τα χαρακτηριστικά x_{i1}, \dots, x_{in} του i -οστού δείγματος το οποίο καλούμαστε να αντιστοιχίσουμε στον στόχο t_i . Παρακάτω φαίνεται το διάγραμμα του συγκεκριμένου τύπου μάθησης.



Εικόνα 6. Το μοντέλο της μάθησης με επίβλεψη[17]

Οι δύο τύποι προβλημάτων μάθησης με επίβλεψη που προκύπτουν πιο συχνά στη μηχανική μάθηση είναι:

- **Προβλήματα ταξινόμησης (Classification problems):**

Οι στόχοι σε αυτά τα ζητήματα είναι κατηγορίες αντικειμένων και έχουν τη μορφή διακριτών τιμών $t_i \in \{0, 1, 2, \dots, C\}$, και αντιστοιχούν σε κλάσεις αντικειμένων. Για παράδειγμα, αν το ζητούμενο του προβλήματος είναι η ταξινόμηση ζώων σε δύο κλάσεις, σκύλος ή γάτα, θα ήταν εφικτό να θέσουμε ως στόχο το $t_i = 0$ αν το πρότυπο x_i αντιστοιχεί στον σκύλο ή $t_i = 1$ αν το x_i αντιστοιχεί στη γάτα.

- **Προβλήματα Παλινδρόμησης (Regression Problems):**

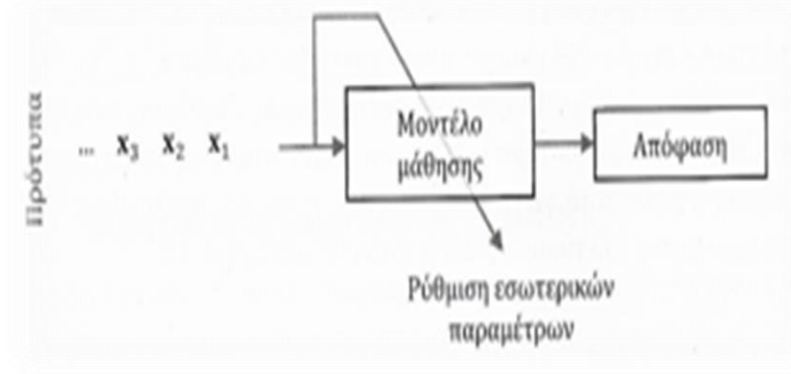
Αυτά τα προβλήματα έχουν στόχους που είναι είτε συνεχείς αριθμοί είτε μια απεριόριστη συλλογή διακριτών τιμών, δηλαδή $t_i \in \mathbb{R}$. Οι στόχοι ευθυγραμμίζονται με τις τιμές συγκεκριμένων ποσοτήτων. Προβλήματα παλινδρόμησης αποτελούν:

1. Η εύρεση του πληθυσμού μιας πόλης
2. Η πρόβλεψη ηλικίας ενός ατόμου
3. Η πρόβλεψη εισοδήματος ενός ατόμου

Στις περιπτώσεις αυτές τα αποτελέσματα θα είναι πραγματικές θετικές τιμές.

3.2.2 Μάθηση χωρίς επίβλεψη

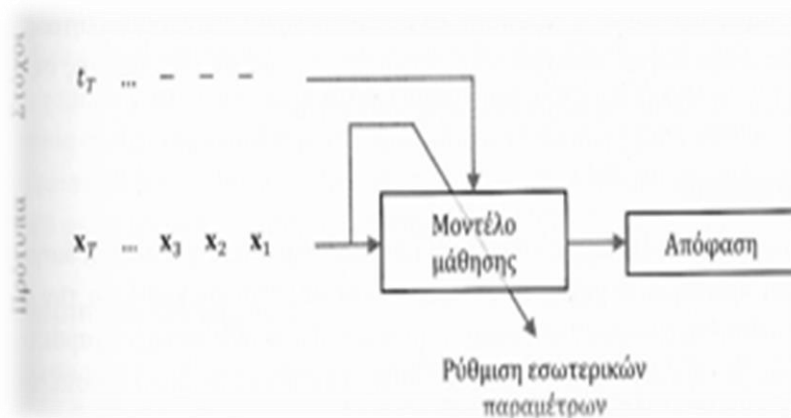
Κατά τη διαδικασία μάθησης αυτού του τύπου το μοντέλο χρησιμοποιεί πρότυπα εισόδου x_1, x_2, x_3, \dots , χωρίς όμως να διαθέτει πληροφορίες σχετικά με τους στόχους. Το μοντέλο αυτό χρησιμοποιεί μεθόδους ώστε να ανακαλύψει όσο το δυνατόν περισσότερα στοιχεία για τα δεδομένα του. Παραδείγματα μάθησης χωρίς επίβλεψη αποτελούν οι κανόνες συσχέτισης (Association rules) και η ομαδοποίηση (Clustering). Το διάγραμμα αυτού του τύπου μάθησης φαίνεται παρακάτω:



Εικόνα 7. Το μοντέλο μάθησης χωρίς επίβλεψη[17]

3.2.3 Μάθηση με ενίσχυση

Η μάθηση με ενίσχυση είναι ένας σημαντικός τύπος της μηχανικής μάθησης και χρησιμοποιείται όταν το σύστημα αλληλεπιδρά με το περιβάλλον. Οι στόχοι σε αυτό το σενάριο είναι συχνά τιμές ανταμοιβής ή τιμωρίας και το μοντέλο χρησιμοποιεί μια σειρά μοτίβων εισαγωγής, όπως x_1, x_2, x_3, \dots . Συχνά υπάρχει μία μόνο ανταμοιβή και αυτή γίνεται γνωστή στο τέλος της ακολουθίας. Το διάγραμμα αυτού του τύπου μάθησης φαίνεται παρακάτω:



Εικόνα 8. Το μοντέλο μάθησης με ενίσχυση[17]

3.3 Εφαρμογές μηχανικής μάθησης

Η μηχανική μάθηση έχει πολλές χρήσεις σε μια μεγάλη ποικιλία βιομηχανιών, όπως η υγεία, η ασφάλεια, η οικονομία, η επικοινωνία, η αυτοκινητοβιομηχανία, η ενέργεια και άλλα.

3.4 Αναγνώριση εικόνας

Η αναγνώριση εικόνας είναι μία από τις πρώτες εφαρμογές των μοντέλων μηχανικής μάθησης. Χωρίζεται σε διάφορες υποπεριπτώσεις όπως η αναγνώριση προσώπου, η αναγνώριση αντικειμένων, η αναγνώριση δακτυλικών αποτυπωμάτων, η αναγνώριση χειρόγραφου ή τυπωμένου κειμένου, η αναγνώριση ασθενειών από ιατρικές εικόνες και πολλά άλλα. Υπάρχουν πολλά μοντέλα αναγνώρισης εικόνας που βασίζονται σε χαρακτηριστικά όπως οι ακμές ή η μέση φωτεινότητα, τα οποία εξάγονται κατά τη διάρκεια μίας ξεχωριστής φάσης επεξεργασίας. Τα τελευταία χρόνια μοντέλα μηχανικής μάθησης και πιο συγκεκριμένα βαθιάς μηχανικής μάθησης, τα οποία δέχονται ως είσοδο εικόνες χωρίς ιδιαίτερη προ επεξεργασία, έχουν αναδειχθεί στα καλύτερα εργαλεία αναγνώρισης εικόνας με γενική εφαρμογή σε μεγάλη γκάμα περιπτώσεων και με εξαιρετικά αποτελέσματα σε συλλογές εικόνων πολύ μεγάλου όγκου. Η ανάκτηση εικόνων ή βίντεο με βάση το περιεχόμενο είναι επίσης μια εφαρμογή που επωφελείται από τις μεθόδους της μηχανικής μάθησης. Η δυνατότητα αυτή εφαρμόζεται ήδη σε μεγάλες βάσεις δεδομένων και προσφέρεται σαν λειτουργία από μεγάλες μηχανές αναζήτησης όπως οι Google και Bing.



Κεφάλαιο 4

Νευρωνικά δίκτυα

4.1 Ορισμός Νευρωνικών Δικτύων

Η δομή και η λειτουργία του εγκεφάλου χρησιμεύουν ως έμπνευση για την έρευνα σε τεχνητά νευρωνικά δίκτυα. Οι νευρώνες, οι οποίοι σχηματίζουν ένα δίκτυο επικοινωνίας μεταξύ τους, αποτελούν θεμελιώδες δομικό συστατικό του εγκεφάλου. Τα τυπικά νευρωνικά δίκτυα διατηρούν μόνο τις πιο χονδροειδείς ιδιότητες των μοντέλων νευρώνων που χρησιμοποιούνται στη νευροεπιστήμη, δεδομένου ότι χρησιμοποιούν απλοποιημένες εκδόσεις τέτοιων μοντέλων. Ωστόσο, ακόμη και τα μικρά μοντέλα μπορούν να παράγουν σημαντικά δίκτυα εάν πληρούν δύο βασικά κριτήρια:

- Οι νευρώνες πρέπει να έχουν ρυθμιζόμενες παραμέτρους ώστε να διευκολύνεται η διαδικασία της μάθησης.
- Το δίκτυο πρέπει να αποτελείται από μεγάλο πλήθος νευρώνων έτσι ώστε να επιτυγχάνεται ο παραλληλισμός της επεξεργασίας και η κατανομή των πληροφοριών.

Η εύρεση κατάλληλων αλγορίθμων για εκπαίδευση και ανάκτηση πληροφοριών είναι ο στόχος της έρευνας τεχνητών νευρωνικών δικτύων.



4.2 Ιστορική αναδρομή

Ο ανθρώπινος εγκέφαλος αποτελεί βασικό σημείο έρευνας εδώ και πάρα πολλά χρόνια. Η πρώτη αναφορά σε νευρωνικό δίκτυο έγινε από τον Alan Turing το 1936 και αναφερόταν στον τρόπο με τον οποίο ο ανθρώπινος εγκέφαλος χρησιμοποιεί τους νευρώνες για να κάνει υπολογισμούς. Στην συνέχεια ο John von Neumann ανέφερε έναν τρόπο υλοποίησης υπολογισμών που προκύπτει από την δομή ενός δικτύου. Έπειτα ο Norbert Wiener δήλωσε πως τόσο στον ανθρώπινο εγκέφαλο, όσο και στη μηχανική, η κατανόηση των μαθηματικών υπολογισμών, αποτελεί συνδυασμό λογικής και δικτυακής μορφής.

Η μελέτη των βιολογικών δικτύων του εγκεφάλου και η μαθηματική τους μοντελοποίηση κατέβαλε μεγάλη προσπάθεια τη δεκαετία του 1940. Οι Αμερικανοί ερευνητές McCulloch και Walter Pitts, οι οποίοι ανέπτυξαν ένα απλό μοντέλο ενεργοποίησης νευρώνων, ήταν πρωτοπόροι σε αυτόν τον κλάδο. Κατασκεύασαν ένα νευρωνικό δίκτυο με απλά ηλεκτρικά κυκλώματα και έδειξαν ότι αυτό το δίκτυο μπορεί να πραγματοποιεί υπολογισμούς.

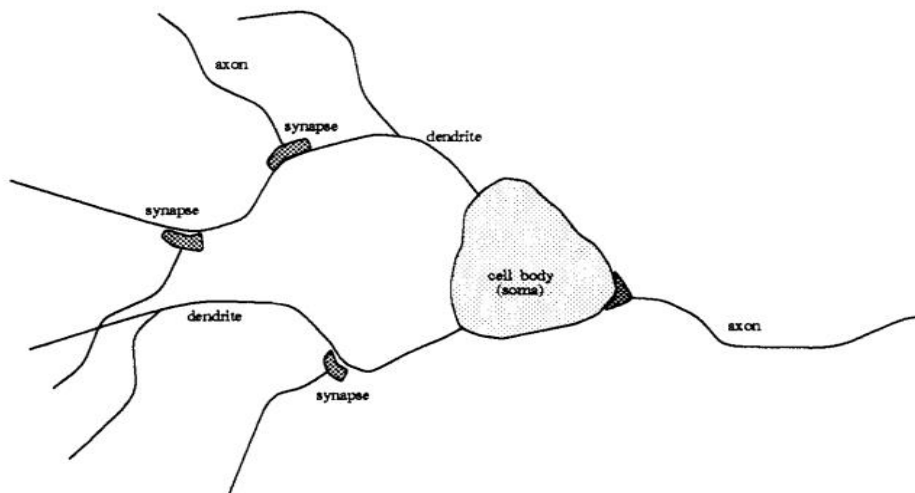
Άλλη μία πολύ σημαντική προσέγγιση για την εξέλιξη των νευρωνικών δικτύων ήταν αυτή του Donald Hebb το 1949. Ο Hebb δήλωσε πως η μάθηση επιτυγχάνεται μέσω της τροποποίησης των βαρών των συναπτικών συνδέσεων μεταξύ των νευρώνων, έτσι ώστε όταν δύο νευρώνες τείνουν να ενεργοποιηθούν ταυτόχρονα, το βάρος της μεταξύ τους σύναψης πρέπει να αυξηθεί.

Το πρώτο νευρωνικό δίκτυο κατασκευάστηκε το 1958 και ακούει στο όνομα Perceptron. Κατασκευάστηκε από τον Frank Rosenblatt ο οποίος βασίστηκε στο μοντέλο των McCulloch – Pitts. Την δεκαετία του 1980 ο John Hopfield διατύπωσε την θεωρία ότι τα νευρωνικά δίκτυα μπορούν να χρησιμοποιηθούν για την κατασκευή προηγμένων μνημών για τους υπολογιστές. Έκτοτε και έως σήμερα, τα νευρωνικά δίκτυα χρησιμοποιούνται με μεγάλη επιτυχία σε πολλές εφαρμογές και παράλληλα λειτουργούν ερευνητικά προγράμματα για την περαιτέρω εξέλιξή τους.



4.3 Δομή βιολογικού νευρώνα

Ένας νευρώνας είναι ένα μεγάλο κύτταρο με άξονα, συνάψεις, δενδρίτες και πυρήνα ή σώμα. Τα σημεία εισόδου του νευρώνα ονομάζονται δενδρίτες. Άλλοι νευρώνες τους παρέχουν ηλεκτρικές ώσεις. Η πύλη εξόδου του νευρώνα είναι ο άξονας. Επικοινωνεί με άλλους νευρώνες στέλνοντας ηλεκτρικούς παλμούς με ποικίλες συχνότητες αλλά σταθερά πλάτη. Οι συνάψεις είναι τα σημεία όπου οι δενδρίτες ενός νευρώνα συναντούν τους κλάδους του άξονα ενός άλλου νευρώνα. Το συναπτικό βάρος είναι το ποσοστό της ηλεκτρικής δραστηριότητας που μεταφέρεται στον δενδρίτη. Σύμφωνα με το εάν το φορτίο που παράγεται από τη σύναψη ενθαρρύνει τον νευρώνα να παράγει παλμούς με υψηλότερη συχνότητα ή τον αναστέλλει εμποδίζοντάς τον να δημιουργήσει ώσεις, οι συνάψεις ταξινομούνται ως διεγερτικές ή ανασταλτικές.



Εικόνα 9. Τα βασικά χαρακτηριστικά ενός βιολογικού νευρώνα[10]

Οι ηλεκτρικές ώσεις που τρέχουν κατά μήκος του άξονα κάθε νευρώνα είναι οι φορείς πληροφοριών στους βιολογικούς νευρώνες. Το ηλεκτρικό φορτίο που παίρνει κάθε νευρώνας από κάθε σύναψη συλλέγεται και δίνεται το σχετικό συναπτικό βάρος.

Όταν το συνολικό φορτίο φτάσει σε ένα συγκεκριμένο επίπεδο, ή κατώφλι, ο άξονας αρχίζει να δημιουργεί ηλεκτρικούς παλμούς υψηλής συχνότητας. Ωστόσο, εάν το φορτίο παραμένει εντός του καθορισμένου εύρους, ο νευρώνας παράγει αραιούς παλμούς σε τυχαία διαστήματα. Επομένως, ο νευρώνας θεωρείται αδρανής. Όταν το δυναμικό δράσης αυξάνει αρκετά το δυναμικό της σύναψης, εμφανίζεται η απελευθέρωση μορίων γνωστών ως νευρομεταδότες. Οι νευρομεταδότες της σύναψης διαχέονται στο διάκενο και ανοίγουν χημικά τις πύλες στους δενδρίτες, οι οποίοι όταν είναι κλειστοί επιτρέπουν τη ροή φορτισμένων ιόντων. Οι πολυάριθμες συνάψεις που ενεργοποιούνται σε κάθε δενδρίτη μπορεί να επιτρέψουν εξαιρετικά υψηλά επίπεδα διασύνδεσης. Η ποσότητα των νευρομεταδοτών που παράγονται στη συναπτική διασταύρωση καθορίζει πόσες πύλες ανοίγονται στον δενδρίτη.

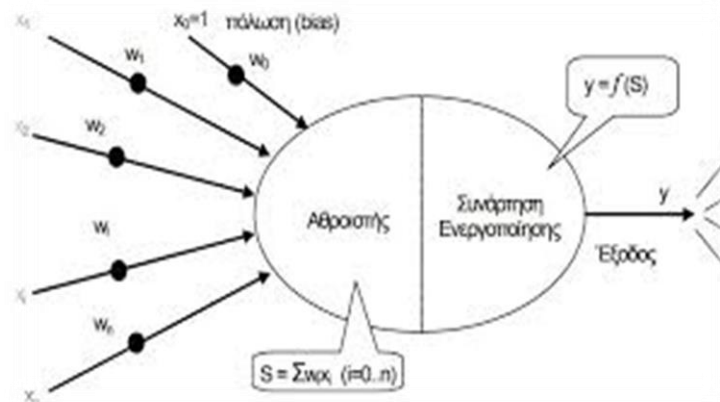
4.4 Δομή τεχνητού νευρώνα

Όπως ο βιολογικός νευρώνας, έτσι και ο τεχνητός αποτελείται από τρεις βασικούς άξονες: τον αθροιστή, τις συνάψεις και την συνάρτηση ενεργοποίησης. Συναπτικά βάρη w_i , που είναι θετικοί πραγματικοί αριθμοί για ενίσχυση των συνάψεων και αρνητικοί πραγματικοί αριθμοί για ανασταλτικές συνάψεις, χρησιμοποιούνται για την περιγραφή των συνάψεων. Έστω ότι x_1, x_2, \dots, x_n είναι οι είσοδοι του νευρώνα. Ελέγχουμε αν το άθροισμα $w_1x_1 + \dots + w_nx_n$ του φορτίου που δέχεται ο νευρώνας είναι μεγαλύτερο από ένα κατώφλι θ :

- Αν ισχύει $w_1x_1 + \dots + w_nx_n > \theta$, ο νευρώνας πυροδοτεί παλμούς
- Σε διαφορετική περίπτωση είναι αδρανής

Τα συνοπτικά βάρη δίνουν πληροφορίες για την συνεισφορά του κάθε σήματος στην δημιουργία της δομής του δικτύου. Έπειτα τα σήματα περνάνε στον αθροιστή.

Το σήμα που δημιουργείται περνάει μέσα από μια συνάρτηση ενεργοποίησης η οποία είναι επιφορτισμένη με τον περιορισμό του πλάτους εξόδου του αντίστοιχου νευρώνα.



Εικόνα 10. Διάγραμμα τεχνητού νευρώνα[17]

Ένα τεχνητό νευρωνικό δίκτυο μπορεί να έχει όσους νευρώνες απαιτούνται. Ο αριθμός των στρωμάτων είναι επίσης εναλλάξιμος. Η βασική συνάρτηση ενεργοποίησης, η οποία μπορεί να είναι γραμμική ή μη γραμμική, καθορίζει πώς συμπεριφέρεται ένα τεχνητό νευρωνικό δίκτυο.

Τα δεδομένα προεξόρυξης χρησιμοποιούνται για την εκπαίδευση του τεχνητού νευρωνικού δικτύου και τα βάρη αποθηκεύονται.

Επιπλέον, το τεχνητό νευρωνικό δίκτυο είναι ικανό να μάθει από οποιαδήποτε περίπλοκη δομή δεδομένων. Επομένως, μπορεί να εφαρμόσει μια διαδικασία μάθησης και να δημιουργήσει δυνατότητες υποστήριξης αποφάσεων για την πρόβλεψη της θαλάσσιας κυκλοφορίας.

4.5 Συναρτήσεις ενεργοποίησης

Ο νευρώνας αντιπροσωπεύεται από μια ποικιλία μοντέλων. Η μη γραμμική συνάρτηση της εξόδου, $f()$, έχει διαφορετικό σχήμα, που είναι η κύρια διάκριση. Η συνάρτηση ενεργοποίησης νευρώνων, όπως είναι επίσης γνωστή, μπορεί επίσης να έχει τις ακόλουθες μορφές:

- Βηματική συνάρτηση -1/1 (step function -1/1)

$$f(u) = \begin{cases} -1, & x \leq 0 \\ 1, & x > 0 \end{cases}$$



- Σιγμοειδής (sigmoid)

$$f(u) = \frac{1}{1+e^{-u}}$$



- Υπερβολική εφαπτομένη (hyperbolic tangent)

$$f(u) = \tanh(u) = \frac{1-e^{-u}}{1+e^{-u}}$$



- Συνάρτηση κατωφλίου (threshold function)

$$f(u) = \begin{cases} 0, & \text{αν } u \leq 0 \\ u, & \text{αν } 0 < u < 1 \\ 1, & \text{αν } u \geq 1 \end{cases}$$



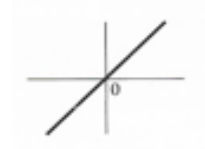
- Συνάρτηση ράμπας (ramp function)

$$f(u) = \begin{cases} -1, & u \leq 0 \\ u, & u > 0 \end{cases}$$



- Γραμμική συνάρτηση (linear function)

$$f(u) = u$$



4.6 Αρχιτεκτονική δικτύων

Η κατανομή των νευρώνων ενός νευρωνικού δικτύου και ο αλγόριθμος εκμάθησης που χρησιμοποιείται για την εκπαίδευσή του είναι άρρηκτα συνδεδεμένοι. Στη συνέχεια παρατίθενται οι κύριες αρχιτεκτονικές δικτύου.

4.6.1 Δίκτυα πρόσθιας τροφοδότησης πολλών στρωμάτων

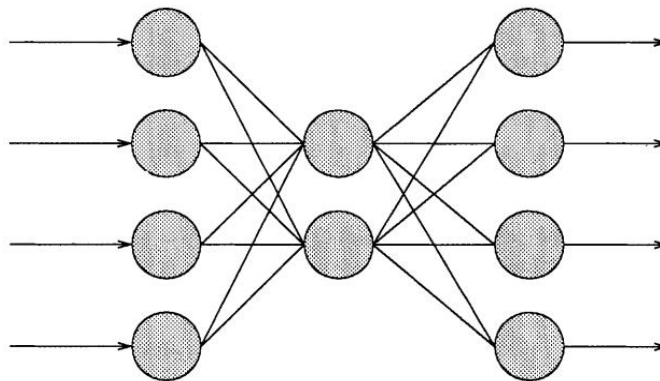
Τα δίκτυα Perceptron πολλών στρωμάτων (Multi-Layer Perceptron – MLP) μπορούν να υλοποιήσουν συναρτήσεις οι οποίες δεν είναι εφικτό να υλοποιηθούν από ένα απλό δίκτυο Perceptron. Σε αντίθεση με το απλό δίκτυο Perceptron, το οποίο μπορεί να υλοποιήσει μόνο ευθείες επιφάνειες, ένα δίκτυο MLP μπορεί να υλοποιήσει οποιαδήποτε επιφάνεια συνεχούς διαχωρισμού σε n διαστάσεις. Κάθε στρώμα ενός δικτύου MLP τροφοδοτείται αποκλειστικά από τους νευρώνες από το στρώμα κάτω από αυτό, που είναι το κύριο χαρακτηριστικό του. Το στρώμα εισόδου αναφέρεται ως μηδενικό στρώμα για ευκολία. Ως αποτέλεσμα, ο κανόνας γενικεύεται και εφαρμόζεται στο πρώτο στρώμα, το οποίο τροφοδοτείται αποκλειστικά από το μηδενικό στρώμα, το οποίο χρησιμεύει ως στρώμα εισόδου.

Η επιλογή των συναπτικών βαρών ενός δικτύου MLP με τρόπο που να ικανοποιεί ένα κριτήριο καταλληλότητας είναι γνωστή ως εκπαίδευση του δικτύου. Η εκπαιδευτική διαδικασία οποιουδήποτε νευρωνικού δικτύου έχει επίσης αυτόν τον στόχο. Το χαρακτηριστικό καθολικής προσέγγισης ενός δικτύου MLP κάνει την εκπαίδευσή του πολύ πιο συναρπαστική. Αν έχουμε το κατάλληλο δίκτυο σε μέγεθος μπορούμε να το εκπαιδύσουμε ώστε να μάθει οποιαδήποτε συνάρτηση θέλουμε με οποιαδήποτε ποιότητα προσέγγισης επιθυμούμε.

Οι εφαρμογές δικτύου MLP μπορούν να χωριστούν σε δύο κατηγορίες:

1. Προβλήματα ταξινόμησης προτύπων
2. Προβλήματα εκτίμησης συναρτήσεων

Και αφορούν τομείς όπως η οικονομία και το εμπόριο, η βιομηχανία, η ιατρική και οι υπολογιστές.



Εικόνα 11. Το πολυστρωματικό perceptron[10]

Το πολυστρωματικό perceptron αποτελείται από τρία στρώματα: ένα στρώμα εισόδου, ένα στρώμα εξόδου και ένα στρώμα που βρίσκεται στη μέση και δεν είναι άμεσα συνδεδεμένο με κανένα. Αυτό το επίπεδο είναι γνωστό ως το κρυφό στρώμα.

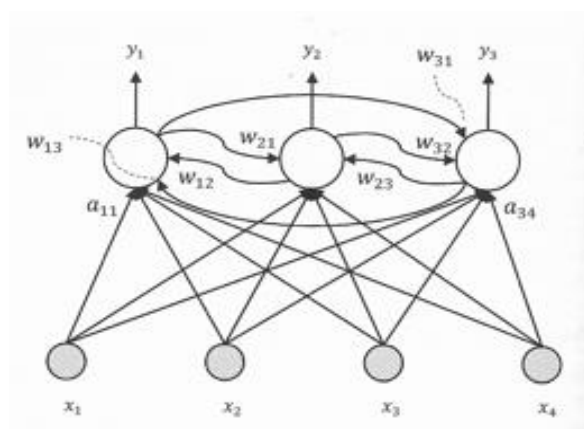
4.6.2 Αναδρομικά Νευρωνικά Δίκτυα

Ένα αναδρομικό νευρωνικό δίκτυο (Recurrent Neural Network) περιέχει νευρώνες οι οποίοι συνδέονται μεταξύ τους έτσι ώστε να δημιουργούνται κύκλοι. Η τιμή ενός νευρώνα εξαρτάται από τις τιμές των άλλων νευρώνων και αντίστροφα οι τιμές των άλλων νευρώνων εξαρτώνται από την τιμή ενός νευρώνα.

Ένα παράδειγμα αναδρομικού νευρωνικού δικτύου φαίνεται στην παρακάτω εξίσωση:

$$y_i = f\left(\sum_{j=1}^4 a_{i,j} x_j + \sum_{j=1}^3 w_{i,j} y_j + b_i\right), i = 1, 2, 3$$

Όπου $a_{i,j}$ είναι τα πρόσθια βάρη μεταξύ των εισόδων και των νευρώνων εξόδου. w_{ij} είναι τα πλάγια βάρη μεταξύ των νευρώνων εξόδου και b_i είναι οι πολώσεις. Επειδή οι άγνωστοι y_i βρίσκονται και στα δύο μέλη της εξίσωσης και η συνάρτηση f είναι μη γραμμική, δεν υπάρχει απλή λύση του συστήματος των εξισώσεων και πρέπει να δημιουργηθεί μια δυναμική προσέγγιση.

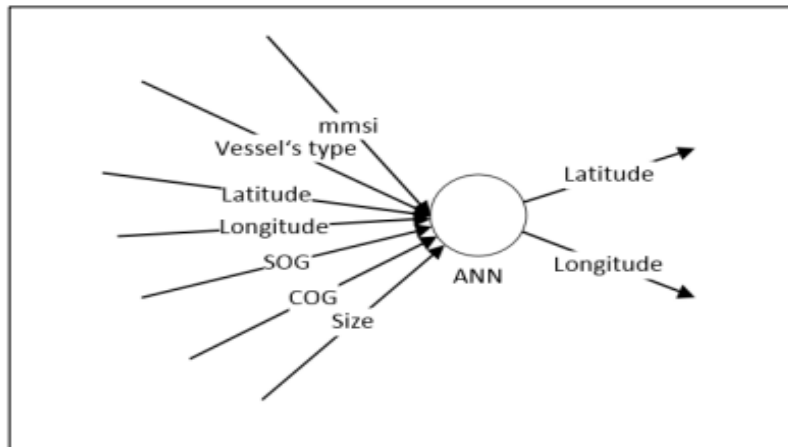


Εικόνα 12. Σχεδιάγραμμα αναδρομικού νευρωνικού δικτύου[17]

4.7 Ανάπτυξη τεχνητού νευρωνικού δικτύου για πρόβλεψη θαλάσσιας κυκλοφορίας

Για να προβλέψουμε πού θα στραφούν τα πλοία στη θαλάσσια κυκλοφορία, δημιουργήθηκε ένα μοντέλο νευρωνικού δικτύου τριών επιπέδων. Οι είσοδοι είναι: Γεωγραφικό πλάτος περιοχών, γεωγραφικό μήκος περιοχών, ταχύτητα, διαδρομή, αριθμός ταυτότητας ναυτιλιακής κινητής υπηρεσίας, διάσταση και τύπος πλοίου.

Οι έξοδοι είναι: Γεωγραφικό πλάτος της επόμενης περιοχής και γεωγραφικό μήκος της επόμενης περιοχής.



Εικόνα 13. Τεχνητό νευρωνικό δίκτυο για θαλάσσια κυκλοφορία [20]

Η έξοδος κάθε νευρώνα υπολογίστηκε από μια συνάρτηση ενεργοποίησης. Για να αναπτυχθεί το νευρωνικό δίκτυο, τα δεδομένα χωρίζονται σε δύο μέρη. Την εκπαίδευση και την επικύρωση του νευρωνικού δικτύου.

Η σχέση μεταξύ της εξόδου (y_t) και της εισόδου (y_{t-1}, \dots, y_{t-p}) έχει την ακόλουθη μαθηματική αναπαράσταση:

$$y_t = w_0 + \sum_{j=1}^Q w_j g(w_{0j} + \sum_{i=1}^p w_{ij} y_{t-1}) + e_t$$

Όπου $w_{i,j}$ ($i = 0, 1, 2, \dots, P, j = 1, 2, \dots, Q$) και w_j ($j = 0, 1, 2, \dots, Q$) αποτελούν τις παραμέτρους που ονομάζονται συχνά βάρη σύνδεσης. P είναι ο αριθμός των κόμβων εισόδου.

Η συνάρτηση ενεργοποίησης είναι:

$$\text{Sig}(x) = \frac{1}{1 + \exp(-x)}$$

Μόλις καθοριστεί μια δομή δικτύου (P, Q) το δίκτυο είναι έτοιμο για εκπαίδευση. Οι παράμετροι εκτιμώνται έτσι ώστε το κόστος λειτουργίας του νευρωνικού δικτύου να ελαχιστοποιείται.

Το κόστος λειτουργεί ως κριτήριο συνολικής ακρίβειας όπως το ακόλουθο μέσο τετραγωνικό σφάλμα:

$$E = \frac{1}{N} \sum_{n=1}^N (e_j)^2 = \frac{1}{N} \sum_{n=1}^N \left(y_t - \left(w_0 + \sum_{j=1}^Q w_j g(w_{0j} + \sum_{i=1}^P w_{ij} y_{t-1}) \right) \right)^2$$

Όπου N είναι ο συνολικός αριθμός των λαθών. Εκτός από τη βασική διαδικασία εκπαίδευσης, χρησιμοποιούνται ορισμένοι αποτελεσματικοί μη γραμμικοί αλγόριθμοι βελτιστοποίησης για να επιτευχθεί αυτή η μείωση. Αφού το δίκτυο τεχνητών νευρώνων λάβει τις εισόδους, θα τις διαδώσει από το στρώμα εισόδου μέσω των κρυφών στρωμάτων στο επίπεδο εξόδου, όπου λαμβάνονται οι απαντήσεις.

Κεφάλαιο 5

Βαθιά μάθηση

5.1 Ορισμός Βαθιάς Μάθησης

Η βαθιά μάθηση είναι ένας τύπος μηχανικής μάθησης και τεχνητής νοημοσύνης που μιμείται τον τρόπο με τον οποίο οι άνθρωποι μαθαίνουν για συγκεκριμένα θέματα. Στην επιστήμη δεδομένων, η οποία περιλαμβάνει επίσης στατιστικές και προγνωστικά μοντέλα, η βαθιά μάθηση παίζει σημαντικό ρόλο. Η βαθιά μάθηση καθιστά αυτή τη διαδικασία ταχύτερη και ευκολότερη για τους επιστήμονες δεδομένων, οι οποίοι είναι υπεύθυνοι για τη συλλογή, την ανάλυση και την ερμηνεία τεράστιου όγκου δεδομένων.

Η βαθιά μάθηση μπορεί να θεωρηθεί ένα μέσο αυτοματοποίησης της προγνωστικής ανάλυσης στο πιο βασικό της επίπεδο. Οι αλγόριθμοι βαθιάς μάθησης συσσωρεύονται σε μια ιεραρχία αυξανόμενης πολυπλοκότητας, σε αντίθεση με τους συμβατικούς αλγόριθμους μηχανικής μάθησης, οι οποίοι είναι γραμμικοί.

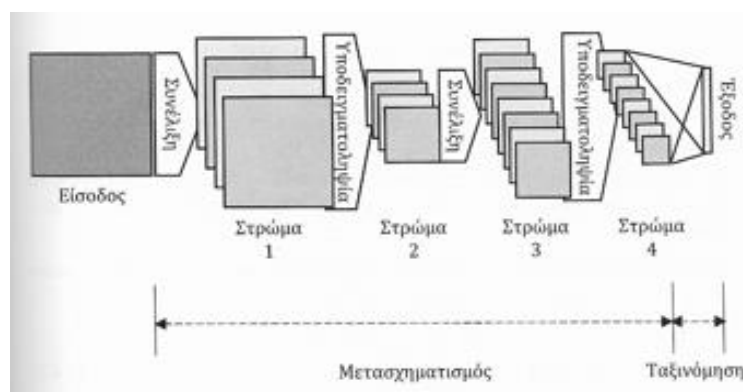
5.2 Νευρωνικά δίκτυα βαθιάς μάθησης

Η πλειοψηφία των μοντέλων βαθιάς μάθησης υποστηρίζεται από ένα τεχνητό νευρωνικό δίκτυο, ένα είδος εξελιγμένου αλγόριθμου μηχανικής μάθησης. Η βαθιά μάθηση είναι επομένως επίσης γνωστή ως βαθιά νευρωνική μάθηση ή βαθιά νευρωνική δικτύωση. Κάθε τύπος νευρωνικού δικτύου, όπως τα νευρωνικά δίκτυα τροφοδοσίας, τα επαναλαμβανόμενα νευρωνικά δίκτυα, τα συνελκτικά νευρωνικά δίκτυα και τα τεχνητά νευρωνικά δίκτυα, προσφέρουν διάφορα πλεονεκτήματα για συγκεκριμένες περιπτώσεις.

Ωστόσο, όλα λειτουργούν περίπου με τον ίδιο τρόπο καθώς όλα τα δεδομένα τροφοδοτούνται στο μοντέλο και το μοντέλο στη συνέχεια αποφασίζει μόνο του εάν έχει κάνει ή όχι τη σωστή ερμηνεία για ένα συγκεκριμένο στοιχείο δεδομένων. Δεδομένου ότι τα νευρωνικά δίκτυα μαθαίνουν κάνοντας λάθη, απαιτούν τεράστιο όγκο δεδομένων εκπαίδευσης. Τα δεδομένα που χρησιμοποιούνται κατά το στάδιο εκπαίδευσης πρέπει να φέρουν ετικέτα, ώστε το μοντέλο να μπορεί να προσδιορίσει εάν η εκτίμησή του ήταν σωστή.

5.3 Συνελικτικά νευρωνικά δίκτυα

Τα δίκτυα πολλαπλών επιπέδων που ονομάζονται συνελικτικά νευρωνικά δίκτυα (CNN) είναι κατάλληλα για εργασίες αναγνώρισης εικόνας. Αποτελούνται από στρώματα που εναλλάσσονται μεταξύ συνέλιξης και υποδειγματοληψίας. Ένα υποδίκτυο ταξινομητή αποτελείται από ένα ή περισσότερα πλήρως συνδεδεμένα επίπεδα που έρχονται μετά την τελική συνέλιξη ή το επίπεδο υποδειγματοληψίας. Η τυπική αρχιτεκτονική του μοντέλου CNN φαίνεται παρακάτω:



Εικόνα 14. Η αρχιτεκτονική ενός δικτύου CNN[17]

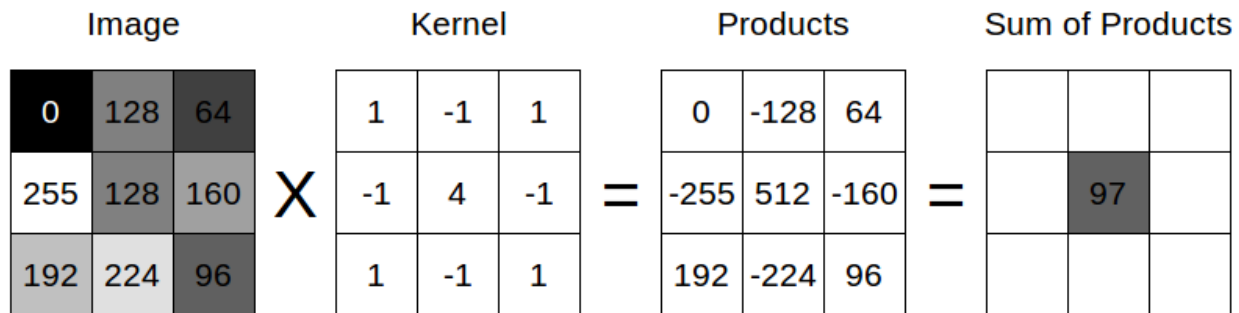
Ένα μοντέλο CNN αποτελείται από την είσοδο όπου είναι ένας δισδιάστατος πίνακας, το στρώμα 1 που είναι το στρώμα συνέλιξης, το στρώμα 2 το οποίο είναι στρώμα υποδειγματοψίας, το στρώμα 3 που είναι στρώμα συνέλιξης, το στρώμα 4 το οποίο είναι επίσης στρώμα υποδειγματοληψίας και ένα στρώμα ταξινόμησης το οποίο είναι πλήρως συνδεδεμένο με το τελευταίο στρώμα υποδειγματοληψίας.

Η έξοδος ενός συνελκτικού στρώματος ή ενός στρώματος υποδειγματοληψίας είναι μια σειρά από χάρτες χαρακτηριστικών. Κάθε χάρτης χαρακτηριστικών είναι ένα δισδιάστατο πλέγμα από νευρώνες όπου ο καθένας διεγείρεται από μια μικρή περιοχή στο προηγούμενο στρώμα. Η περιοχή καλείται τοπικό υποδεκτικό πεδίο (local receptive field) του συγκεκριμένου νευρώνα. Τα συναπτικά βάρη που συνδέουν έναν νευρώνα με τους νευρώνες του τοπικού πεδίου του είναι γνωστά ως μάσκα. Όλοι οι νευρώνες που βρίσκονται στον ίδιο χάρτη χαρακτηριστικών έχουν την ίδια μάσκα, οπότε έχουμε επανάληψη των βαρών.

5.4 Επίπεδο συνέλιξης

Κάθε συνελκτικό νευρωνικό δίκτυο έχει τουλάχιστον ένα επίπεδο συνέλιξης που βασίζεται σε φίλτρο που συνδέεται με τη διάσταση βάθους της εικόνας εισόδου και αποτελείται από συνέλιξη στα δεδομένα εισόδου. Κάθε φίλτρο προκαλεί τη δημιουργία ενός δισδιάστατου χάρτη χαρακτηριστικών μετακινώντας κατά το πλάτος και το ύψος της εικόνας. Η συνέλιξη του φίλτρου και της εικόνας εισόδου υλοποιείται από αυτή τη λειτουργία. Το τελικό αποτέλεσμα της συνέλιξης καθορίζεται από τη συνένωση των τιμών του φίλτρου και της εικόνας. Ο υπολογισμός των διαστάσεων πλάτους και ύψους εξαρτάται από δύο παράγοντες, το γέμισμα (padding) και το βήμα (stride). Το γέμισμα χρησιμοποιείται για την προσθήκη μηδενικών στην είσοδο προκειμένου να γίνει κατάλληλη επιλογή στο μέγεθος φίλτρου που θα χρησιμοποιηθεί για την σάρωση των χαρακτηριστικών.

Το βήμα καθορίζει την ένταση της δειγματοληψίας εισόδου ή την ποσότητα των εικονοστοιχείων που μετατοπίζονται τόσο οριζόντια όσο και κάθετα. Η σάρωση είναι πυκνή εάν η τιμή βήματος είναι ένα. Αν είναι μεγαλύτερη της μονάδας, τότε η σάρωση είναι αραιή.



Εικόνα 15. Πίνακας συνέλιξης[28]

5.5 Επίπεδα συγκέντρωσης

Σε πολλές περιπτώσεις, ένα στρώμα συγκέντρωσης προστίθεται μετά από ένα στρώμα συνέλιξης. Χωρίς απώλεια στον διαχωρισμό αντικειμένων, στόχος αυτού του επιπέδου είναι να κάνει το δίκτυο λιγότερο ευαίσθητο και να συρρικνώσει το μέγεθος της εικόνας εισόδου. Είναι χρήσιμο για τη σταδιακή μείωση της εικόνας εισόδου, τη συντόμευση του χρόνου εκπαίδευσης στο δίκτυο και την πρόληψη της υπερβολικής εκπαίδευσης.

Τα τέσσερα είδη συγκέντρωσης είναι η συγκέντρωση μέγιστης τιμής (max pooling), η συγκέντρωση μέσης τιμής (average pooling), η παγκόσμια μέγιστη συγκέντρωση(global max pooling) και η παγκόσμια μέση συγκέντρωση(global average pooling).

Για την υλοποίηση της εργασίας θα ασχοληθούμε με τη παγκόσμια μέση συγκέντρωση την οποία και αναλύουμε παρακάτω.

Κατά τη χρήση της συνολικής μέσης συγκέντρωσης, το μέγεθος της συγκέντρωσης εξακολουθεί να ορίζεται στο μέγεθος της εισόδου του επιπέδου, αλλά χρησιμοποιείται ο μέσος όρος της συγκέντρωσης αντί για το μέγιστο. Το τελικό αποτέλεσμα προκύπτει από το συνολικό άθροισμα των στοιχείων σε κάθε γραμμή δια το πλήθος των στοιχείων στις γραμμές.

4	3	1	5
1	3	4	8
4	5	4	3
6	5	9	4

$$\text{Avg} \left(\begin{matrix} [4, 3, 1, 5] \\ [1, 3, 4, 8] \\ [4, 5, 4, 3] \\ [6, 5, 9, 4] \end{matrix} \right) = 4.3125$$

Εικόνα 16. Παράδειγμα παγκόσμιας μέσης συγκέντρωσης[22]

Το τελευταίο επίπεδο του μοντέλου είναι ένα συνελικτικό επίπεδο που δημιουργεί τόσους χάρτες χαρακτηριστικών όσες είναι οι κλάσεις-στόχοι και εφαρμόζει μια συνολική μέση συγκέντρωση σε καθεμία για να τους συνδυάσει σε μια ενιαία τιμή. Οι χάρτες του τελικού επιπέδου μπορούν αποτελεσματικά να "μάθουν να αναγνωρίζουν" την παρουσία μιας συγκεκριμένης κλάσης σε αυτήν την αρχιτεκτονική, καθώς οι χάρτες χαρακτηριστικών μπορούν να αναγνωρίσουν συγκεκριμένα στοιχεία στα δεδομένα εισόδου. Η επιθυμητή κατανομή πιθανότητας πολλαπλών κλάσεων μπορεί να ληφθεί για άλλη μια φορά τροφοδοτώντας τα δεδομένα που παράγονται από τη συγκέντρωση του παγκόσμιου μέσου όρου σε μια συνάρτηση ενεργοποίησης.

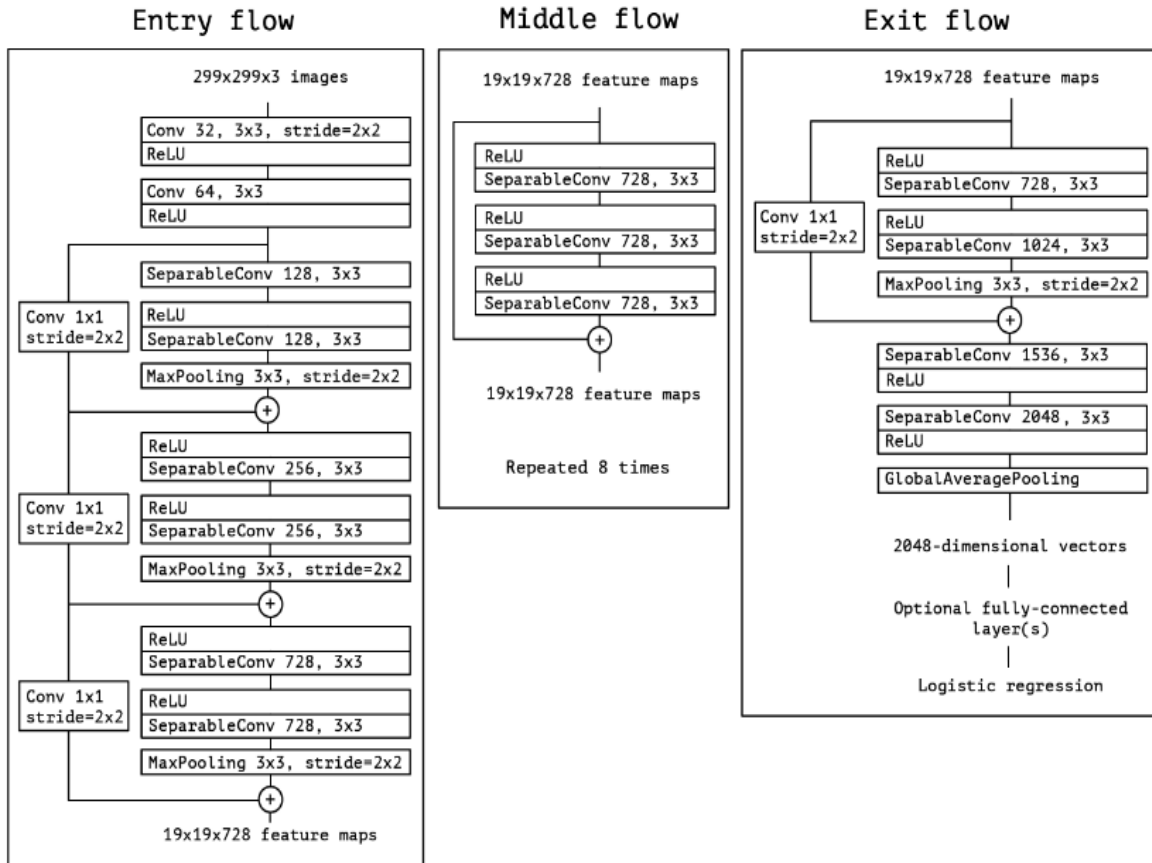
Επιπλέον, αυτή η μέθοδος μπορεί να βελτιώσει την απόδοση του μοντέλου επειδή ο "ταξινομητής" και ο "εξαγωγέας χαρακτηριστικών" είναι εγγενείς μεταξύ τους και μπορεί επίσης να αποτρέψει την υπερπροσαρμογή επειδή δεν υπάρχει παράμετρος στο επίπεδο συγκέντρωσης του παγκόσμιου μέσου όρου που πρέπει να μάθει.

5.6 Μοντέλο Xception

Το μοντέλο Xception είναι ένα βαθύ συνελκτικό νευρωνικό δίκτυο το οποίο αντικαθιστά τις μονάδες της συνηθισμένης συνέλιξης με κατά βάθος διαχωρίσιμη συνέλιξη. Η κατά βάθος διαχωρίσιμη συνέλιξη (Depthwise separable convolution) χωρίζει τον υπολογισμό σε δύο στάδια: η συνέλιξη σε βάθος χρησιμοποιεί ένα μόνο συνελκτικό φίλτρο για κάθε κανάλι εισόδου και η σημειακή συνέλιξη εφαρμόζεται για την δημιουργία ενός γραμμικού συνδυασμού. Σύμφωνα με την αρχιτεκτονική του Xception οι διακαναλικοί και χωρικοί συσχετισμοί μπορεί να διαχωριστούν πλήρως ο ένας από τον άλλο όταν αντιστοιχούν σε χάρτες χαρακτηριστικών συνελκτικών νευρικών δικτύων. Η βάση εξαγωγής χαρακτηριστικών του δικτύου στην αρχιτεκτονική Xception αποτελείται από 36 συνελκτικά επίπεδα. Υπάρχουν 14 μονάδες που αποτελούνται από τα 36 συνελκτικά στρώματα τα οποία περιβάλλονται από γραμμικές υπολειμματικές συνδέσεις, με εξαίρεση την πρώτη και την τελευταία ενότητα.

Η αρχιτεκτονική Xception είναι μια γραμμική στοίβα από υπολειμματικά συνδεδεμένα σε βάθος διαχωρίσιμα επίπεδα συνέλιξης. Αυτό καθιστά την αρχιτεκτονική σχετικά απλή στη δημιουργία και την προσαρμογή χρησιμοποιώντας ένα πλαίσιο υψηλού επιπέδου όπως το Keras ή το TensorFlow και απαιτεί συνήθως 30 έως 40 γραμμές κώδικα.

Το μοντέλο Xception προτάθηκε από τον Francois Chollet τον δημιουργό και κύριο συντηρητή της βιβλιοθήκης Keras. Κατά την χρήση του είναι απαραίτητη η δημιουργία ενός συνόλου δεδομένων για εκπαίδευση και επικύρωση. Τα δεδομένα περνούν πρώτα από τη ροή εισόδου, έπειτα από τη μεσαία ροή και τέλος από τη ροή εξόδου. Όλα τα επίπεδα, συνελκτικά και βαθιά συνελκτικά ακολουθούνται από κανονικοποίηση παρτίδας.



Εικόνα 17. Διάγραμμα μοντέλου Xception[16]

Στην αρχιτεκτονική Xception τα δεδομένα περνούν πρώτα από τη ροή εισόδου, ύστερα από τη μεσαία ροή, η οποία επαναλαμβάνεται οκτώ φορές, και μετά τη ροή εξόδου. Η κανονικοποίηση παρτίδας έρχεται μετά από κάθε επίπεδο Convolution και SeparableConvolution. Ο πολλαπλασιαστής βάθους για όλα τα επίπεδα SeparableConvolution είναι 1.

Επιλέγουμε το μοντέλο Xception γιατί παρουσιάζει μεγαλύτερη ακρίβεια σε σχέση με άλλα μοντέλα.

		Top-1 accuracy	Top-5 accuracy
VGGNet – 1 st Runner Up in ILSVRC 2014	VGG-16	0.715	0.901
ResNet – Winner in ILSVRC 2015	ResNet-152	0.770	0.933
Inception-v3 – 1 st Runner Up in ILSVRC 2015	Inception V3	0.782	0.941
	Xception	0.790	0.945

Εικόνα 18. Σύγκριση ImageNet[16]

Κεφάλαιο 6

Περιβάλλον ανάπτυξης και βιβλιοθήκες

6.1 Anaconda Jupyter Notebook

Ένα πρόγραμμα ανοιχτού κώδικα που ονομάζεται Jupyter Notebook μας δίνει τη δυνατότητα να δημιουργήσουμε και να μοιραστούμε έγγραφα με κώδικα, εξισώσεις, γραφικά και κείμενο. Ο μετασχηματισμός δεδομένων, ο καθαρισμός, η στατιστική μοντελοποίηση, η οπτικοποίηση δεδομένων, η μηχανική μάθηση και άλλες διαδικασίες είναι μεταξύ των χαρακτηριστικών του. Ο στόχος του Jupyter είναι η ανάπτυξη λογισμικού ανοιχτού κώδικα για διαδραστικούς υπολογιστές με χρήση πολλών γλωσσών προγραμματισμού.



Εικόνα 19. Σύμβολο Jupyter Notebook

6.2 TensorFlow

Η Google δημιούργησε τη βιβλιοθήκη ανοιχτού κώδικα TensorFlow ειδικά για εφαρμογές βαθιάς εκμάθησης. Οι πολυδιάστατοι πίνακες είναι ο μόνος τύπος δεδομένων που λαμβάνει το TensorFlow. Κατά το χειρισμό πολλών δεδομένων, οι πολυδιάστατοι πίνακες είναι αρκετά χρήσιμοι.

Τα γραφήματα ροής δεδομένων με κόμβους και ακμές χρησιμεύουν ως βάση για τη λειτουργία του TensorFlow. Μπορεί να θεωρηθεί ως ένα διαφοροποιήσιμο επίπεδο υποδομής προγραμματισμού. Συνδυάζει τέσσερις βασικές δεξιότητες: Την αποτελεσματική εκτέλεση λειτουργιών τανυστή χαμηλού επιπέδου σε CPU, GPU ή TPU. Τον υπολογισμό της κλίσης οποιασδήποτε διαφοροποιήσιμης έκφρασης. Τον υπολογισμό σε μεγάλο αριθμό συσκευών, όπως συμπλέγματα με εκατοντάδες GPU. Προγράμματα ("γραφήματα") μπορούν να εξαχθούν σε εξωτερικούς χρόνους εκτέλεσης όπως διακομιστές, προγράμματα περιήγησης, κινητές συσκευές και ενσωματωμένες συσκευές.



Εικόνα 20. Σύμβολο Tensorflow

6.3 Keras

Το Keras είναι μία διεπαφή προγραμματισμού εφαρμογών γραμμένο σε Python, που τρέχει πάνω από την πλατφόρμα TensorFlow. Δημιουργήθηκε με στόχο τη διευκόλυνση του γρήγορου πειραματισμού. Ικανότητα διεξαγωγής γρήγορων πειραμάτων ήταν βασικός παράγοντας κατά την ανάπτυξη. Η αποτελεσματική έρευνα απαιτεί την ικανότητα γρήγορης μετάβασης από μια ιδέα σε ένα συμπέρασμα. Το Keras είναι απλό, αλλά όχι υπερβολικά. Μειώνει τη γνωστική πίεση στους προγραμματιστές, ώστε να μπορούν να επικεντρωθούν στις κρίσιμες πτυχές του ζητήματος. Επιπλέον, το Keras είναι ευέλικτο.

Τηρεί την έννοια της προοδευτικής αποκάλυψης της πολυπλοκότητας, η οποία δηλώνει ότι οι αυθαίρετες περίπλοκες ροές εργασίας θα πρέπει να είναι εφικτές μέσω μιας σαφούς διαδρομής που βασίζεται σε όσα έχετε ήδη μάθει. Οι απλές ροές εργασίας πρέπει να είναι γρήγορες και εύκολες. Τέλος, το Keras είναι ισχυρό. Προσφέρει απόδοση και επεκτασιμότητα και χρησιμοποιείται από πολλές μεγάλες επιχειρήσεις.



Εικόνα 21. Σύμβολο Keras

Κεφάλαιο 7

Σχεδιασμός και υλοποίηση

Για την υλοποίηση του project το dataset λήφθηκε από το Kaggle. 6252 εικόνες θα χωριστούν σε δεδομένα εκπαίδευσης (train), επικύρωσης (validation) και δοκιμής (test). Επιπλέον εφαρμόζεται η διαδικασία data augmentation με σκοπό την αύξηση απόδοσης του μοντέλου. Για την εκπαίδευση χρησιμοποιείται το προ-εκπαιδευμένο μοντέλο Xception.

Αρχικά φορτώνουμε τις απαραίτητες βιβλιοθήκες.

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from IPython.display import Image
import os
```

```
In [2]: !pip install opencv-python
```

```
Requirement already satisfied: opencv-python in c:\users\pkomp\anaconda3\lib\site-packages (4.5.5.64)
Requirement already satisfied: numpy>=1.17.3 in c:\users\pkomp\anaconda3\lib\site-packages (from opencv-python) (1.20.3)
```

```
In [3]: import cv2
```

```
In [4]: from tensorflow.keras.layers import Dense, Flatten, Conv2D, MaxPool2D, GlobalAveragePooling2D, Dropout
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.applications.xception import Xception
from tensorflow.keras.optimizers import Adam
```

Η numpy παρέχει υποστήριξη για πίνακες και περιλαμβάνει μια ποικιλία μαθηματικών συναρτήσεων για εργασία με αυτές τις δομές. Η pandas παρέχει μια δομή δεδομένων για την αποτελεσματική αποθήκευση και τον χειρισμό δεδομένων σε μορφή πίνακα. Η matplotlib είναι μια βιβλιοθήκη σχεδίασης που παρέχει ένα ευρύ φάσμα γραφημάτων. Το IPython.display είναι μια λειτουργική μονάδα που παρέχει μια βολική διεπαφή για την εμφάνιση μιας ποικιλίας δεδομένων σε ένα σημειωματάριο Jupyter. Το os είναι μια λειτουργική μονάδα που παρέχει έναν τρόπο χρήσης λειτουργιών που εξαρτώνται από το λειτουργικό σύστημα.

Επειδή προέκυψε πρόβλημα με την εκτέλεση του cv2 έγινε χρήση της εντολής

```
< !pip install opencv – pytho>.
```

Η λειτουργική μονάδα cv2 ανήκει στην βιβλιοθήκη OpenCV και χρησιμοποιείται για ανίχνευση αντικειμένων, ανάλυση βίντεο και επεξεργασία εικόνας.

Έπειτα φορτώνουμε πακέτα από Tensorflow και Keras για τη δημιουργία μοντέλων βαθιάς μάθησης. Το Dense, το Flatten, το Conv2D, το MaxPool2D, το GlobalAveragePooling2D και το Dropout είναι τύποι επιπέδων που χρησιμοποιούνται για την κατασκευή νευρωνικών δικτύων.

Το Dense υλοποιεί ένα πλήρως συνδεδεμένο στρώμα, όπου κάθε νευρώνας λαμβάνει είσοδο από όλους τους νευρώνες στο προηγούμενο επίπεδο.

Το Flatten χρησιμοποιείται για την ισοπέδωση της πολυδιάστατης εισόδου σε ένα διάνυσμα 1D.

Το Conv2D υλοποιεί ένα 2D συνελκτικό επίπεδο, το οποίο χρησιμοποιείται συνήθως για εργασίες ταξινόμησης εικόνων.

Το MaxPool2D υλοποιεί ένα μέγιστο επίπεδο συγκέντρωσης, το οποίο υποβάλλει δειγματοληπτικά την είσοδο επιλέγοντας τη μέγιστη τιμή σε κάθε περιοχή συγκέντρωσης.

Το GlobalAveragePooling2D είναι ένα επίπεδο συγκέντρωσης που υπολογίζει τον μέσο όρο όλων των χαρτών χαρακτηριστικών.

Το Dropout είναι ένα επίπεδο τακτοποίησης που μηδενίζει τυχαία ορισμένες από τις εισόδους κατά τη διάρκεια της προπόνησης για να αποτρέψει την υπερπροσαρμογή.

Το Sequential και το Model είναι δύο τύποι μοντέλων στο keras API του TensorFlow για τη δημιουργία νευρωνικών δικτύων. Το Sequential είναι μια γραμμική στοίβα στρώσεων, όπου απλά προσθέτουμε ένα στρώμα τη φορά. Το μοντέλο επιτρέπει πιο σύνθετες αρχιτεκτονικές, όπου μπορούμε να καθορίσουμε πολλαπλές εισόδους και εξόδους.

Το Adam είναι ένας βελτιστοποιητής που ενημερώνει τα βάρη του μοντέλου με βάση τη διαβάθμιση της συνάρτησης απώλειας. Είναι μια δημοφιλής επιλογή για την εκπαίδευση μοντέλων βαθιάς μάθησης.

Το Xception είναι ένα προεκπαιδευμένο μοντέλο βαθιάς μάθησης το οποίο πορεί να χρησιμοποιηθεί ως εξαγωγή χαρακτηριστικών, αφαιρώντας τα επάνω στρώματα και προσθέτοντας νέα για να ταιριάζουν σε ένα νέο πρόβλημα ή ως βελτιστοποιητής, διατηρώντας μερικά από τα αρχικά επίπεδα και εκπαιδεύοντας το μοντέλο σε νέα δεδομένα.

Στην συνέχεια φορτώνουμε τα δεδομένα των εικόνων και τα εμφανίζουμε.

Φορτώνουμε το αρχείο csv.

```
In [5]: data1 = pd.read_csv('train.csv')
```

Ορίζουμε τους τύπους των πλοίων.

```
In [6]: ship = {0: 'Cargo', 1: 'Military', 2: 'Carrier', 3: 'Cruise', 4: 'Tankers'}
```


Εμφάνιση των 5 πρώτων γραμμών δεδομένων.

```
In [10]: def map_to_label(x):  
         return ship[x - 1]  
  
data1['label'] = data1['category'].apply(map_to_label)  
data1['label'] = pd.Categorical(data1['label'])
```

```
In [11]: data1.head(5)
```

Out[11]:

	image	category	label
0	2823080.jpg	1	Cargo
1	2870024.jpg	1	Cargo
2	2662125.jpg	2	Military
3	2900420.jpg	3	Carrier
4	2804883.jpg	2	Military

Στη πρώτη γραμμή ορίζουμε μια συνάρτηση `map_to_label` η οποία λαμβάνει μια είσοδο `x` και επιστρέφει την αντίστοιχη τιμή συμβολοσειράς στη λίστα πλοίων. Στη δεύτερη γραμμή εφαρμόζεται η συνάρτηση `map_to_label` σε κάθε στοιχείο της στήλης `category` δημιουργώντας μια νέα στήλη `label` και στην τρίτη γραμμή η στήλη `label` μετατρέπεται σε κατηγορική μεταβλητή δηλαδή σε μεταβλητή που λαμβάνει έναν περιορισμένο αριθμό πιθανών τιμών, που ονομάζονται κατηγορίες.

Εμφάνιση διαστάσεων πίνακα γραμμές=δείγματα, στήλες=χαρακτηριστικά.

```
In [12]: data1.shape
```

Out[12]: (6252, 3)

Άρα έχουμε 6252 γραμμές και 3 στήλες.

Καταμέτρηση μοναδικών τιμών για συγκεκριμένο χαρακτηριστικό.

```
In [13]: data1['label'].value_counts()
```

```
Out[13]: Cargo      2120  
         Tankers    1217  
         Military   1167  
         Carrier    916  
         Cruise     832  
         Name: label, dtype: int64
```

Παρατηρούμε ότι το σύνολο των δεδομένων δεν κατανέμεται εξίσου μεταξύ των κατηγοριών. Οπότε έχουμε 2120 δεδομένα για τα φορτηγά πλοία, 1217 για τα δεξαμενόπλοια, 1167 για τα στρατιωτικά, 916 για τα επιβατηγά και 832 για τα κρουαζιερόπλοια.

Στην συνέχεια ορίζουμε την διεύθυνση του αρχείου εικόνων.

```
In [14]: path = 'images'  
         os.listdir(path)
```

```
'1010082.jpg',  
'1016683.jpg',  
'1019271.jpg',  
'1048999.jpg',  
'1049854.jpg',  
'1051155.jpg',  
'1056547.jpg',  
'1056555.jpg',  
'1059245.jpg',  
'1062001.jpg',  
'1062006.jpg',  
'1069397.jpg',  
'1072012.jpg',
```

Η συνάρτηση `os.listdir(path)` επιστρέφει μια λίστα με όλα τα στοιχεία του αρχείου.

Αφού εισάγουμε τις εικόνες δημιουργούμε μια λίστα με τα ονόματα των εικόνων.

```
In [15]: img_list = list(data1['image'])
```

Ο κώδικας αρχικοποιεί πρώτα μια κενή λίστα `data_img`. Στη συνέχεια γίνεται εξαγωγή κάθε εικόνας από το αρχείο και διαβάζεται. Η ανάγνωση της εικόνας γίνεται χρησιμοποιώντας τη συνάρτηση `cv2.imread(each_path)`, η οποία διαβάζει την εικόνα και την επιστρέφει ως πίνακα NumPy. Έπειτα, οι εικόνες μετατρέπονται σε μορφή RGB και αλλάζουν μέγεθος ώστε να έχουν τις ίδιες διαστάσεις. Αυτό γίνεται γιατί οι περισσότερες βιβλιοθήκες διαβάζουν εικόνες σε RGB. Στο τέλος, κάθε εικόνα αποθηκεύεται σε μία λίστα με την μορφή πίνακα NumPy.

```
In [16]: data_img = []

for each in img_list:
    each_path = os.path.join(path, each)
    each_img = cv2.imread(each_path)
    each_img = cv2.cvtColor(each_img, cv2.COLOR_BGR2RGB)
    each_img_resized = cv2.resize(each_img, (126,126))
    data_img.append(each_img_resized)
X = np.array(data_img)
```

Στη συνέχεια καλούμε το `OneHotEncoder` από την `sklearn`, το οποίο δημιουργεί μία δυαδική στήλη για κάθε κατηγορία και επιστρέφει έναν πίνακα.

```
In [17]: from sklearn.preprocessing import OneHotEncoder

y = OneHotEncoder(dtype='int8', sparse=False).fit_transform(data1['category'].values.reshape(-1,1))
```

Το όρισμα `dtype` ορίζεται σε `'int8'` για να καθορίσει τον τύπο δεδομένων της εξόδου ενώ το `sparse=False` υποδεικνύει ότι η έξοδος θα πρέπει να είναι ένας πυκνός numpy πίνακας.

Η μέθοδος `fit_transform` εφαρμόζεται στη μεταβλητή προορισμού `data1['category']` αφού την αναδιαμορφώσει σε έναν πίνακα 2D χρησιμοποιώντας τη μέθοδο `reshape` (`data1['category'].values.reshape(-1,1)`).

Η πρώτη διάσταση του σχήματος έχει τιμή `-1`, που σημαίνει ότι ο αριθμός των σειρών πρέπει να υπολογίζεται αυτόματα από τον αριθμό των στοιχείων και των στηλών του πίνακα.

Παρακάτω βλέπουμε κάποιες φωτογραφίες από τις κατηγορίες πλοίων.

Οι μέθοδοι `xticks` και `yticks` χρησιμοποιούνται για την αφαίρεση των ετικετών των αξόνων `x` και `y`.

```
In [27]: train_images = np.random.randint(0,6252,4)
i = 1
plt.figure(figsize=(20,10))
for each in train_images:
    plt.subplot(1,4,i)
    i += 1
    plt.xticks([])
    plt.yticks([])
    plt.title(data1['label'].loc[each])
    plt.imshow(X[each])
```



Στην συνέχεια, περνάμε στον διαχωρισμό των δεδομένων για εκπαίδευση, επικύρωση και δοκιμή.

```
In [29]: from sklearn.model_selection import train_test_split

X_data, X_test, y_data, y_test = train_test_split(X, y, test_size=0.15, random_state=45)
X_train, X_val, y_train, y_val = train_test_split(X_data, y_data, test_size=0.15, random_state=45)
```

Η πρώτη γραμμή χρησιμοποιεί `train_test_split` για να χωρίσει τα δεδομένα (X) και τις ετικέτες τους (y) σε δύο σύνολα: X_{test} και y_{test} ως σύνολο δοκιμών (με μέγεθος 15% των συνολικών δεδομένων) και X_{data} και y_{data} ως τα υπόλοιπα δεδομένα. Στη συνέχεια, η δεύτερη γραμμή χωρίζει τα X_{data} και y_{data} σε σύνολο εκπαίδευσης (X_{train} και y_{train}) και σε σύνολο επικύρωσης (X_{val} και y_{val}) με την ίδια μέθοδο και με μέγεθος δοκιμής 15%. Το όρισμα `random_state=45` διασφαλίζει ότι τα ίδια τυχαία δείγματα επιλέγονται για κάθε εκτέλεση.

```
In [30]: print('X_train: ', X_train.shape)
print('y_train: ', y_train.shape)

X_train: (4378, 126, 126, 3)
y_train: (4378, 5)
```

```
In [31]: print('X_val: ', X_val.shape)
print('y_val: ', y_val.shape)

X_val: (937, 126, 126, 3)
y_val: (937, 5)
```

```
In [32]: print('X_test: ', X_test.shape)
print('y_test: ', y_test.shape)

X_test: (937, 126, 126, 3)
y_test: (937, 5)
```

Παρατηρούμε, λοιπόν, ότι έχουμε 4378 εικόνες για την εκπαίδευση, 937 για την επικύρωση και 937 για δοκιμή.

Στην συνέχεια εφαρμόζεται το data augmentation το οποίο αφορά τα δεδομένα εκπαίδευσης και επικύρωσης. Με την βοήθειά του το μοντέλο τροφοδοτείται με τυχαίες εικόνες και έτσι εκπαιδεύεται με περισσότερα δεδομένα. Τα αρχικά δεδομένα χρησιμοποιούνται ως είσοδος για το Keras ImageDataGenerator, το οποίο στη συνέχεια μετασχηματίζει τα δεδομένα με τυχαίο τρόπο και παράγει μια συνέχεια που περιέχει αποκλειστικά τα πρόσφατα τροποποιημένα δεδομένα. Το horizontal flip γυρίζει και τις γραμμές και τις στήλες. Επομένως, στον κατασκευαστή ImageDataGenerator, πρέπει να παρέχουμε την επιλογή horizontal flip=True για να γίνει αυτό. Η τιμή του είναι false από προεπιλογή.

```
In [33]: from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
In [34]: train_datagen = ImageDataGenerator(horizontal_flip=True,
                                             rotation_range=42,
                                             zoom_range=0.2,
                                             height_shift_range=0.5,
                                             width_shift_range=0.5)

train_datagen.fit(X_train)
train_generator = train_datagen.flow(X_train, y_train, batch_size=32)
```

```
In [35]: val_datagen = ImageDataGenerator(horizontal_flip=True,
                                             rotation_range=42,
                                             zoom_range=0.2,
                                             height_shift_range=0.5,
                                             width_shift_range=0.5)

val_datagen.fit(X_val)
val_generator = val_datagen.flow(X_val, y_val, batch_size=32)
```

Το rotation_range καθορίζει το εύρος εντός του οποίου θα περιστραφεί τυχαία η εικόνα. Το zoom_range καθορίζει το εύρος για τυχαίο ζουμ της εικόνας. Το height_shift_range καθορίζει το εύρος για την τυχαία κατακόρυφη μετατόπιση της εικόνας. Το width_shift_range καθορίζει το εύρος για τυχαία οριζόντια μετατόπιση της εικόνας.

Εκπαίδευση μοντέλου

Καθορισμός μεγεθών παρτίδας και εποχής.

Η ποσότητα των δειγμάτων σε μια παρτίδα καθορίζεται από το μέγεθος της παρτίδας. Μετά την επεξεργασία κάθε παρτίδας, το μοντέλο θα ενημερώσει τα βάρη του. Ο αριθμός των φορών που θα φανεί το πλήρες σύνολο δεδομένων εκπαίδευσης από το μοντέλο καθορίζεται ως προς τις εποχές κατά τη διάρκεια της εκπαίδευσης. Μια εποχή ολοκληρώνεται αφού το μοντέλο έχει περάσει από κάθε δείγμα στο σύνολο δεδομένων εκπαίδευσης μία φορά.

```
In [36]: batch_size = 100  
         epochs = 20
```

Καθορισμός προ-εκπαιδευμένου μοντέλου

Το βασικό μοντέλο είναι ένα μοντέλο Xception με τα επάνω επίπεδα απενεργοποιημένα (include_top=False).

Το head είναι ένα πυκνό στρώμα που εξάγει μια κατανομή πιθανότητας στις 5 κατηγορίες και έχει 5 μονάδες εξόδου με συνάρτηση ενεργοποίησης softmax.

```
In [37]: base = Xception(include_top=False, weights='imagenet', input_shape=(126,126,3))  
         x = base.output  
         x = GlobalAveragePooling2D()(x)  
         head = Dense(5, activation='softmax')(x)  
         model = Model(inputs=base.input, outputs=head)
```

Ο αλγόριθμος βελτιστοποίησης που χρησιμοποιείται είναι ο Adam με ρυθμό εκμάθησης 0,0001. Η συνάρτηση απώλειας που χρησιμοποιείται είναι η categorical_crossentropy, η οποία είναι κατάλληλη για προβλήματα ταξινόμησης πολλών κατηγοριών. Η ακρίβεια είναι η μέτρηση που χρησιμοποιείται για να μετρήσει πόσο καλά απέδωσε το μοντέλο κατά τη διάρκεια της εκπαίδευσης και της επικύρωσης.

```
In [38]: model.compile(optimizer=Adam(lr=0.0001),  
                      loss = 'categorical_crossentropy',  
                      metrics=['accuracy'])
```

Η μέθοδος `fit_generator` χρησιμοποιείται σε αυτήν τη γραμμή κώδικα για την εκπαίδευση του μοντέλου εκμάθησης μεταφοράς, επιτρέποντας στο μοντέλο να εκπαιδεύεται σε τεράστια σύνολα δεδομένων χωρίς να αντιμετωπίζει περιορισμούς στη μνήμη.

```
In [39]: model12 = model.fit_generator(train_datagen.flow(X_train, y_train, batch_size=batch_size),  
                                     epochs = epochs,  
                                     validation_data = val_datagen.flow(X_val, y_val, batch_size=batch_size),  
                                     steps_per_epoch = X_train.shape[0] // batch_size)
```

Το μοντέλο επαναλαμβάνει την διαδικασία για 20 εποχές. Παρακάτω φαίνονται κάποιες από αυτές.

```
Epoch 16/20  
43/43 [=====] - 357s 8s/step - loss: 0.2369 - accuracy: 0.9077 - val_loss: 0.4002 - val_accuracy: 0.86  
77  
Epoch 17/20  
43/43 [=====] - 352s 8s/step - loss: 0.2080 - accuracy: 0.9231 - val_loss: 0.4026 - val_accuracy: 0.86  
45  
Epoch 18/20  
43/43 [=====] - 356s 8s/step - loss: 0.2080 - accuracy: 0.9205 - val_loss: 0.3932 - val_accuracy: 0.86  
77  
Epoch 19/20  
43/43 [=====] - 356s 8s/step - loss: 0.1833 - accuracy: 0.9306 - val_loss: 0.3828 - val_accuracy: 0.87  
09  
Epoch 20/20  
43/43 [=====] - 343s 8s/step - loss: 0.1877 - accuracy: 0.9280 - val_loss: 0.4411 - val_accuracy: 0.85  
27
```


Στην συνέχεια φορτώνουμε τα δεδομένα για δοκιμή(test) και δοκιμάζουμε το μοντέλο μας για πρόβλεψη με τυχαίες εικόνες.

```
In [40]: test_img = pd.read_csv('test.csv')
```

```
In [41]: test_img.head(5)
```

Out[41]:

	image
0	1007700.jpg
1	1011369.jpg
2	1051155.jpg
3	1062001.jpg
4	1069397.jpg

Αρχικά πραγματοποιείται λήψη τυχαίων εικόνων και αποθηκεύονται σε μια λίστα. Στην συνέχεια πραγματοποιείται επεξεργασία αυτών των εικόνων και αποθηκεύονται σε έναν πίνακα NumPy. Έπειτα χρησιμοποιείται το εκπαιδευμένο μοντέλο για πρόβλεψη. Για να προσδιοριστεί η κλάση με την υψηλότερη πιθανότητα για κάθε εικόνα, το όρισμα του άξονα ορίζεται σε 1.

```
In [42]: sample_img = test_img.sample(10)
img_list = list(sample_img['image'])
```

```
In [43]: sample_img_data = [cv2.resize(cv2.cvtColor(cv2.imread(os.path.join(path, each)), cv2.COLOR_BGR2RGB), (126, 126)) for each in img_
sample_img_data = np.array(sample_img_data)
```

```
In [44]: sample_pred = model.predict(sample_img_data)
sample_classes = np.argmax(sample_pred, axis = 1)
```

```
In [45]: y_pred = model.predict(X_test)
y_pred_classes = np.argmax(y_pred, axis = 1)
y_test_classes = np.argmax(y_test, axis = 1)
```

Παρακάτω βλέπουμε κάποιες εικόνες από το dataset και παρατηρούμε ότι δεν έχουν ετικέτα.

```
In [47]: test_images = np.random.randint(0,6252,4)
i = 1
plt.figure(figsize=(20,10))
for each in test_images:
    plt.subplot(1,4,i)
    i += 1
    plt.xticks([])
    plt.yticks([])
    plt.title(data1['image'].loc[each])
    plt.imshow(X[each])
```



Μετά την πρόβλεψη παρατηρούμε ότι οι εικόνες έχουν αντιστοιχηθεί στις κατηγορίες των πλοίων.

```
In [48]: i = 0
plt.figure(figsize=(20,14))
for each in range(10):
    i += 1
    plt.subplot(2,5,i)
    plt.xlabel(str(ship[sample_classes[each]]))
    plt.imshow(sample_img_data[each])
    plt.xticks([])
    plt.yticks([])
```



Cargo



Tankers



Cargo



Carrier



Cargo



Cargo



Tankers



Cargo



Cruise



Military

Για την βαθμολογία μοντέλου εισάγουμε το `classification_report` από την βιβλιοθήκη `sklearn`. Η συνάρτηση `classification_report` λαμβάνει ως είσοδο τις αληθινές ετικέτες κλάσεων (`y_test_classes`) και τις προβλεπόμενες ετικέτες κλάσεων (`y_pred_classes`) και επιστρέφει μια αναφορά που περιλαμβάνει τις ακόλουθες μετρήσεις αξιολόγησης:

```
In [49]: from sklearn.metrics import classification_report
```

```
In [52]: print(classification_report(y_test_classes, y_pred_classes))
```

	precision	recall	f1-score	support
0	0.88	0.88	0.88	333
1	0.95	0.92	0.93	175
2	0.99	0.97	0.98	140
3	0.99	0.89	0.94	124
4	0.79	0.88	0.83	165
accuracy			0.90	937
macro avg	0.92	0.91	0.91	937
weighted avg	0.91	0.90	0.90	937



Εικόνα 22. Ανάλυση της απόδοσης των μοντέλων ταξινόμησης[29]

Precision

Ο αριθμός των αληθινών θετικών περιπτώσεων διαιρεμένος με τον αριθμό των αληθινών θετικών συν ψευδώς θετικών περιπτώσεων. Η ακρίβεια μετρά πόσο καλά ο ταξινομητής αποφεύγει τις ψευδώς θετικές προβλέψεις.

Το Precision είναι ο λόγος: $tp / (tp + fp)$.

Recall

Ο αριθμός των αληθινών θετικών περιπτώσεων διαιρεμένος με τον αριθμό των αληθινών θετικών συν ψευδώς αρνητικών περιπτώσεων. Η ανάκληση μετρά πόσο καλά ο ταξινομητής ανιχνεύει θετικές περιπτώσεις.

Η ανάκληση είναι ο λόγος: $tp / (tp + fn)$.



F1_score

Το F1_score είναι ο μέσος όρος ακρίβειας (Precision) και ανάκλησης (Recall). Η σχετική συμβολή της ακρίβειας και της ανάκλησης στο F1_score είναι ίση. Η τιμή του F1score κυμαίνεται από 0 έως 1, όπου το 1 αντιπροσωπεύει την καλύτερη έξοδο μοντέλου και το 0 αντιπροσωπεύει τη χειρότερη έξοδο μοντέλου. Ο τύπος για τη βαθμολογία F1 είναι: $F1 = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$.

Support

Ο αριθμός των περιπτώσεων στο σύνολο δοκιμών για κάθε τάξη.

Η αναφορά ταξινόμησης παρέχει μια επισκόπηση του πόσο καλά αποδίδει ο ταξινομητής όσον αφορά την ακρίβεια, την ανάκληση και τη βαθμολογία f1 για κάθε τάξη στο σύνολο δοκιμής.

Κεφάλαιο 8

Συμπεράσματα

Στην παρούσα διπλωματική εργασία παρουσιάζεται το απαραίτητο θεωρητικό υπόβαθρο για την κατανόηση τόσο των βασικών εννοιών όσο και των αλγορίθμων που χρησιμοποιούνται για την ανάλυση των μεθόδων εύρεσης συντομότερων διαδρομών, καθώς και των δομών σχεδίασης και υλοποίησης ενός μοντέλου ταξινόμησης με την βοήθεια ενός προ-εκπαιδευμένου νευρωνικού δικτύου. Στο πρώτο στάδιο της εργασίας πραγματοποιείται ανάλυση τεσσάρων βασικών αλγορίθμων δρομολόγησης και ενός τροποποιημένου και αναφέρεται χαρακτηριστικά η δομή και ο τρόπος λειτουργίας τους. Για λόγους πληρότητας ακολουθεί το θεωρητικό κομμάτι για την επεξήγηση των μεθόδων Μηχανικής Μάθησης, Νευρωνικών Δικτύων και Βαθιάς Μάθησης και στην συνέχεια παρουσιάζεται το πρακτικό μέρος της εργασίας. Κατά το πρακτικό κομμάτι, δημιουργείται ένα μοντέλο κατηγοριοποίησης πλοίων μέσα από ένα σύνολο δεδομένων εικόνων. Δημιουργούνται πέντε κλάσεις, μια για κάθε τύπο πλοίου και πραγματοποιείται ανάλυση των δεδομένων. Στην συνέχεια προχωράμε στον διαχωρισμό των δεδομένων για εκπαίδευση, επικύρωση και δοκιμή. Για την εκπαίδευση χρησιμοποιείται το μοντέλο Xception το οποίο είναι ένα βαθύ συνελικτικό νευρωνικό δίκτυο. Από τα αποτελέσματα που εμφανίζονται στον πίνακα των σκορ διακρίνουμε ότι το μοντέλο έχει καλή απόδοση με 91% ακρίβεια και F1_score. Επιπλέον δοκιμάζουμε να δούμε αν το μοντέλο μας ανταποκρίνεται σε ένα σύνολο δεδομένων με τυχαίες εικόνες, πραγματοποιώντας αντιστοίχιση ετικετών των κατηγοριών των πλοίων σε κάθε εικόνα.

Το σημείο που χρειάζεται περαιτέρω βελτίωση είναι η βελτιστοποίηση κατά την διάρκεια εκπαίδευσης του μοντέλου καθώς ο χρόνος που απαιτείται για τουλάχιστον είκοσι εποχές, είναι αρκετός. Σίγουρα και η επίτευξη μεγαλύτερου ποσοστού ακρίβειας αφήνει ανοιχτό το ενδεχόμενο για μελλοντική βελτίωση.

Βιβλιογραφία

1. Heuristic shortest path algorithms for transportation applications: State of the art, L. Fu, D. Sun, L.R. Rilett
2. Dynamic anti-collision A-star algorithm for multi-ship encounter situations, Zhibo He, Chenguang Liu, Xiumin Chu, Rudy R. Negenborn , Qing Wu
3. A formal Basis for the Heuristic Determination of Minimum Cost Paths, PETER E. HART, NILS J. NILSSON, BERTRAM RAPHAEL
4. An Improved multi-objective a-star algorithm for path planning in a large workspace: Design, Implementation, and Evaluation, Oluwaseun Opeyemi Martins, PhD, Adefemi Adeyemi Adekunle, Olatayo Moses Olaniyan, Bukola Olalekan Bolaji
5. Bi-Directional Search, Thomas J. Watson Research Center I B M Corporation, New York
6. A Theorem on the Expected Complexity of Dijkstra's Shortest Path Algorithm, KOHEI NOSHITA, Department of Computer Science, Denkitusin University, Choju, Tokyo 182, Japan Received November 10,1983; accepted June 26,1984
7. Introduction to Algorithms, Third Edition, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein
8. Improved A-Star Algorithm for Long-Distance Off-Road Path Planning Using Terrain Data Map, Zhonghua Hong, Pengfei Sun, Xiaohua Tong, Haiyan Pan, Ruyan Zhou, Yun Zhang, Yanling Han, Jing Wang, Shuhu Yang and Lijun Xu
9. Implications of Pooling Strategies in Convolutional Neural Networks: A Deep Insight, Shallu Sharma*, Rajesh Mehra
10. Neural Computing: An Introduction, R Beale and T Jackson Department of Computer Science, University of York
11. Speed-Up Techniques for Shortest-Path Computations, Dorothea Wagner and Thomas Willhalm

12. Autonomous Vessels' Pathfinding Using Visibility Graph, Eric Kulbiej
13. AUV Path Planning in Dynamic Environment Based on Improved Artificial Potential Field Method Based on Visibility Graph, Zhou Xin, Xu Rongwu, Cheng guo
14. Path planning of multiple autonomous marine vehicles for adaptive sampling using Voronoi-based ant colony optimization, Chengke Xiong, Danfeng Chen, Di Lu, Zheng Zeng a,b,c, Lian Lian
15. Lecture 7: Voronoi Diagrams, Presented by Allen Miu 6.838 Computational Geometry September 27, 2001
16. Xception: Deep Learning with Depthwise Separable Convolutions, Francois Chollet
17. Διαμαντάρας Κωνσταντίνος, Μπότσης Α. Δημήτρης, «Μηχανική Μάθηση», 2019.
18. Μηχανική Μάθηση και Ανάλυση Μεγάλων Δεδομένων(Σημειώσεις), Πανεπιστήμιο Δυτικής Αττικής
19. Data Science with Python, Προγράμματα Συμπληρωματικής Εκπαίδευσης, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
20. A Neural Network approach to predict Marine Traffic, Andrius Daranda October 2016 Technical Report MII-DS-07T-16-9-16
21. Keras & TensorFlow: <https://keras.io/about/>
22. what-are-max-pooling-average-pooling-global-max-pooling-and-global-average-pooling: <https://github.com/christianversloot/machine-learning-articles/blob/main/what-are-max-pooling-average-pooling-global-max-pooling-and-global-average-pooling.md>
23. Multi-Class Ship Classification: <https://www.kaggle.com/code/akdagmelih/multi-class-ship-classification-xception>
24. Classification of Ship Images: <https://www.kaggle.com/code/teeyee314/classification-of-ship-images>
25. Path Planning and Real-Time Collision Avoidance Based on the Essential Visibility Graph, Luciano Blasi, Egidio D'Amato, Massimiliano Mattei, Immacolata Notaro



26. An energy-efficient path planning algorithm for unmanned surface vehicles, Hanlin Niu, Yu Lu, Al Savvaris, Antonios Tsourdos
27. Graph Theory and Its Applications, By Jonathan L. Gross, Jay Yellen, Mark Anderson
28. Exploring Convolution Matrices with JIMP
29. <https://viso.ai/deep-learning/analyzing-machine-learning-model-performance-strategies/>